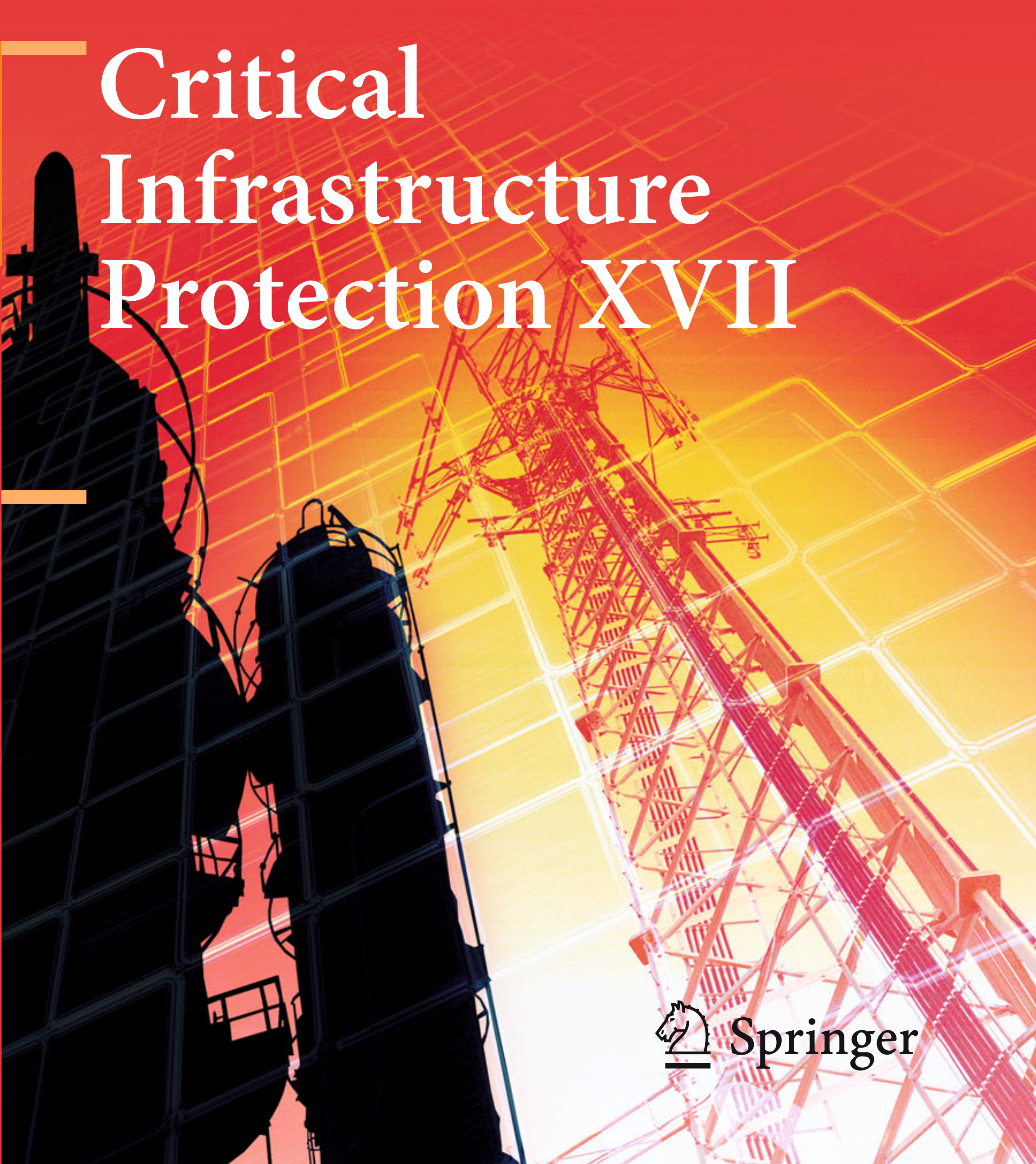
IFIP AICT 686

Jason Staggs
Sujeet Shenoi
(Eds.)

Critical Infrastructure Protection XVII



Springer

IFIP Advances in Information and Communication Technology 686

IFIP Advances in Information and Communication Technology

The IFIP AICT series publishes state-of-the-art results in the sciences and technologies of information and communication. The scope of the series includes: foundations of computer science; software theory and practice; education; computer applications in technology; communication systems; systems modeling and optimization; information systems; ICT and society; computer systems technology; security and protection in information processing systems; artificial intelligence; and human-computer interaction.

Edited volumes and proceedings of refereed international conferences in computer science and interdisciplinary fields are featured. These results often precede journal publication and represent the most current research.

The principal aim of the IFIP AICT series is to encourage education and the dissemination and exchange of information about all aspects of computing.

More information about this series at https://link.springer.com/bookseries/6102

Jason Staggs · Sujeet Shenoi
Editors

Critical Infrastructure Protection XVII

17th IFIP WG 11.10 International Conference, ICCIP 2023
Arlington, VA, USA, March 13–14, 2023
Revised Selected Papers

Springer

Editors
Jason Staggs
University of Tulsa
Tulsa, OK, USA

Sujeet Shenoi
University of Tulsa
Tulsa, OK, USA

ISSN 1868-4238 ISSN 1868-422X (electronic)
IFIP Advances in Information and Communication Technology
ISBN 978-3-031-49584-7 ISBN 978-3-031-49585-4 (eBook)
https://doi.org/10.1007/978-3-031-49585-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

The information infrastructure – comprising computers, embedded devices, networks and software systems – is vital to operations in every sector: chemicals, commercial facilities, communications, critical manufacturing, dams, defense industrial base, emergency services, energy, financial services, food and agriculture, government facilities, healthcare and public health, information technology, nuclear reactors, materials and waste, transportation systems, and water and wastewater systems. Global business and industry, governments, indeed society itself, cannot function if major components of the critical information infrastructure are degraded, disabled or destroyed.

This book, *Critical Infrastructure Protection XVII*, is the seventeenth volume in the annual series produced by IFIP Working Group 11.10 on Critical Infrastructure Protection, an active international community of scientists, engineers, practitioners and policy makers dedicated to advancing research, development and implementation efforts related to critical infrastructure protection. The book presents original research results and innovative applications in the area of critical infrastructure protection. Also, it highlights the importance of weaving science, technology and policy in crafting sophisticated, yet practical, solutions that will help secure information, computer and network assets in the various critical infrastructure sectors.

This volume contains 11 selected papers from the Seventeenth Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, which was held at SRI International in Arlington, Virginia, USA on March 13–14, 2023. A total of 19 full-length papers were submitted for presentation at the conference. The papers were refereed in a single-blind manner by members of the conference program committee and other individuals, all of them internationally-recognized experts in critical infrastructure protection. The 11 accepted papers were rewritten by the authors after the conference to accommodate the suggestions provided by the referees and by the conference attendees. The 11 post-conference manuscripts were subsequently revised by the editors to produce the final chapters published in this volume.

The chapters are organized into five thematic sections: (i) Themes and Issues; (ii) Smart Grid Risks and Impacts; (iii) Network and Telecommunications Systems Security; (iv) Infrastructure Security; and (v) Automobile Security. The coverage of topics showcases the richness and vitality of the discipline, and offers promising avenues for future research in critical infrastructure protection.

This book is the result of the combined efforts of several individuals and organizations. In particular, we thank Laura Tinnel for her tireless work on behalf of IFIP Working Group 11.10. We also thank the National Science Foundation, U.S. Department of Homeland Security, National Security Agency and SRI International for their support of IFIP Working Group 11.10 and its activities. Finally, we wish to note that all opinions, findings, conclusions and recommendations in the chapters of this book are those of the authors and do not necessarily reflect the views of their employers or funding agencies.

October 2023

Jason Staggs
Sujeet Shenoi

Conference Organization

General Chair

Laura Tinnel	SRI International, USA

Program Chairs

Jason Staggs	University of Tulsa, USA
Sujeet Shenoi	University of Tulsa, USA

Program Committee

David Balenson	University of Southern California, Information Sciences Institute, USA
Raymond Chan	Singapore Institute of Technology, Singapore
Kam-Pui Chow	University of Hong Kong, China
Chiara Foglietta	University Roma Tre, Italy
Richard George	Johns Hopkins University, Applied Physics Laboratory, USA
Scott Graham	Air Force Institute of Technology, USA
Janne Hagen	Norwegian Water Resources and Energy Directorate, Norway; and University of Oslo, Norway
Michael Haney	University of Idaho, USA
Shaun Kelso	U.S. Navy, USA
Stefano Panzieri	University Roma Tre, Italy
Mason Rice	Oak Ridge National Laboratory, USA
Leon Strous	De Nederlandsche Bank, The Netherlands
Øyvind Toftegaard	Norwegian University of Science and Technology, Norway; and Norwegian Energy Regulatory Authority, Norway
Zachary Tudor	Idaho National Laboratory, USA

Contents

Themes and Issues

Smart Grid Risks and Impacts

Network and Telecommunications Systems Security

Infrastructure Security

Automobile Security

Themes and Issues

Redefining Homeland Security

Richard White[✉]

University of Colorado Colorado Springs, Colorado Springs, CO, USA
rwhite2@uccs.edu

Abstract. Definitions are important, especially in the U.S. federal government. They are the basis of laws that justify budgets, fund programs and determine capabilities. However, definitions are notoriously difficult to cast because they must contend with exceptions and changing circumstances. This is the case with the U.S. definition of homeland security.

Despite its importance, the definition of homeland security has languished for years. The definition posted on the U.S. Department of Homeland Security website is a throwback to the original 2002 definition and apparently ignores the lessons of history that demonstrate it is deficient. In 2007, the U.S. Congress passed a law mandating a Quadrennial Homeland Security Review to prevent future lapses in homeland security. However, the definition that emerged from the first review in 2010 persists. Although it improves on the original 2002 definition, it does not adequately consider new and resurgent threats that face the nation.

This chapter examines various definitions of homeland security, discusses why they are inadequate and proposes a new definition that is accurate and concise. A good definition is important to help shape the U.S. Department of Homeland Security mission, set priorities, justify budgets and ensure that programs are successful.

Keywords: Homeland Security · Definition · Non-State Actors · Terrorism · Weapons of Mass Destruction

1 Introduction

The International Federation for Information Processing (IFIP) is a leading multinational, non-governmental organization in the information and communications science and technology domains. So why should members of an IFIP Technical Committee focused on security and privacy protection in information processing systems care about homeland security? The answer is simple. Cyber security is an essential component of critical infrastructure protection, which is essential to homeland security, which is about safeguarding a nation from domestic catastrophic destruction. Homeland security is a priority to all nations. Yet for something so important, the concept of homeland security is often misunderstood in the United States, resulting in confusion and disruptions that have undermined and detracted from the homeland security mission. The root of the problem, at least in the United States, lies with those entrusted with managing

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 3–14, 2024.
https://doi.org/10.1007/978-3-031-49585-4_1

homeland security. Indeed, for many reasons, over more than 20 years, they have failed to cast an accurate and concise definition of homeland security.

Definitions are useful tools that help distinguish, separate and bound concepts. In a governmental context their importance cannot be underestimated because they underpin missions, funding and capabilities. But definitions are also notoriously difficult to cast. Within a universe of infinite possibilities, exceptions are a certainty. Definitions are also subject to change over time due to evolving language and impinging circumstances. The definition of homeland security is no different. The concept is seemingly too complex to capture and previous attempts have proved insufficient to cope with the scope and scale of the unprecedented events encountered in recent U.S. history. This may be why the official U.S. definition of homeland security has languished since it was last updated in 2010. Certainly, the world situation has not languished during this time. Now, as the United States faces a new array of unprecedented threats it is appropriate to revisit the definition of homeland security.

2 Motivation

On June 12, 2016, a shooter entered the Pulse nightclub in Orlando, Florida, killing 49 people and wounding 53 more. It was the deadliest mass shooting in the United States at the time. The Federal Bureau of Investigation (FBI) deemed it a terrorist attack because the shooter mentioned that it was intended to stop U.S. bombing in Syria and Iraq.

On October 1, 2017, a shooter on the 32nd floor of the Mandalay Bay Resort and Casino fired more than 1,000 rounds into a concert crowd on the Las Vegas strip, killing 60 people and wounding at least 413. The motive for the attack has not been determined and it remains the deadliest mass shooting in U.S. history [1].

The Orlando and Las Vegas shootings epitomize an unfortunate trend in domestic mass killings. They have conveyed perceptions among the general public that every mass shooting is a terrorist incident and every terrorist incident is a homeland security incident. Neither perception is correct, but members of the public can hardly be blamed for their confusion.

Immediately after the Orlando shooting, Secretary Jeh Johnson of the U.S. Department Homeland Security (DHS) said his agency was “dedicated to investigating the tragedy, along with the [Federal Bureau of Investigation] and [its] state and local partners, and supporting the Orlando community in the tragedy’s aftermath” [6]. After the Las Vegas incident, Acting Secretary Elaine Duke announced that the agency was “closely monitoring the situation and working with [its] federal, state and local partners in responding to and investigating [the] tragedy” [14].

Although the statements made by the U.S. Department of Homeland Security leaders expressed the sincere desire of public officials to do everything in their power in the wake of the national tragedies, the fact of the matter is that the incidents fell outside the Department’s mission and charter. Indeed, the U.S.

Department of Homeland Security had no investigative authority or capability to intercede in what were fundamentally local law enforcement matters [25]. Despite their good intentions, the statements by the U.S. Department of Homeland Security leadership may have made matters worse by blurring jurisdictional boundaries and putting other agencies on the defensive. Indeed, one might contend that the overall homeland security efforts were negatively impacted, if not placed in jeopardy. Fixing the problem requires an accurate and concise definition of homeland security.

3 Searching for Hidden Meaning

The U.S. Department of Homeland Security website (www.dhs.gov) would appear to be the right place to look up the definition of homeland security. However, the definition is difficult to find. One has to conduct a site search that points to Instruction Manual 262-12-001-01, DHS Lexicon (2017 edition, revision 2.1) [23]. After downloading the PDF file and scrolling down to page 301, the following definition of homeland security is encountered:

> "... a concerted national effort to prevent terrorist attacks within the United States, reduce America's vulnerability to terrorism, and minimize the damage and recover from attacks that do occur."

In the same instruction manual, the definition is extended as follows:

> "... includes actions to prepare for, protect against, prevent, respond to, and recover from all threats or acts of terrorism."

And finally, the definition is annotated as follows:

> "While the Department of Homeland Security is the lead federal agency for mitigating vulnerabilities, threats and incidents related to terrorism, its responsibilities also include: preparing for, responding to, and recovering from natural disasters; stemming illegal drug flows; thwarting fraudulent immigration; strengthening border security; promoting the free flow of commerce; and maintaining civil rights."

This definition of homeland security is interesting. First, it appears to regress to the original definition and overlook the historical reasons for subsequent changes to the definition. Second, the dissembling additions acknowledge the inadequacy of the current definition and lend credence to the difficulty of casting the definition. Third, the definition, which is buried deep in the U.S. Department of Homeland Security website, lends weak support to the Department's mission statement [24] that prominently extols:

> "With honor and integrity, we will safeguard the American people, our homeland, and our values."

This raises the question of safeguarding Americans from what? Terrorism? Not exactly. Title 18, Sect. 2331 of the United States Code defines "terrorism" as a violent act designed to coerce the U.S. government. Terrorism has been a U.S. concern since the 19th century, but homeland security did not enter the jargon until the 21st century. What changed to bring homeland security to the forefront of U.S. security concerns?

4 Terrorism

The 1995 Tokyo Subway attacks brought homeland security to the fore. On March 20, 1995, five members of AUM Shinrikyo entered Tokyo Subway stations and punctured bags filled with liquid Sarin inside passenger cars packed with office workers on their morning commute. Twelve people were killed and 5,500 sought medical attention. Experts say it could have been much worse. Sarin is a chemical nerve agent so deadly that a single drop can kill a grown man [13].

This was deemed a terrorist attack because the leader of AUM Shinrikyo intended it would bring down the Japanese government. But what made it unique was that it was the first deployment of a weapon of mass destruction by a non-state actor. At the time, only nation states were thought to have the technical resources needed to manufacture weapons of mass destruction. AUM Shinrikyo proved this notion wrong. Also, it positioned homeland security between national security and law enforcement in order to deal the potential use of weapons of mass destruction by non-state actors.

In the United States, weapons of mass destruction was a national security concern and non-state actors was a law enforcement concern. Strict separation was maintained between the two to protect the rights of citizens. As a result of the Tokyo Subway attacks, in June 1995, President Clinton signed Presidential Decision Direction #39 (PDD-39) [18], creating a framework for U.S. law enforcement and intelligence agencies to coordinate to prevent similar weapons of mass destruction attacks in the United States.

Unfortunately, this did not work, as the events of Tuesday morning, September 11, 2001 demonstrated to worldwide horror. Nineteen men hijacked four aircraft and flew three of them into the Twin Towers in New York City, and the Pentagon outside Washington, DC. Learning of their impending fate from cell phone conversations with friends and family, passengers aboard the fourth jet attempted to seize control of the plane until the hijackers flew it into the ground outside Shanksville, Pennsylvania. The 9/11 attacks killed 2,996 people (including the hijackers) and caused up to $50 billion in direct damage, including the total destruction of the iconic Twin Towers [4].

Passenger manifests revealed that the hijackers were members of al-Qaeda operating in Afghanistan. Al-Qaeda staged the 9/11 attacks to coerce the U.S. to remove its forces stationed in Saudi Arabia. By definition, the 9/11 attacks were collectively an act of terrorism, but this is not what made it unique. According to the investigative National Commission [12], the 9/11 attacks were unique due to their surprising disproportionality. Nineteen hijackers inflicted as much damage as the Japanese Imperial Fleet did to Pearl Harbor on December 7, 1941. The hijackers did not use weapons of mass destruction. Instead, they created weapons of mass destruction effects by turning passenger jets into guided missiles [12].

Following the 9/11 attacks, weapons of mass destruction attacks by non-state actors became shorthand for terrorism. Faced with a problem perched between national security and law enforcement, President Bush decided to create a new agency devoted to homeland security.

The 9/11 attacks brought homeland security to the forefront of U.S. security concerns and shaped the very definition of the concept. The 2002 National Strategy for Homeland Security [19] highlights the following definition in a striking blue box:

> "Homeland Security is a concerted national effort to prevent terrorist attacks within the United States, reduce America's vulnerability to terrorism, and minimize the damage and recover from attacks that do occur."

The Homeland Security Act of 2002 that established the U.S. Department of Homeland Security was greatly influenced by this definition. The largest U.S. government reorganization since the National Security Act of 1947 began with a new definition that arose from an unprecedented threat. Indeed, the National Security Act of 1947 that reorganized U.S government after World War II was itself precipitated by the failures leading up to the Japanese attack on Pearl Harbor and an attempt to prevent surprise attacks potentially using atomic weapons.

The 2002 government reorganization was perfectly logical, except that the underlying definition was flawed. The concept of homeland security did not consider an important non-state actor that is more than capable of creating weapons of mass destruction effects – Mother Nature.

5 Natural Disasters

The tropical depression that became Hurricane Katrina formed over the Bahamas on August 23, 2005. It became a Category 5 storm with winds in excess of 157 mph after it crossed Florida and entered the Gulf of Mexico on August 26. It churned along the Gulf Coast causing destruction across Florida, Alabama and Mississippi before striking New Orleans on August 29. Although the city was heavily battered, it withstood the brunt of the 125 mph winds from the downgraded Category 3 storm. But the 8 to 10 in. of rain and the 14-foot storm surge overwhelmed the city's levees and drainage canals. The Mississippi River Gulf Outlet breached its levees at 20 locations and the federally-built levee system protecting downtown New Orleans was breached at 53 locations. By August 31, 80% of New Orleans was flooded, some parts with 15 feet of water. Hurricane Katrina resulted in 1,392 fatalities, many of them considered preventable [20].

Government entities at all levels failed to plan and prepare for and aggressively respond to Hurricane Katrina. However, the primary blame fell on the U.S.

Department of Homeland Security. In 1979, President Carter issued Executive Order 12148 that created the Federal Emergency Management Agency to coordinate federal efforts and assist state and local governments in protecting citizens from natural disasters. In 2003, the Federal Emergency Management Agency was moved into the new U.S. Department of Homeland Security under the premise, proven during 9/11, that the same first responders who deployed during natural disasters would also be needed following weapons of mass destruction attacks. However, after the disastrous hurricane response, the U.S. Congress felt that the U.S. Department of Homeland Security's focus on terrorism had detracted from its emergency management responsibilities [20].

After the Hurricane Katrina failures, a revised National Strategy for Homeland Security was released in 2007 that expanded the homeland security mission beyond terrorism by stressing the importance of preventing, protecting, responding and recovering from catastrophic events stemming from all hazards [8]. It did not go unnoticed that the United States had twice been caught unprepared by unprecedented new threats in less than a decade. Accordingly, when the U.S. Congress passed the Implementing Recommendations of the 9/11 Commission Act of 2007, it included provisions for the U.S. Department of Homeland Security to conduct a comprehensive examination of its mission and organization every four years starting in 2009. The first Quadrennial Homeland Security Review was released in 2010. The 2010 Quadrennial Homeland Security Review [22] redefined homeland security as follows:

> "Homeland Security is a concerted national effort to ensure a homeland that is safe, secure and resilient against terrorism and other hazards where American interests, aspirations and way of life can thrive."

Again, the DHS Lexicon [23] is not particularly helpful with the word "hazard," only mentioning that it is a "source or cause of harm or difficulty." Hazard alludes to a natural disaster given its inclusion following Hurricane Katrina, although the extended definition says hazards "may be natural, technological or human-caused." Nevertheless, the current definition of homeland security found on the U.S. Department of Homeland Security website focuses exclusively on terrorism. It makes no mention of other hazards, natural or human-initiated. Indeed, the current definition is a recapitulation of the very first definition of homeland security and is flawed for all the same reasons as the first definition.

6 Words Matter

Perhaps the current definition of homeland security is an oversight. After all, it is highly unlikely that U.S. Department of Homeland Security leadership pored over the 746-page lexicon to validate its contents, although they should not be expected to do so. However, the leadership is expected to maintain a viable definition of homeland security and ensure that it is prominently exhibited and widely known. This is because words matter, especially in the U.S. federal government.

Consider, for example, the definition of “mass killing.” Title 28, Sect. 530C of the United States Code defines a mass killing as an incident in which three or more people are killed. In 2013, the Federal Bureau of Investigation sought to set this definition in U.S. law to establish a threshold for lending assistance when requested by state and local law enforcement [21]. The threshold had to be codified into law because the Federal Bureau of Investigation is funded by the U.S. Congress. In fact, all federal agencies are funded by congressional appropriations enacted as laws and restrictions on the appropriations are also stipulated in laws. The funding amounts and restrictions on appropriations play a large role in determining agency priorities and programs, that in turn, determine boundaries and capabilities [16]. Accordingly, definitions are very important.

Assume, with good reason, that the definition in the 2010 Quadrennial Homeland Security Review is the official U.S. definition of homeland security. Given the historical insights, this definition is more comprehensive than the one in the lexicon because it addresses terrorism and other hazards. But is the definition complete? The remarks by the U.S. Department of Homeland Security leadership following the Orlando and Las Vegas shootings would indicate that the definition is incomplete.

7 Scale and Scope Matter

As horrific as they were, the Orlando and Las Vegas killings were not of the same scale as the 9/11 attacks and Hurricane Katrina, the two homeland security benchmarks. Missing from the official definition of homeland security is a sense of scale that delineates the boundary between homeland security incidents and other incidents. Perhaps what is needed is some form of the word “catastrophe.”

It was understood at the outset that scale matters. President Bush’s 2002 proposal for the U.S. Department of Homeland Security [5] made the distinction very clear by stating: “This would create a single office whose primary mission is the critical task of protecting the United States from catastrophic terrorism.” Somehow the word “catastrophic” got lost along the way.

The term catastrophic needs to be included in the homeland security definition. It would be significant in distinguishing homeland security incidents from other criminal acts. The addition would clarify U.S. Department of Homeland Security’s jurisdictional boundaries, which affect organizational planning and preparation, and shape budget priorities that are ultimately established by law. It would also inform public expectations so that the U.S. Department of Homeland Security would not have to make pretensions that lead to confusion.

Establishing a scale threshold is an improvement, but the notion of scope is also needed in the homeland security definition. A problem with the 2010 Quadrennial Homeland Security Review is its focus on terrorism and hazards. Terrorism covers intentional acts whose motive is to coerce the U.S. government whereas hazards are unintentional acts of nature and accidents. Are there no other acts, intentional or unintentional, that could result in domestic catastrophic destruction? Is the U.S. Department of Homeland Security making the same mistake it made before Hurricane Katrina?

The 2010 Quadrennial Homeland Security Review that created the current homeland security definition also identified the emerging threat from cyber attacks. In fact, cyber attacks had previously been identified as a potential catastrophic threat in a 1997 report by the President's Commission on Critical Infrastructure Protection after the 1995 Tokyo Subway attacks [15]. Specifically, the report noted that large-scale integration of computer controls in critical infrastructure assets might one day make them vulnerable to cyber attacks. This was prescient because back in 1997 the Internet had no more than 70 million users worldwide (1.7% of the global population), was just entering into commercial use and cyber attacks were relatively unknown.

By 2010, the Internet had grown to two billion users (28.7% of the global population) and three major cyber attacks had given security officials cause for concern. In 2007, government, bank and media websites across Estonia were shut down in what is considered to be the first act of cyber warfare [11]. In 2008, malicious code in a thumb drive breached Pentagon security and infiltrated classified and unclassified U.S. military networks [2]. In 2010, word got out that the Stuxnet virus breached a high-security uranium processing facility in Iran and successfully targeted Siemens industrial control systems that operated uranium hexafluoride centrifuges [10].

Cyber attacks on critical infrastructure assets are a clear and present danger. Project Aurora, a joint experiment conducted in 2007 by the U.S. Departments of Homeland Security and Energy demonstrated the ability to physically destroy a baseline electrical generator by remotely hacking into its controls [26]. The possibility became real in December 2015 when a cyber attack knocked out power in Ukraine's capital Kiev [7]. In 2018, the cyber threat struck closer to home when the U.S. Department of Homeland Security issued an alert about Russian infiltration into the U.S. power grid [17]. The consequences of shutting down power would be worse than any hurricane or earthquake because the damage would be nationwide and long term – it could be the worst disaster since the Civil War. Other cyber attack scenarios with disastrous consequences include undermining the Federal Reserve and causing a nuclear reactor meltdown.

Massive cyber attacks are not included the current definition of homeland security. Should they be incorporated in a new definition along with terrorism and hazards?

8 Effects Not Threats

Extending the definition of homeland security to include every possible threat is a futile pursuit. For example, if cyber attacks are added, should civil defense also be incorporated?

Civil defense protects the domestic population from enemy attacks. The Civil Defense Act of 1950 was enacted to protect U.S. citizens from the growing threat of nuclear attacks by the Soviet Union. The act created the Federal Civil Defense Administration tasked with assisting state and local governments with developing plans and preparing measures to deal with the unthinkable. Remaining secluded in a fallout shelter for weeks on end only to emerge to unimaginable devastation was deemed the best way to survive a nuclear attack. The idea was so abhorrent that public support quickly evaporated and U.S. Congress never approved funding for a massive sheltering program.

In 1991, the world breathed a collective sigh of relief when the Soviet Union collapsed and the Cold War ended, taking with it the threat of nuclear annihilation. The Civil Defense Act of 1950 was repealed in 1994, but not before its authorities were transferred to the Federal Emergency Management Agency.

The transferred authorities appear to have been forgotten until Russia invaded Ukraine in 2022 [9]. At the outset, Russia outmatched Ukraine in every measure of military might and was heavily favored to overrun its neighbor in a matter of weeks. In a classic David versus Goliath encounter, Ukraine bested its larger opponent with better morale, experience and weapons supplied by Western powers. Russian forces attacking on three fronts were halted in their tracks and in some places turned back. Angered by Ukraine's stunning successes, Vladimir Putin threatened to defend Russia with a nuclear attack against the United States. It was the first time in 30 years that the world felt a shiver from the Cold War. The Federal Emergency Management Agency went to the bookshelf and began dusting off its civil defense plans [3].

Civil defense is most certainly a homeland security concern. The results would be catastrophic if Russia were to attack the U.S. homeland with electromagnetic pulse, cyber or nuclear weapons. But how can a homeland security definition be casted that covers terrorism, hazards, cyber attacks and civil defense, along with every other potential threat? The answer is that it is not necessary.

This is because it is effects not threats that concern homeland security. Although homeland security began with the deployment of weapons of mass destruction in the 1995 Tokyo Subway attacks, it was the weapons of mass destruction effects created by subverting the nation's transportation infrastructure that brought homeland security to the forefront of U.S. policy concerns after the 9/11 attacks. Terrorist acts are certainly a homeland security concern. But Hurricane Katrina that was bereft of motive suggests that motive is not an issue. Indeed, the salient characteristics of a homeland security incident are not cause or motive but scale and location – the effects must be catastrophic and they must occur within U.S. territory.

9 Redefining Homeland Security

The proposed definition of homeland security is:

> "Safeguarding the United States from domestic catastrophic destruction."

This definition improves on the 2010 Quadrennial Homeland Security Review definition. It incorporates a qualifying threshold for what constitutes a homeland security incident. It is inclusive of all potential threats that could cause domestic catastrophic destruction without attempting to enumerate them. The definition explicitly states that homeland security encompasses actions to "safeguard," which is a verb, unlike the current definition that only implies actions by using the adjective "safe." Finally, the proposed definition is more concise, just seven words instead of 32, focusing on the primary concern and making it easier to remember.

In order to qualify as a homeland security incident, the effects must be domestic and catastrophic. It is, of course, necessary to specify what constitutes a catastrophic incident. This requires research, and a threshold such as that for mass killing incidents could be determined later. For now, absent a specific threshold, the 9/11 attacks and Hurricane Katrina could be considered to be benchmarks. As tragic as they were, the Orlando and Las Vegas shootings do not measure up to these benchmarks. Perhaps, the U.S. Department of Homeland Security leadership would not have made their statements after the two incidents if the proposed definition had been in place.

Clearly, any threat that could result in domestic catastrophic destruction is a homeland security concern. The 9/11 attacks and Hurricane Katrina certainly qualify. So would cyber attacks, incidents mitigated by civil defense and many more incidents that are yet to be named or conceived. The proposed definition is inclusive not exclusive.

Absolute security is unattainable. Security is a relative state. Risk is the measure of security. Cost is the determinant for risk. Security, therefore, is a dynamic quantity based on changing risk and cost factors. Safeguard is an appropriate term because it implies no specific end state other than continuing action that balances risk against cost. Mitigating actions are conducted over the four phases of disaster management to prevent, protect, respond and recover from threat agents.

The proposed definition is accurate and concise, which make it understandable and memorizable. Its simplicity can make it a unifying force that could guide every element of a massive organization such as the U.S. Department of Homeland Security, as well as the American people, towards a common goal.

10 Conclusions

Homeland security is an important concept whose definition must be widely known, or at least easy to discover and comprehend. The proposed definition is accurate, concise, simple and inclusive. The definition is also useful. It can be a unifying theme for the U.S. Department of Homeland Security. It can set public expectations. It should help shape the U.S. Department of Homeland Security, enabling it to determine priorities and programs, justify budgets and appropriations, set boundaries and enhance capabilities.

References

1. Advanced Law Enforcement Rapid Response Training Center and Federal Bureau of Investigation, Active Shooter Incidents in the United States in 2016 and 2017, Texas State University, San Marcos, Texas and Washington, DC (2018). www.fbi.gov/file-repository/active-shooter-incidents-us-2016-2017.pdf
2. Barnes, J.: Pentagon computer networks attacked, Los Angeles Times (2008)
3. Bowen, A.: Ukrainian Military Performance and Outlook, CRS In Focus IF12150. Congressional Research Service, Washington, DC (2022)
4. Bram, J., Orr, J., Rapaport, C.: Measuring the effects of the September 11 attack on New York City. Econ. Policy Rev. **8**(2), 5–20 (2002)
5. Bush, G.: The Department of Homeland Security. The White House, Washington, DC (2002)
6. DHS Press Office, Statement by Secretary Johnson on the Orlando, Florida attack, Press Release, U.S. Department of Homeland Security, Washington, DC (2016)
7. Electricity Information Sharing and Analysis Center (E-ISAC), TLP: White - Analysis of the Cyber Attack on the Ukrainian Power Grid, Defense Use Case, Washington, DC (2016)
8. Homeland Security Council: National Strategy for Homeland Security. The White House, Washington, DC (2007)
9. Homeland Security National Preparedness Task Force, Civil Defense and Homeland Security: A Short History of National Preparedness Efforts, U.S. Department of Homeland Security, Washington, DC (2006)
10. Kerr, P., Rollins, J., Theohary, C.: The Stuxnet Computer Worm: Harbinger of an Emerging Warfare Capability, CRS Report R41524. Congressional Research Service, Washington, DC (2010)
11. McGuinness, D.: How a cyber attack transformed Estonia, BBC News (2017)
12. National Commission on Terrorist Attacks upon the United States, The 9/11 Commission Report, Washington, DC (2004)
13. Neifert, A.: Case Study: Sarin Poisoning of Subway Passengers in Tokyo, Japan in March 1995. Camber Corporation, Huntsville, Alabama (1999)
14. Office of the Press Secretary, DHS statement on Las Vegas shooting, Press Release, U.S. Department of Homeland Security, Washington, DC (2017)
15. President's Commission on Critical Infrastructure Protection: Critical Foundations: Protecting America's Infrastructures. The White House, Washington, DC (1997)
16. Saturno, J., Heniff, B., Lynch, M.: The Congressional Appropriations Process: An Introduction, CRS Report R42388. Congressional Research Service, Washington, DC (2016)
17. Tatum, S.: U.S. accuses Russia of cyber attacks on power grid, CNN (2018)
18. The White House, PDD-39 - U.S. Policy on Counterterrorism, Washington, DC (1995)
19. The White House, National Strategy for Homeland Security, Washington, DC (2002)
20. The White House, The Federal Response to Hurricane Katrina - Lessons Learned, Washington, DC (2006)
21. U.S. Congress, Public Law 112–265, Investigative Assistance for Violent Crimes Act of 2012, 112th Congress, Washington, DC (2012)
22. U.S. Department of Homeland Security, Quadrennial Homeland Security Review Report: A Strategic Framework for a Secure Homeland, Washington, DC (2010)

23. U.S. Department of Homeland Security, DHS Lexicon, Washington, DC (2023). www.dhs.gov/publication/dhs-lexicon
24. U.S. Department of Homeland Security, Mission, Washington, DC (2023). www.dhs.gov/mission
25. White, R., Bynum, T., Supinski, S. (eds.) Homeland Security: Safeguarding the U.S. from Domestic Catastrophic Destruction, CW Productions, Colorado Springs, Colorado (2016)
26. Zeller, M.: Myth or reality - Does the Aurora vulnerability pose a risk to my generator? In: Proceedings of the Sixty-Fourth Annual Conference for Protective Relay Engineers, pp. 130–136 (2011)

Smart Grid Risks and Impacts

Smart-Grid-Enabled Business Cases and the Consequences of Cyber Attacks

Øyvind Toftegaard[1,2](✉), Doney Abraham[2], Sujeet Shenoi[3], and Bernhard Hämmerli[2,4]

[1] Norwegian Energy Regulatory Authority, Oslo, Norway
oyat@nve.no
[2] Norwegian University of Science and Technology, Gjøvik, Norway
[3] University of Tulsa, Tulsa, OK, USA
[4] Lucerne University of Applied Sciences and Arts, Lucerne, Switzerland

Abstract. The introduction of smart metering systems is a paradigm shift for the power grid. New business cases such as virtual power plants and local flexibility markets are evolving. Security risks and the potential consequences of smart-grid-enabled business cases have been assessed by researchers. However, the research efforts have not ranked the business cases according to their potential disruptive consequences, which makes it difficult to prioritize risk reduction measures.

This chapter describes the results of a survey of market players that sought to rank smart-grid-enabled business cases based on their perceptions of cyber attack consequences. As expected, the consequence perceptions of the market players vary considerably between the business cases. Consequence scenarios suggested by the market players are employed to explain the highest-ranked business cases, which include digital twins, remote access to smart meter circuit breakers, and grid flexibility and balance management. The survey results can support governments and market players in assessing power grid risk and prioritizing risk reduction measures.

Keywords: Smart Grids · Business Cases · Cyber Attacks · Consequences

1 Introduction

Power grids are large and complex systems of systems. Regional grids are connected by transmission lines and national grids are synchronized across borders. Market players such as authorities, grid operators, end-users, vendors and generators must work coherently to ensure safe and reliable grid operation. Digitalization and smart functions are increasingly employed to support and enhance market player interactions.

The U.S. National Institute of Standards and Technology defines a smart grid as "a power network that uses information technology to deliver electricity

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 17–39, 2024.
https://doi.org/10.1007/978-3-031-49585-4_2

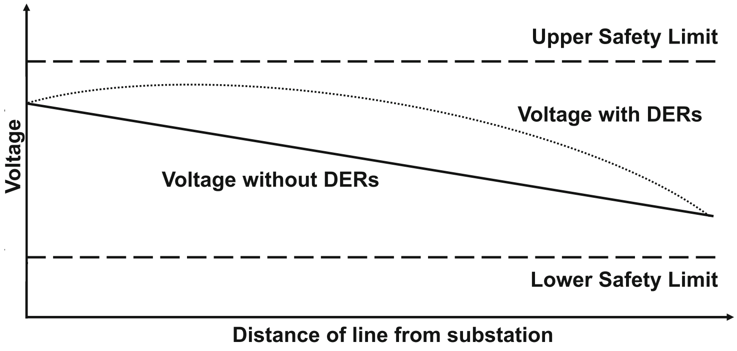


Fig. 1. Voltage properties on a power distribution line [11].

efficiently, reliably and securely" [16]. Future smart grids will need much digitalization to accommodate the shift to green energy in Europe and elsewhere in the world. Because green energy resources are highly decentralized and volatile, digital systems are necessary to balance power production and consumption. The expected increases in digital management and grid complexity will render it more challenging than ever to combat cyber attacks and mitigate their consequences.

This study applies the U.S. Cybersecurity and Infrastructure Security Agency definition of consequences – "[t]he effect of a loss of confidentiality, integrity or availability of information or an information system on an organization's operations, its assets, on individuals, other organizations, or on national interests" [15]. Table 1 summarizes the consequences of key cyber attacks on European power market players as reported by the news media between 2015 and 2022.

The key function of a smart grid is power supply. The grid requires supervision, control and protection equipment to remain operational. Grid protection is provided by protective relays, automatic devices that sense abnormal grid conditions and operate circuit breakers to disconnect faulty portions of the grid.

The most common protective relays are overcurrent, differential, directional and distance (impedance) relays. They differ in their functions, input measurements and triggering abnormalities. While protective relays are adequate in classical power grids, significant complexities to their use are imposed by the smart grid concept. The complexities arise from the large volume of distributed energy resources (DERs) and the need for self-healing [10]. As a result, protective relays that change their settings in real-time are required [13,17–19].

Figure 1 shows a simplified illustration of voltages on a power distribution line with and without the presence of distributed energy resources [11]. Manipulations of distributed energy resources may cause the power line voltage to peak or drop, crossing beyond the safety limits. As a result, protective relays will disconnect certain distributed energy resources or, in the worst case, disconnect the power line itself.

The introduction of smart metering systems is the first step in the realization of smart grids [28]. The next steps will involve artificial intelligence, digital

Table 1. Consequences of cyber attacks on European power market players.

Target	Method	Objective	Consequence
Ukraine Grid Operator (2015)	Black Energy, KillDisk wiper	Power disruption	225,000+ customers lost power for more than two hours
Ukraine Grid Operator (2016)	Crash override, wiper tool	Power disruption	Customers connected to a substation lost power for about one hour
Hydro (Norwegian End-User) (2019)	LockerGoga ransomware	Financial gain	Business systems lost functionality
Elexon (UK Market Platform) (2020)	Revil/Sodinokibi ransomware	Financial gain	Office systems lost functionality and internal data posted on dark web
Volue (Norwegian Tech Supplier) (2021)	Ryuk ransomware	Financial gain	Office systems lost functionality
Nordex Designs (German Wind Turbine Manufacturer) (2022)	Unknown	Unknown	Turbines lost remote monitoring functionality temporarily
Enercon (German Wind Turbine Manufacturer) (2022)	AcidRain wiper	Satellite communications disruption	5,800 turbines lost remote monitoring and control functionality
Rosseti (Russian Power Corporation) (2022)	Ukraine supplier backdoor	E-mobility disruption	Electric vehicle chargers on Moscow/Saint Petersburg motorway disabled
Deutsche Windtechnik (German Wind Farm Operator) (2022)	Unknown	Unknown	Turbines lost remote monitoring functionality for one to two days
Encevo Group (Luxemburg Power Corporation) (2022)	BlackCat/AlphV ransomware	Financial gain	Office systems lost functionality and customer data posted on the dark web

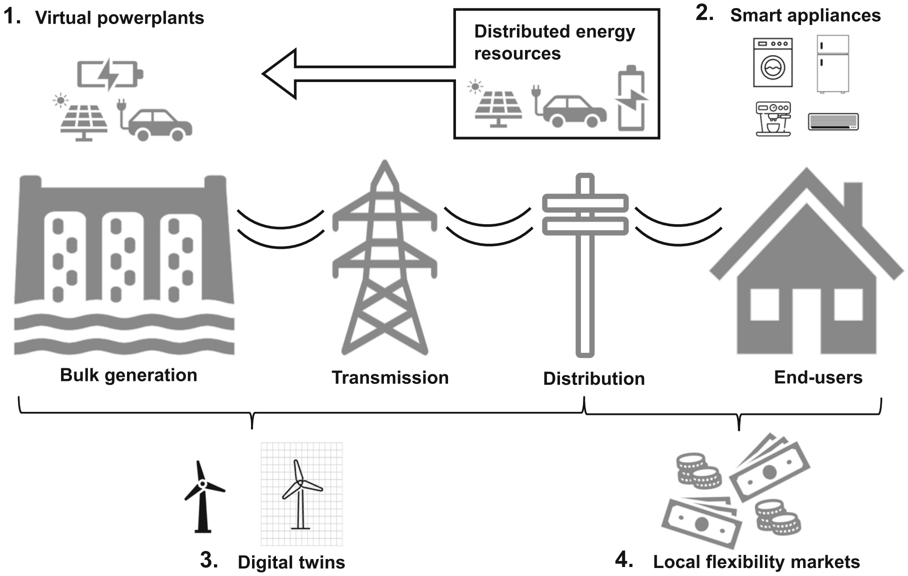


Fig. 2. Smart-grid-enabled business case examples.

twins and other evolving and potentially disruptive technologies. The massive amounts of data collected, communicated and processed by smart grids will propel innovation and many new business cases.

A business case describes perceived business needs that provide services or products. In this study, smart-grid-enabled (SGE) business cases are defined as services or products that utilize the information technology layer of the power network to support smart grid functions.

Figure 2 shows examples of smart-grid-enabled business cases. The business cases include virtual power plants that aggregate distributed energy resources to sell energy in the wholesale market, smart appliances that react automatically to pricing and other management signals to manage power consumption, digital twins that simulate turbine and grid component wear and tear for maintenance planning, and local flexibility markets that leverage end-users to balance distribution grids.

This research has focused on developing a ranking of the perceived consequences of cyber attacks on smart-grid-enabled business cases. Researchers have attempted to evaluate the risks to smart-grid-enabled business cases [1,12,21]. However, their efforts cover the potential consequences of cyber attacks on single business cases or limited sets of business cases. Additionally, since the business cases are not ranked by their consequence levels, it is difficult to determine which business cases should be prioritized for risk reduction investments. This is problematic because security investments may be directed at business cases with low cyber attack consequences instead of business cases that are critical.

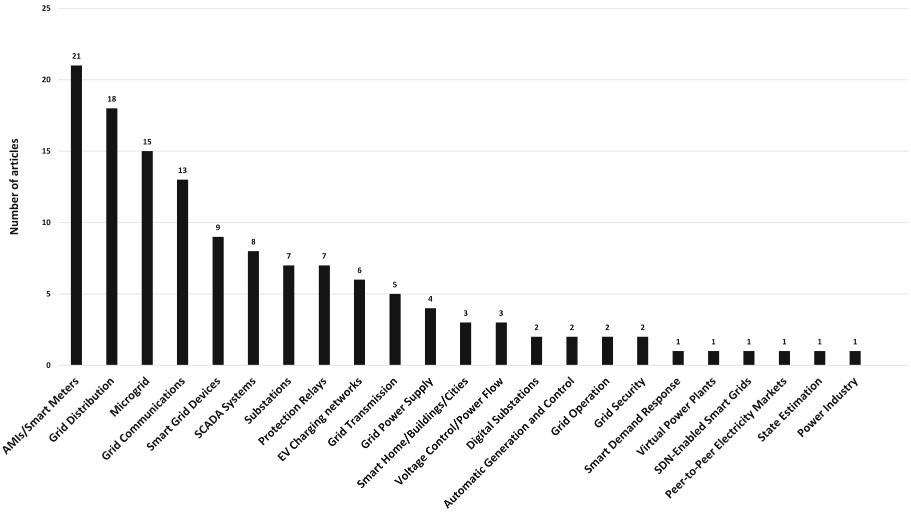


Fig. 3. Research articles in smart-grid-enabled business areas [1].

The focus of this research was to determine the smart-grid-enabled business cases perceived as having the most severe cyber attack consequences. The goal was to rank smart-grid-enabled business cases based on their potential consequence levels. The ranking is vital to entities that own or operate grid infrastructure assets, entities that provide services and authorities that regulate grid security. The ranking would also be a good starting point for researchers pursuing other inquiries such as validating consequence levels through simulation.

2 Previous Work

Several researchers have discussed the consequences of cyber attacks on smart-grid-enabled business cases. However, most of them consider single business cases such as smart meters [2], electrical vehicle (EV) charging [9] or distributed energy resources [11]. Other researchers have analyzed the consequences of cyber attacks on multiple business cases. For example, Li et al. [12] reviewed cyber attack methods on cyber-physical power systems, identifying outages as consequences of cyber attacks on smart substations and financial loss and billing difficulties as consequences of cyber attacks on smart meter systems. Procopiou and Komninos [21] analyzed current and future smart grid threats and their consequences. Their analysis used smart homes as the starting point and included evaluations of load control, demand response and outage management systems.

Abraham et al. [1] have conducted a study of research articles that discuss consequence verification during smart-grid-related risk assessments. Figure 3 shows the distribution of articles by business area. If the most-covered business areas are those with the greatest consequences, the distribution suggests that business cases involving advanced metering infrastructures/smart metering systems, grid distribution and microgrids have the highest consequence levels.

In summary, previous work has focused on the consequences of cyber attacks on smart-grid-enabled business cases. However, the efforts do not rank business cases based on their consequence levels. Additionally, the efforts are relatively narrow in that they focus on single or small sets of business cases.

3 Survey Methodology

The research methodology described in this chapter engaged an interview-based exploratory survey. The research objective was to establish a ranked list of smart-grid-enabled business cases based on their perceived cyber attack consequence levels. The ranking would indicate the business cases that require further consequence assessments to identify high-risk products and services in future smart grids.

Specifically, the research study sought to determine the smart-grid-enabled business cases in Norway with the most severe perceived cyber attack consequences. Norway is one of the most digitalized countries in the world [14]. In 2022, the Norwegian energy mix was 89% hydroelectric and 10% wind [4]. The country achieved 97% smart meter coverage in January 2019 [30]. In 2022, 21% of all operational automobiles and 79% of automobiles sold were electric vehicles [27]. The rapid digitalization and increasing complexity of the Norwegian power grid make it vital to understand the consequences of cyber attacks on smart-grid-enabled business cases.

3.1 Interviews

A total of 22 interviewees from 17 Norwegian power market players were solicited for their perceptions of the potential consequences of cyber attacks on smart-grid-enabled business cases. Nineteen interviews were conducted in Norway between December 2022 and April 2023, each interview lasting between 45 and 60 min. The interviewees comprised 16 males and six females. Two interviewees were in the 20–30 age group, six in the 30–40 age group, five in the 40–50 age group, three in the 50–60 age group and six in the 60–70 age group. All the interviewees, except for the four end-users and three of the five authority employees, had extensive technical backgrounds in cyber security and/or information technology.

Table 2 provides details about the 22 interviewees. The interviewees were drawn from five types of entities, authorities, grid operators, end-users, vendors and generators. The sizes of the entities were determined based on their Norwegian krone revenues converted to their euro equivalents. The Proff Forvalt business information tool [22] was used to obtain annual revenue turnover data. The European Commission definitions of entity sizes [5] were employed based on their annual turnover: micro (up to two million euros), small (above two million

Table 2. Interviewee characteristics.

Market Player	Entity/Role	Size	I	N_I	BC
Authorities	Norwegian Energy Regulatory Authority	Medium	3	3	24
	Norwegian Water Resources and Energy Directorate	Large	1	1	
	Norwegian Data Protection Authority	Medium	1	1	
Grid Operators	Transmission system operator	Large	1	1	21
	Distribution system operator	Large	1	1	
	Distribution system operator	Small	1	1	
	Grid operator association	Small	1	1	
End-Users	Private consumer	N/A	1	1	16
	Private prosumer	N/A	1	1	
	Real estate company	Large	1	2	
	Private end-user association	Medium	1	1	
Vendors	Smart meter vendor	Small	1	2	15
	Grid component/technology vendor	Large	1	1	
	Technology vendor	Large	1	1	
Generators	Hydroelectric and wind power	Large	1	1	13
	Renewables and energy community	Micro	1	2	
	Hydroelectric power	Large	1	1	
			19	22	59

I: Number of interviews, N_I: Number of interviewees, BC: Number of business cases

up to ten million euros), medium (above ten million up to 50 million euros) and large (above 50 million euros).

The interview process relied on the Australian Council for Educational Research (ACER) creative thinking framework [23]. ACER defines creative thinking as “the capacity to generate many different kinds of ideas, manipulate ideas in unusual ways and make unconventional connections in order to outline novel possibilities that have the potential to elegantly meet a given purpose.”

In the context of this research, the novelty of the business cases called for creativity in identifying potential consequences. The ACER creative thinking framework provides three overarching strands comprising various aspects that support creative thinking. The three strands and six aspects shown in Table 3 were used in the interview process.

Table 4 shows an example of a completed survey form.

Consequence ranks and consequence ratings were assigned to assess smart-grid-enabled business cases based on the perceived consequences:

- **Consequence Ranks:** 1 (highest rank), 2, ..., N (lowest rank).
- **Consequence Ratings:** Very High, High, Moderate, Low, Very Low.

Consequence ranking was employed to compare smart-grid-enabled business cases against each other by assigning ranks from 1 to N, where N is the number of business cases. Consequence rating was used to compare smart-grid-enabled business cases using a scale from Very High to Very Low. The advantage gained from using consequence ranks and ratings is that the two methods mutually validate each other.

Table 3. Interview process based on the ACER creative thinking framework [23].

Strand	Aspect	Description
Strand 1		Generation of business cases
	Aspect 1.1	Number of business cases
	Aspect 1.2	Detail levels of business cases
Strand 2		Scenario experimentation
	Aspect 2.1	Perspective shifting
	Aspect 2.2	Scenario manipulation
Strand 3		Ranking and quality control
	Aspect 3.1	Fitting after ranking
	Aspect 3.2	Rank validation through rating

Table 4. Example of a completed survey form.

Business Case	Scenario Description	Consequence Rank (Rating)
Remote access to smart meter circuit breakers	Remote access to large numbers of circuit breakers leading to a massive outage	1 (Very High)
E-mobility and charging services	Remote access to manage charging loads leading to a small outage	2 (High)
... ...	... ...	... (...) ... (...) ... (...)
Direct metering of individual appliances	Disclosure of private consumption data	N (Low)

Cyber attacks compromise the confidentiality, integrity or availability of information and information systems [15]. In turn, the compromises negatively impact an organization's assets, operations and individuals, other organizations or national interests. The interview guide used in the study specified the evaluation of consequences according to the European Union Network and Information Security (NIS) Directive (Article 6, No. 1) [6]. The directive lists six factors that should be considered when determining the significance of a disruptive impact:

- Number of users relying on the business case.
- Dependencies of other sectors on the business case.

- Potential impacts of incidents, in terms of degrees and durations, on economic and societal activities or public safety.
- Market share of the business case.
- Geographic spread with regard to the areas affected by the incidents.
- Importance of the business case for maintaining a sufficient level of service, taking into account the availability of alternative means for providing the service.

The study did not employ a consequence matrix with threshold values, such as for financial loss or blackout duration, for the consequence levels. Instead, interviewees provided ranks and ratings for smart-grid-enabled business cases subjectively based on their perceptions.

3.2 Data Analysis

To enable the data analysis, perceived consequence ratings were given consequence scores S as follows: Very High (S = 5), High (S = 4), Moderate (S = 3), Low (S = 2) and Very Low (S = 1). The individual consequence scores provided by the interviewees were combined to determine the total consequence score for each smart-grid-enabled business case. The computation employed a methodology used to evaluate the evidence strength of identity documents [29]. Specifically, the total consequence score B_j for the j^{th} smart-grid-enabled business case is given by:

$$B_j = S_{1,j} + 2\sum_{i=2}^{N} \frac{S_{i,j}}{2^i} \qquad (1)$$

where $S_{i,j}$ is the score provided by interviewee i for business case j. The convergent series used to compute the total score requires the first score $S_{1,j}$ to have the greatest value and the remaining scores have exponential reductions in their values. For this reason, the individual scores for a business case are ordered from the highest to the lowest values. The first score $S_{1,j}$ is always the highest individual score and the last score $S_{N,j}$ is always the lowest individual score.

The advantage of the methodology is that a single individual outlier score of say, Very High, for a business case would not be valued too highly. Specifically, the business case would not be valued higher than a business case whose individual scores have more consensus. Another advantage is that a large number of low individual scores would prevent the total consequence score from having a high value. Figure 4 demonstrates the convergent function properties for two computations, one (A) with individual ordered scores $1, 1, 1, 1$ and the other (B) with individual ordered scores $3, 2, 2, 2$.

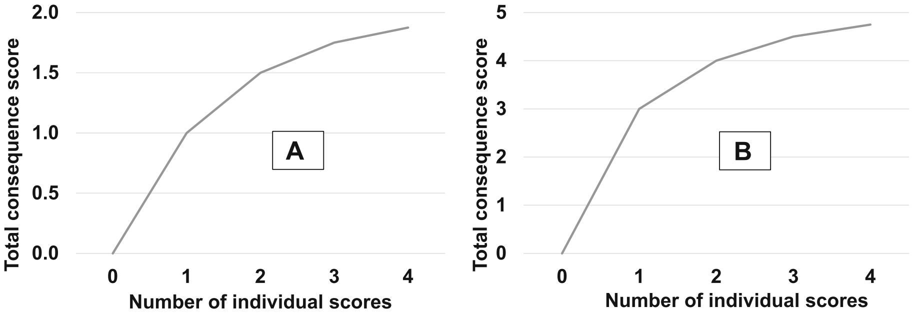


Fig. 4. Convergent function examples.

4 Results

Tables 5, 6, 7 and 8 present the 59 smart-grid-enabled business cases provided by the interviewees ranked by their perceived consequence scores adjusted for consensus. The business cases are ranked based on the combination of the interviewees' perceived consequence ratings and interviewee consensus according to Eq. 1. The smart-grid-enabled business cases with the greatest perceived consequences adjusted for consensus are digital twins, remote access to smart meter circuit breakers, and grid flexibility and balance management.

To understand the consequence rating data in the tables, consider the top-ranked digital twins business case. For this business case, the entry 1 (O1) for the High rating means that one interviewee (1) gave it a High rating and this one interviewee was from a grid operator (O1). Also, the entry 3 (O1, E2) for the Very High rating means that three interviewees (3) gave it Very High ratings, and one of the three interviewees was from a grid operator (O1) and the other two interviewees were end-users (E2).

Table 9 shows the consequence scenarios for the smart-grid-enabled business cases with the top ten ranks in Tables 5, 6, 7 and 8. The top four business cases all have power outage as their main consequence. Privacy and financial consequences are relevant to the fifth-ranked business case. Business cases ranked six through nine have grid instability in their consequence scenarios. National security and financial consequences are relevant to the tenth-ranked business case.

Table 10 shows the smart-grid-enabled business cases with the greatest perceived consequence ranks per interviewee for each market player group. Some of the market players in the same group ranked the same business cases on top. However, none of the top-ranked business cases were identified by two or more market player groups. Thus, there are considerable differences in the perceptions of different market players regarding the business cases with the greatest cyber attack consequences.

Figure 5 shows the smart-grid-enabled business cases with the highest consensus on the consequence ratings. Despite having 22 interviewees, the maximal consensus is four interviewees for one smart-grid-enabled business case. Also,

Table 5. Business cases ranked by perceived consequence ratings while considering consensus.

Rank	Business Case	Consequence Ratings				
		Very Low	Low	Moderate	High	Very High
1	Digital twins				1 (O1)	3 (O1, E2)
2	Remote access to smart meter circuit breakers		1 (A1)	1 (O1)		3 (A1, V2)
3	Grid flexibility and balance management			1 (A1)	1 (G1)	2 (E2)
4	Substation automation (circuit breakers)					3 (A3)
5	Centralized storage of personal data				2 (G2)	1 (A1)
6	SCADA system and sensor communications integration			1 (A1)	1 (E1)	1 (O1)
7	Virtual power plants			1 (G1)	1 (V1)	1 (O1)
8	Battery park management systems		1 (A1)	3 (O1, G2)		1 (A1)
9	System integration and operational technology digitalization					2 (A1, V1)
10	Smart meter consumption data				4 (V2, E2)	
11	Advanced process automation for grid management				1 (O1)	1 (O1)
12	Artificial intelligence and machine learning for optimizing production and maintenance				3 (A1, O2)	
13	E-mobility and charging services				3 (E1, G2)	
14	Substation integration of advanced metering infrastructures and SCADA systems				3 (O1, V2)	
15	Microgrids and energy communities		1 (A1)	1 (V1)	2 (G2)	

A: Authority, O: Grid operator, E: End-user, V: Vendor, G: Generator

Table 6. Business cases ranked by perceived consequence ratings while considering consensus (continued).

Rank	Business Case	Consequence Ratings				
		Very Low	Low	Moderate	High	Very High
16	Central control systems for building management		2 (E2)		2 (G2)	
17	Power grid sensors			1 (O1)		1 (O1)
18	Market platforms		1 (A1)		2 (E2)	
19	Digital components from untrusted parties				2 (E2)	
20	Multiple smart home appliance suppliers				2 (E2)	
21	Integration of production plans in SCADA systems				2 (G2)	
22	Digital management of hydroelectric power plants				2 (A1, G1)	
23	Concurrent consumption tariffs				2 (E2)	
24	Operators of smart home products and home energy management systems		1 (O1)	4 (A1, O3)		
25	Processing of personal customer information			1 (V1)	1 (E1)	
26	Managed service provider operation of renewables			3 (A1, O2)		
27	Applications integrated by aggregators			3 (A1, E2)		
28	Remote management of smart appliances			3 (G3)		
29	Smart meters and smart appliances		2 (E1, G1)	2 (E2)		
30	Digital supply chains for SCADA systems					1 (V1)
31	Aggregator management of power consumption		1 (A1)		1 (V1)	

A: Authority, O: Grid operator, E: End-user, V: Vendor, G: Generator

Table 7. Business cases ranked by perceived consequence ratings while considering consensus (continued).

Rank	Business Case	Consequence Ratings				
		Very Low	Low	Moderate	High	Very High
32	Smart meter/home area network data streams			2 (A2)		
33	Additional data collected by smart meters		2 (V2)	1 (O1)		
34	Information-driven emergency response				1 (O1)	
35	Grid self healing				1 (O1)	
36	Heating appliances that pose fire hazards				1 (E1)	
37	Near real-time algorithm-based grid balancing				1 (O1)	
38	Physical robots on the ground				1 (O1)	
39	Smart appliance performance monitoring				1 (A1)	
40	Additional information-driven processes				1 (A1)	
41	Home area network or price based consumption management		1 (A1)	1 (A1)		
42	Peer-to-peer electricity trading	1 (A1)	2 (V2)			
43	In-home battery or electric vehicle to grid			1 (E1)		
44	Grid frequency stabilization			1 (G1)		
45	Sharing and transportation of grid data and information			1 (O1)		

A: Authority, O: Grid operator, E: End-user, V: Vendor, G: Generator

Table 8. Business cases ranked by perceived consequence ratings while considering consensus (continued).

Rank	Business Case	Consequence Ratings				
		Very Low	Low	Moderate	High	Very High
46	Automated data processing for grid operations			1 (O1)		
47	Transmission line routing optimization			1 (O1)		
48	Wind farms and solar parks			1 (O1)		
49	Prosumers with smart home systems			1 (O1)		
50	Market platforms for local end-user flexibility			1 (A1)		
51	Drones			1 (O1)		
52	Autonomous grid management for businesses			1 (V1)		
53	Direct metering of individual appliances		2 (V2)			
54	Sensors for building energy management systems		2 (G2)			
55	Dynamic electricity supplier transfer based on price		1 (A1)			
56	Solar panels		1 (A1)			
57	Smart energy products and services		1 (E1)			
58	Artificial intelligence driven protective relays		1 (V1)			
59	Information about maintenance and end of use		1 (A1)			

A: Authority, O: Grid operator, E: End-user, V: Vendor, G: Generator

Table 9. Ten business cases with the greatest perceived consequences.

Rank	Business Case	Consequence Scenario
1	Digital twins	Adversaries with access to grid-related digital twins may use them to identify vulnerabilities, optimize damage or disturb operations, leading to large outages. Access to digital twins of building energy management systems can help enable manipulations leading to financial consequences or physical damage
2	Remote access to smart meter circuit breakers	Adversaries may gain remote access to circuit breakers, leading to small to large outages
3	Grid flexibility and balance management	Manipulation or loss of access to management systems controlling large aggregated loads may lead to outages
4	Substation automation (circuit breakers)	Adversaries may gain remote access to circuit breakers leading to injuries or death, grid imbalance or small to large outages
5	Centralized storage of personal data	Assuming future data storage with very high resolution, potential consequences of cyber attacks include privacy breaches, financial consequences or data being used for various nefarious purposes
6	SCADA system and sensor communications integration	Adversaries with access to sensors may inject false data, leading to power disturbances or outages due to bad decision making.
7	Virtual power plants	Cyber attacks on management systems of virtual power plants may lead to grid instabilities or outages
8	Battery park management systems	Adversaries with access to battery park management systems may manipulate or dis-connect loads, leading to grid imbalances or potential battery fires. The worst case is outages, especially if other loads are disconnected simultaneously
9	System integration and operational technology digitalization	Adversaries with access to operational technology environments may manipulate power production or modify or delete data, leading to grid disturbances or outages
10	Smart meter consumption data	End-user consumption data may reveal military preparations or movements, posing threats to national security. Adversaries may also modify consumption data, leading to financial impacts on victims

three interviewees agreed on the same consequence ratings for seven of the business cases.

Figure 6 shows the smart-grid-enabled business cases with the largest spreads in the consensus on consequence ratings. While the remote access to smart meter

Table 10. Business cases ranked with the greatest consequences.

Market Player	Business Case	N_I
Authorities	Substation automation (circuit breakers)	3
	Artificial intelligence and machine learning for optimizing production and maintenance	1
	System integration and operational technology digitalization	1
Grid Operators	SCADA system and sensor communications integration	1
	Virtual power plants	1
	Digital twins	1
	Information-driven emergency response	1
End-Users	Digital components from untrusted parties	2
	Heating appliances that pose fire hazards	1
	Grid flexibility and balance management	2
Vendors	Advanced process automation for grid management	1
	Remote access to smart meter circuit breakers	2
	Digital supply chains for SCADA systems	1
Generators	Integration of production plans in SCADA systems	1
	Digital management of hydroelectric power plants	1
	Microgrids and energy communities	2

N_I: Number of interviewees

circuit breakers business case has the second highest consequence rank in Table 5, it is also one of four business cases with the least consensus.

5 Discussion

The greatest cyber attack consequences were perceived for the digital twins, remote access to smart meter circuit breakers, and grid flexibility and balance management business cases (Table 5). These three business cases are connected to load control scenarios and power outages in the event of compromises (Table 9). In the case of smart metering, the high rank fits well with the survey paper of Abraham et al. [1] (Fig. 3), where smart meters is the business area whose consequences are the most assessed. Furthermore, the sixth rank for the SCADA system and sensor communications integration business case in this study fits well with grid communications ranked fourth by Abraham and colleagues. Similarities are seen when comparing the business case ranks in this study with the numbers of assessments per business area reported by Abraham et al. However, the large number of business cases reported in this study (59) indicates the complexity of smart grids and how challenging it is to identify the business cases with the greatest cyber attack consequences.

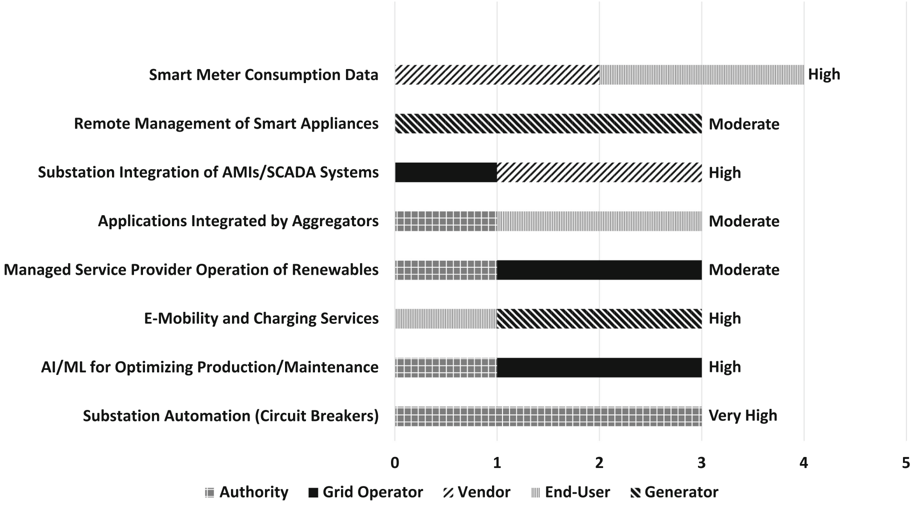


Fig. 5. Perceived consequence/consensus histogram.

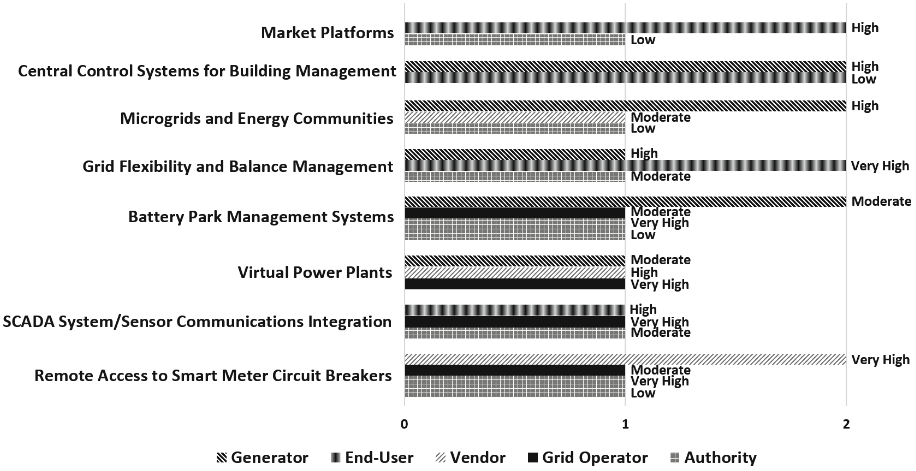


Fig. 6. Perceived consequence/non-consensus histogram.

The differences in the business cases reported with the greatest perceived consequences in Table 10 reveal how differently market players as well as individuals perceive smart grid consequences. The fact that none of the interviewees from all the market player groups gave the top rank to the same smart-grid-enabled business case suggests that the complexity of smart grids makes it challenging to anticipate potential consequences.

Figures 5 and 6 demonstrate little general consensus on the consequence levels of smart-grid-enabled business cases. The most consensus was observed in the smart meter consumption data business case, but the consensus is small, just four of the 22 interviewees. The least consensus was seen in the battery park management system and remote access to smart meter circuit breakers business cases, whose perceived cyber attack consequences ranged from Very High to Low. The reason may be that the potential consequences depend on how business cases are implemented. An example is if smart meters were to have a capacity limit beyond which on-board circuit breakers should not be installed. In this case, large industrial, healthcare and public service facilities would not be impacted as much by cyber attacks on smart meters as small residential buildings. It is not known whether or not the interviewees were aware of and applied such details when they evaluated the cyber attack consequences. Additionally, limited knowledge about a new grid technology such as battery parks likely made it difficult for the interviewees to evaluate their disruptive potential.

Clearly, the differences in perceived consequences point to additional research to verify the findings of this study. Such verification could require going beyond interview-based surveys and performing real-world analyses.

6 Study Validity

This section discusses the threats to the validity of this study, which include critical realism, risk perception, and internal and external validity.

6.1 Critical Realism and Risk Perception

Human perception is known to be influenced by knowledge and experience. Critical realism theory distinguishes between the perceived empirical domain and hidden mechanisms [3]. According to critical realism, unobservable mechanisms cause observable events. These hidden mechanisms exist independent of human perceptions. Figure 7 shows how the perceived empirical domain and hidden mechanisms together constitute the real domain.

According to critical realism theory, interviewees' perceptions are colored by their personal theories, knowledge and understanding. Therefore, the interviewees' responses would not reflect the "real" domain, but their perceptions. When applying critical realism, the real consequences of cyber attacks can be understood only if the underlaying structures that generate each consequence are understood. This is problematic because smart-grid-enabled business cases can be described as complex socio-technical systems, implying that the structures that generate the consequences would be highly complex.

Perceptions of consequence scenarios and their severity levels are based on subjective observations and experiences. Therefore, the perceived consequences do not necessarily reflect the real consequence levels, but are rather the result of best efforts. However, subjective perceptions provide useful indications in risk assessments and are good starting points for further research.

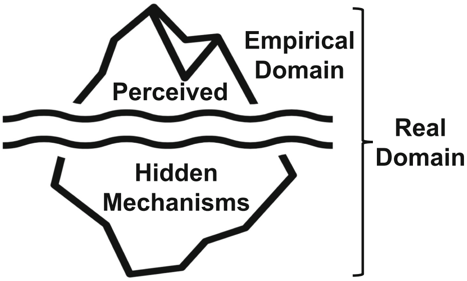


Fig. 7. Critical realism iceberg (based on [3]).

6.2 Internal and External Validity

Internal validity considers the extent to which observed results represent the truth in a study population and are, therefore, not due to methodological errors [20]. To ensure that the interviewees fully understood the task of ranking their proposed business cases from highest to lowest, a numerical rank (1, 2, ..., N) and a categorical rating (Very High to Very Low) were applied. This approach enabled the interviewer and interviewees to identify when logical failures occurred. Examples included a business case ranked 1 and rated High and another business case ranked 2 and rated Very High. During such situations, which occurred multiple times, the interviewees were asked to reconsider their responses. This method of securing interviewee understanding of the task strengthened the internal validity. A threat to the internal validity of this study is that only one to three individuals were interviewed per market player entity. Therefore, uncertainty exists whether or not the opinion of an interviewee's entity as a whole would be the same as that of the interviewee.

External validity considers the extent to which results from a study may be generalized [24]. A threat to the validity of this study is that it only sought perceived consequence rankings in the empirical domain. Thus, the results are influenced by the backgrounds of the interviewees and do not necessarily reflect the perceptions of others. Furthermore, because the only results are perceptions, it is unknown how well the results capture the real world. Therefore, the consequence rates obtained in this study need to be verified, perhaps through simulation.

Figure 8 illustrates how internal validity belongs to the empirical domain whereas external validity belongs to the real domain. Similar to the critical realism iceberg, the truth of the interviewees in this study is limited to their perceptions. The truth in the real domain is constructed by mechanisms, some of which would be invisible to the interviewees.

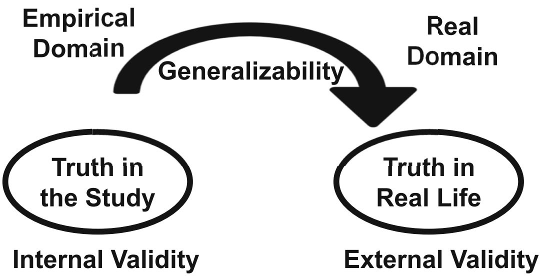


Fig. 8. Internal and external validity (based on [20]).

7 Empirical to Real Consequences

The results derived in the empirical domain are based on the perceived consequences of the interviewees and are, therefore, their personal opinions. However, because some real consequences are not visible as part of the perceived consequences, insights into the real consequences may be gained by conducting an investigation in a near-real (laboratory) domain.

An electric vehicle may be employed as a use case to explore the real consequences. Electric vehicle charging should be performed at intervals so as not to affect other grid operations and reduce the peak load time. Shafiq et al. [26] have shown that a load loss of 6.89% occurs when charging is done at irregular intervals. Fernandez et al. [7] have provided an assessment of the impacts of electric vehicles on the power distribution network. Their suggested strategy is based on a large-scale distribution planning model used to investigate two real power distribution areas. The first was a residential area with 6,000 end-users and 3,676 vehicles and the second was a commercial area with 61,000 end-users and 28,626 vehicles. Simulations of the effects of electric vehicle loads on system losses demonstrated that the loss could increase up to 40%, considering that electric vehicles accounted for 62% of the total number of vehicles.

Sayed et al. [25] studied an electric vehicle charging attack and its impacts on power grid operation. Their case study, which accounted for electric vehicle locations and loads, involved a simulated attack under various scenarios involving the ESCC9-bus system and 7-bus test case [8]. Attack simulations involving residential facility loads and electric vehicle loads demonstrated that electric vehicle loads had greater consequences on grid operations. Sayed and colleagues also showed how adversaries could estimate the grid topology and create targeted attacks that maximized negative impacts on the grid.

Weiss [31] has warned that frequency manipulations in the power grid could lead to catastrophic disruptive events as in the case of the celebrated Aurora generator test conducted in 2007 at Idaho National Laboratory. During the test, researchers caused a generator to catch fire by manipulating the power frequency. Power frequency is a measure of the amount of energy injected into the grid; the greater the energy injected, the higher the frequency. When the frequency deviates from the accepted standards, clocks may show the wrong time and electric equipment can be destroyed. Therefore, it is essential to use near-real-domain experiments to investigate and evaluate the consequences in order to mitigate future attacks.

8 Conclusions

The research study of market players has sought to rank smart-grid-enabled business cases based on their qualitative perceptions of cyber attack consequences. The results reveal that the business cases with the greatest perceived cyber consequences are digital twins, remote access to smart meter circuit breakers, and grid flexibility and balance management. Although it was possible to identify the business cases with the greatest perceived consequences, little consistency was observed in the rankings by individuals and groups of market players. The principal reason for the inconsistent rankings appears to be the complexity of smart grids and smart-grid-enabled business cases.

The study results can support governments and market players in assessing power grid risk and prioritizing risk reduction measures. The results would also be useful to policymakers in defining the scope of smart grid cyber security legislation and regulations, and to researchers who wish to move the study results from the empirical domain to real-world applications and verification through simulation.

Acknowledgments. This research was supported by the Norwegian Energy Regulatory Authority and the Norwegian Water Resources and Energy Directorate. The authors also wish to thank Jon-Martin Storm, Janne Hagen and Silje Aakre for their assistance in this research.

References

1. Abraham, D., Toftegaard, Ø., Gebremedhin, A., Yayilgan, S.: Consequence verification during risk assessments of smart grids. In: Staggs, J., Shenoi, S. (eds.) Critical Infrastructure Protection XVII, pp. 39–61. Springer, Cham (2024)
2. Anderson, R., Fuloria, S.: Who controls the off switch? In: Proceedings of the First IEEE International Conference on Smart Grid Communications, pp. 96–101 (2010)
3. Bhaskar, R.: A Realist Theory of Science. Routledge, New York (2008)
4. Energy Facts Norway, Electricity Production, Oslo, Norway (2023). www.energifaktanorge.no/en/norsk-energiforsyning/kraftproduksjon
5. European Commission, SME Definition, Brussels, Belgium (2023). single-market-economy.ec.europa.eu/smes/sme-definition_en
6. European Parliament and the Council of the European Union, Directive (EU) 2016/1148 of the European Parliament and of the Council of 6: Concerning Measures for a High Common Level of Security of Network and Information Systems Across the Union, Document 32016L1148, Belgium, Brussels (2016)
7. Fernandez, L., Román, T.S., Cossent, R., Domingo, C., Frias, P.: Assessment of the impact of plug-in electric vehicles on distribution networks. IEEE Trans. Power Syst. **26**(1), 206–213 (2011)

8. Glover, J., Overbye, T., Sarma, M.: Power System Analysis and Design, SI Cengage Learning, Boston (2017)
9. Gumrukcu, E., et al.: Impact of cyber-attacks on EV charging coordination: the case of a single point of failure. In: Proceedings of the Fourth Global Power, Energy and Communications Conference, pp. 506–511 (2022)
10. Khalid, H., Shobole, A.: Existing developments in adaptive smart grid protection: a review. Electr. Power Syst. Res. **191**, Article no. 106901 (2021)
11. Langer, L., Smith, P., Hutle, M.: Smart grid cybersecurity risk assessment. In: Proceedings of the International Symposium on Smart Electric Distribution Systems and Technologies, pp. 475–482 (2015)
12. Li, F., Yan, X., Xie, Y., Sang, Z., Yuan, X.: A review of cyber-attack methods in cyber-physical power systems. In: Proceedings of the Eighth IEEE International Conference on Advanced Power System Automation and Protection, pp. 1335–1339 (2019)
13. Liu, Z., Hoidalen, H.: A simple multiagent system based adaptive relay setting strategy for a distribution system with wind generation integration. In: Proceedings of the Thirteenth International Conference on Developments in Power System Protection (2016)
14. Mattila, J., et al.: Digibarometer 2022: A Digital Green Transition (in Finnish). Taloustieto Oy, Helsinki, Finland (2022)
15. National Initiative for Cybersecurity Careers and Studies, Vocabulary, Cybersecurity and Infrastructure Security Agency, Washington, DC (2023). niccs.cisa.gov/cybersecurity-career-resources/glossary#C
16. National Institute of Standards and Technology, Smart Grid: A Beginner's Guide, Gaithersburg, Maryland (2019). www.nist.gov/el/smart-grid/about-smart-grid/smart-grid-beginners-guide
17. Pandakov, K., Hoidalen, H.: Distance protection with fault impedance compensation for distribution network with DG. In: Proceedings of the IEEE Power and Energy Society Innovative Smart Grid Technologies Conference Europe (2017)
18. Pandakov, K., Hoidalen, H., Marvik, J.: Implementation of distance relaying in distribution networks with distributed generation. In: Proceedings of the Thirteenth International Conference on Developments in Power System Protection (2016)
19. Pandakov, K., Hoidalen, H., Marvik, J.: Fast protection against islanding and unwanted tripping of distributed generation caused by ground faults. In: Proceedings of the Twenty-Fourth International Conference on Electricity Distribution, Paper No. 0716 (2017)
20. Patino, C., Ferreira, J.: Internal and external validity: can you apply research study results to your patients? J. Bras. Pneumol. **44**(3), 183 (2018)
21. Procopiou, A., Komninos, N.: Current and future threats framework in the smart grid domain. In: Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, pp. 1852–1857 (2015)
22. Proff, Proff Forvalt Credit and Market Tools (in Norwegian), Oslo, Norway (Forvalt.no) (2023)
23. Ramalingam, D., Anderson, P., Duckworth, D., Scoular, C., Heard, J.: Creative Thinking: Definition and Structure. Australian Council for Educational Research, Camberwell, Australia (2020)
24. Runeson, P., Host, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131–164 (2009)
25. Sayed, M., Atallah, R., Assi, C., Debbabi, M.: Electric vehicle attack impact on power grid operation. Int. J. Electr. Power Energy Syst. **137**, Article no. 107784 (2022)

26. Shafiq, S., Irshad, U., Al-Muhaini, M., Djokic, S., Akram, U.: Reliability evaluation of composite power systems: evaluating the impact of full and plug-in hybrid electric vehicles. IEEE Access **8**, 114305–114314 (2020)
27. Statistics Norway, Four of five new cars in 2022 were EVs (in Norwegian), Oslo, Norway, 24 March 2023. www.ssb.no/transport-og-reiseliv/landtransport/statistikk/bilparken/artikler/fire-av-fem-nye-biler-i-2022-var-elbiler
28. Toftegaard, Ø., Hagen, J., Hämmerli, B.: Are European security policies ready for advanced metering systems with cloud backends? In: Staggs, J., Shenoi, S. (eds.) Critical Infrastructure Protection XVI, pp. 47–69. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20137-0_2
29. Toftegaard, Ø., Yang, B.: Towards an operable evaluation system for evidence of identity. In: Proceedings of the Forty-Second IEEE International Computer Software and Applications, vol. 2, pp. 565–570 (2018)
30. Venjum, A.: Smart Meters (AMS) (in Norwegian), Report No. 24/2019, Norwegian Water Resources and Energy Directorate, Oslo, Norway (2019). publikasjoner.nve.no/rapport/2019/rapport2019_24.pdf
31. Weiss, J.: Aurora generator test. In: Radvanovsky, R., Brodsky, J. (eds.) Handbook of SCADA/Control Systems Security, pp. 107–114. Routledge, New York (2016)

Consequence Verification During Risk Assessments of Smart Grids

Doney Abraham[1](✉), Øyvind Toftegaard[1,2], Alemayehu Gebremedhin[1], and Sule Yayilgan[1]

[1] Norwegian University of Science and Technology, Gjøvik, Norway
doney.abraham@ntnu.no
[2] Norwegian Energy Regulatory Authority, Oslo, Norway

Abstract. The transformation of conventional power grids to smart grids over the past decade has led to increased exposure to cyber attacks. Understanding the impacts of cyber attacks is essential to selecting appropriate mitigation strategies.

This research examines the evolution in the understanding of the consequences of cyber attacks on smart grids. It has explored the literature on consequence verification during risk assessments of smart grids from 2009 to 2023. A total of 839 articles were collected. After filtering duplicate and irrelevant articles, deep content analysis yielded 125 articles that assessed cyber risks to smart grids, with 67 of them also focusing on real consequence verification. Further study identified 23 smart-grid-enabled business areas impacted by cyber risks and six methods for verifying the real consequences of cyber attacks on smart grids. Real consequence verification is important because it helps identify the most critical smart grid vulnerabilities and prioritizes efforts for mitigating cyber attacks and their negative impacts.

Keywords: Smart Grids · Cyber Attacks · Risk Assessment · Consequence Verification

1 Introduction

Conventional electrical grids have centralized power plants that provide energy to consumers with limited governance and consumption monitoring [96]. Information and electricity flows in conventional grids are unidirectional and the grids lack self-restoration capabilities after outages [34]. These deficiencies have led to the transformation of conventional power grids to smart grids.

Smart grids incorporate cyber and physical systems in power networks to support the efficient generation, transmission and distribution of electricity [139]. Smart grids employ industrial control systems that leverage information and communications technology to control physical processes [66]. Cyber attacks on industrial control systems have severe consequences because they target the vital physical systems being monitored and controlled [14]. Cyber attacks on smart grids are a grave concern [46]. The most serious example is the December 2015

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 40–61, 2024.
https://doi.org/10.1007/978-3-031-49585-4_3

cyber attack on the Ukrainian power grid, which caused approximately 225,000 consumers to lose power [31].

It is impossible to defend against every cyber attack on a smart grid because new vulnerabilities constantly emerge and attackers continually find new ways to exploit them [6]. However, risk analysis can play an essential role in preventing cyber attacks by identifying potential vulnerabilities and threats, and determining their likelihood and potential impacts [49]. Impact assessments determine the potential consequences of successful cyber attacks in terms of disruption of critical services, financial losses and reputational damage [70], enabling smart grid entities to prioritize their security efforts and allocate resources to address the most critical risks.

This research has attempted to examine the validity of common methods for verifying the consequences of cyber attacks on smart grids. The effort focused on the research literature on consequence verification during risk assessments of smart grids from 2009 to 2023. A total of 839 articles representing the state of the art were reviewed. After filtering duplicate and irrelevant articles, 155 were subjected to in-depth analysis, which eliminated 30 of the articles. The investigation determined that 120 of the remaining 125 articles studied the impacts of assessed risks, with 67 of them also focusing on real consequence verification. The results provide an understanding of the methods for verifying the consequences of cyber attacks on smart grids and the degree to which real consequences can be verified.

2 Smart Grids and Cyber Attacks

This section discusses smart grids, threats and vulnerabilities, and cyber attacks on smart grids and their consequences.

2.1 Smart Grids

In the European context, a smart grid is an electricity grid that intelligently manages the behaviors and activities of all users linked to the grid [141]. This feature enables a smart grid to deliver power more efficiently than a conventional grid while responding to diverse circumstances and events across the grid. The circumstances and events pertain to power generation, transmission, distribution and consumption [34].

The U.S. National Institute of Standards and Technology (NIST) defines a smart grid as an electric power system that uses information, two-way cyber-secure communications technologies, and computational intelligence in an integrated fashion across the spectrum of an energy system from generation to consumption [41]. NIST also specifies a conceptual smart grid model comprising seven domains: power generation, transmission, distribution, consumers, service providers, operations and markets, all of which interact with each other in real time [139].

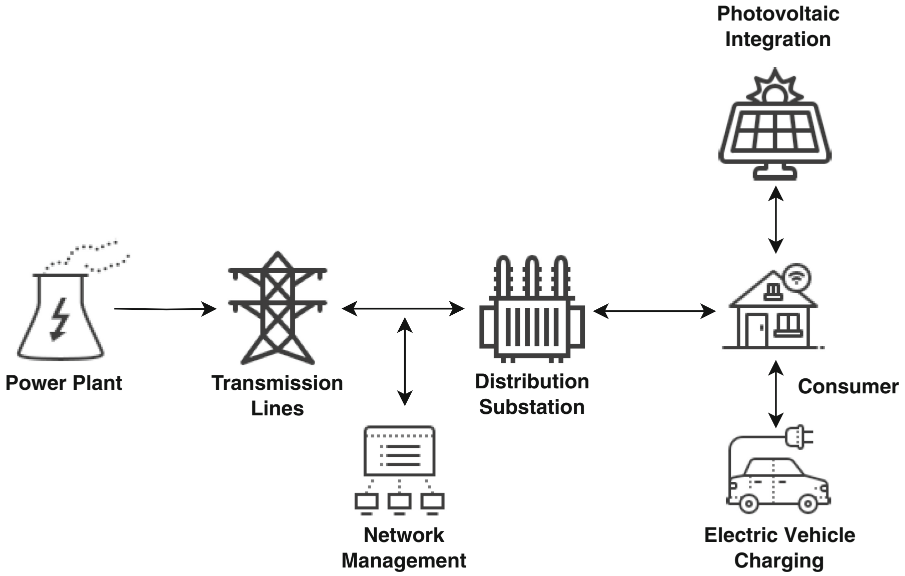


Fig. 1. Conceptual smart grid model (adapted from [131]).

Figure 1 shows a conceptual model of a smart grid. The introduction of various domains and enhancements increases the complexity of a smart grid and renders it vulnerable to myriad attacks.

2.2 Threats and Vulnerabilities

A threat is a potential adverse event or action that has the potential to cause harm or damage to an individual or organization. In cyber security, threats are possible malicious actions that compromise the confidentiality, integrity or availability of information systems and the data they process [56].

A vulnerability is a weakness in a system or process that may be exploited by a threat. The weaknesses may exist in software, hardware or organizational processes. For example, a vulnerability in a software application can be leveraged by an adversary to gain unauthorized access to sensitive information [56].

The principal smart grid attacks, threats and vulnerabilities include [6,42, 47,65]:

- **Cyber Attacks:** Smart grids are highly dependent on computer systems, networks and communications systems, which makes them vulnerable to cyber attacks such as malware, ransomware and denial-of-service attacks.
- **Advanced Persistent Threat:** The advanced persistent threat includes targeted, persistent and sophisticated cyber attacks that are designed to steal sensitive information or disrupt grid operations.

- **Insider Threat:** Smart grid employees and contractors with grid access can introduce malicious software or disrupt operations intentionally or unintentionally.
- **Physical Attacks:** Smart grid assets such as power plants and substations are vulnerable to vandalism and sabotage attacks.
- **Supply Chain Threat:** Smart grid systems often rely on third-party vendors for software and hardware components that have vulnerabilities that can be exploited.
- **Aging Infrastructure:** Smart grid systems and infrastructure are susceptible to malfunctions and failures due to their age.
- **Lack of Security Standards:** Smart grid systems are constantly evolving in their technologies, designs, implementations and operations. Security standards and best practices for vendors and operators may not be followed or may not exist.
- **Interoperability:** Smart grids are complex systems involving multiple vendors, communications protocols and technologies. Attackers can exploit vulnerabilities that arise from the need to achieve system interoperability.

2.3 Cyber Attack Consequences

The consequences of cyber attacks on a smart grid are severe and wide-ranging. Toftegaard et al. [122] list prominent cyber attacks on European power sector assets over the past eight years. The most serious consequences were caused by the December 2015 cyber attack on the Ukrainian power grid, which cut power to approximately 225,000 consumers [31].

Ding et al. [25] present a review of cyber attacks on smart grids from 2010 through 2022. Their review describes attack targets, methods and consequences. Ding and colleagues note that cyber attacks that exploit smart grid vulnerabilities are responsible for the most serious consequences.

Researchers have shown that large blackouts are often the result of cascading failures [44,118]. One of the largest blackouts in European history occurred on November 4, 2006. A single incident originating in Northern Germany led to power supply disruptions at more than 15 million European homes [82]. Su et al. [118] posit that a coordinated software-based attack would have greater negative impacts than physical sabotage.

Although power disruptions are the most common consequences of cyber attacks on smart grids, the information-driven processes in smart grids provide myriad attack opportunities with negative consequences. For example, ransomware attacks do not need to disrupt power supply to be successful; instead, they may cripple maintenance and invoicing functions at a utility, resulting in significant economic losses. Smart grids are also susceptible to privacy breaches of customer credit card information, personal information and detailed customer consumption data that may reveal in-home activities.

The U.S. Cybersecurity and Infrastructure Security Agency (CISA) defines consequence as "the effect of a loss of confidentiality, integrity or availability of information or an information system on an organization's operations, its assets, on individuals, other organizations, or on national interests" [92]. The consequences involve compromises to any or all of the three main security properties, confidentiality, availability and integrity. Because of the great variance in consequences of cyber attacks on smart grids and their potential severity, it is essential to have a deep understanding of the mechanisms that lead to negative consequences.

3 Related Work

Several literature reviews and surveys have focused on cyber security, cyber attacks, threats, impacts and defenses related to smart grids. He and Yan [55] discuss the security challenges facing smart grids. They detail the critical threats, attack schemes and defensive solutions involving protection, detection and mitigation. Attack schemes highlighted include transmission system attacks, distribution system attacks and electricity market attacks. Their work is intended to raise awareness and inspire research efforts focused on developing secure and resilient cyber-physical infrastructures.

Mrabet et al. [87] survey smart grid security challenges and review various attack schemes and defensive strategies. They note that smart grid security efforts tend to focus on confidentiality, integrity and availability, but have yet to consider accountability. They propose a three-step cyber security strategy covering the pre-attack, under-attack and post-attack phases. They review security requirements, describe several severe cyber attacks and classify attacks as focusing on reconnaissance, scanning, exploitation and maintaining access. They also recommend detection techniques and countermeasures, including network security, data security, device security, attack detection and mitigation, and digital forensics.

Ding et al. [25] describe cyber threats that impact the security of smart grid ecosystems. They consider intrinsic system vulnerabilities and external cyber attacks, and analyze the vulnerabilities of smart grid components, including hardware, software, data communications and data management systems. They also present a structured smart grid architecture and a global review of cyber attacks on smart grids between 2010 and 2022.

Gunduz and Das [47] present a comprehensive survey of cyber security issues related to Internet-of-Things-based smart grid applications and proposed solutions. They also analyze various types of cyber attacks, network vulnerabilities, attack countermeasures and security requirements.

Smadi et al. [115] discuss the importance of employing smart grid testbeds to analyze power systems. They provide a comprehensive overview of cyber-physical smart grid testbeds, including their architectures, functional analyses, main vulnerabilities and threats, testbed requirements, constraints and applications. They also highlight the use of simulation testbeds, physical testbeds, interconnectivity of testbeds at multiple locations and integration of software-defined networking (SDN) technology.

Table 1. Summary of research methodology.

Research Questions	Are common methods for verifying the real consequences of cyber attacks on smart grids valid? (a) What are the most common methods for verifying the consequences of cyber attacks on smart grid components? (b) To what degree are common methods capable of revealing real consequences?
Survey Period	January 1, 2009 to December 31, 2022 (13 years)
Databases	Scopus, Web of Science, IEEE Xplore, ScienceDirect
Search Criteria	Keywords in article title or abstract
Search Keywords	(Risk Analysis OR Risk Assessment) AND (Consequence OR Impact)) AND (Smart Grid OR AMI OR Smart Meter OR Home Energy Management System OR HEMS OR Building Management System OR BMS OR Flexibility Market OR Energy Community OR Microgrid OR Peer-to-Peer Trading OR Grid Self Healing OR Substation Automation OR Virtual Power Plants OR Aggregator Service OR E-Mobility)
Inclusion Criteria	Smart Grid Cyber Attack Consequence Verification
Exclusion Criteria	Cyber Attack Simulation without Consequence Verification, Review, Survey

4 Research Methodology

This research involved a systematic review of the methods used to verify the consequences of cyber attacks on smart grids and their validity. Table 1 summarizes the research methodology, including the research questions, survey period for the research literature, databases containing the research literature, search criteria, search keywords and search inclusion and exclusion criteria.

Prominent English research article databases, Scopus, Web of Science, IEEE Xplore and ScienceDirect, that cover the technical areas of interest were selected for the research. Literature reviews and surveys in the field were analyzed to produce an appropriate list of keywords for searching article titles and abstracts. Table 1 shows the search criteria and keywords as well as the search inclusion and exclusion criteria.

The articles were restricted to those published from January 1, 2009 to December 31, 2022. This was because the research sought to focus on the European Union (EU) market and the key starting point was the important 2009 EU Electricity Market Directive (2009/72/EC) that established standards and rules for the European electricity market [33]. The articles were also screened to eliminate duplicates culled from the databases. The database search results comprised 839 articles.

The next important step involved reading the article titles and abstracts and eliminating irrelevant articles. Review and survey articles were also excluded. Articles were included in the pool if their abstracts lacked details to make accurate selections. The final survey pool included 155 articles.

Each of the articles in the survey pool was read carefully to answer the following questions:

- Is a risk assessment performed?
- Are risk consequences studied?
- What business areas are impacted by the identified risks?
- What techniques are used to verify the consequences of cyber attacks on smart grids?
- To what degree is the consequence verification method capable of revealing real consequences?

The answers for each article were documented in a Microsoft Excel file and categorized by database to structure the data for further investigation. Each answer was recorded as a yes or no, along with relevant keywords and comments. The study took approximately three months, from querying the databases, culling articles, and reading and recording data about all the articles in the survey pool.

The questions related to several articles were answered differently by different readers. For example, one reader assessed an article as verifying consequences whereas another reader assessed it as a theoretical study that did not verify real consequences. These articles were examined by the co-authors of this chapter and their comments were compared to obtain consensus answers to the specific questions.

5 Results

Detailed analysis of the 155 articles in the survey pool revealed that 30 articles were not relevant. Thirteen of the 30 articles investigated the impacts of policy decisions or recommendations, or market reform policy related to smart grids. Three other articles focused on the financial profitability of implementing microgrids or smart grids and two articles analyzed the criticality of cyber-physical infrastructure risks to society. Twelve other articles did not apply specifically to the research. These articles examined risk optimization in the electricity sector, monitoring in smart cities, general cyber risk analyses in other critical infrastructure sectors, intrusion detection systems, microgrid design performance, and conceptual models for representing and tracking compliance based on security standards, among other topics.

Figure 2 presents an overview of the research results. The 125 articles in the survey pool focused on risk analyses of smart grids. Of these articles, 120 (96%) also studied the consequences of the assessed risks.

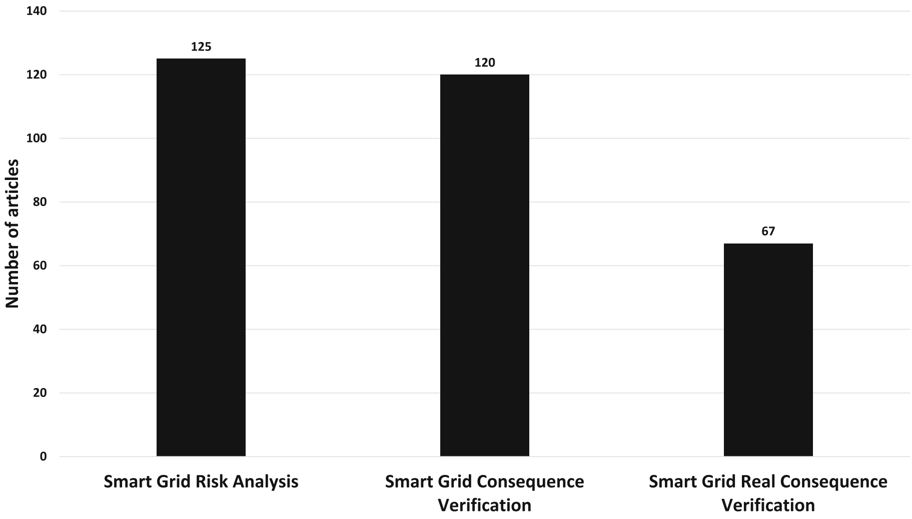


Fig. 2. Overview of research results.

Real consequence verification of cyber attacks on smart grids is the process of assessing the real physical impacts. This involves evaluating potential equipment damage, power outages, economic impacts, privacy impacts to consumers and overall disruptions in electricity delivery. Real consequence verification is important because it helps identify the most critical grid vulnerabilities and prioritize efforts for mitigating potential cyber attacks and their negative impacts. The analysis indicated that 67 (53.6%) of the 125 articles investigated this important issue to some extent. The fairly large percentage (46.4%) of articles that completely ignored consequence verification demonstrates a key gap in the research literature.

Figure 3 shows the 23 significant business areas impacted by the risks identified in the research articles. The most impacted business area in terms of article coverage is advanced metering infrastructures/smart meters with 21 articles, followed by grid distribution, microgrids and grid communications with 18, 15 and 13 articles, respectively.

Table 2 lists the 23 business areas along with the specific articles in the survey pool that cover their risks.

Figure 4 shows the six principal methods for verifying the real consequences of the identified risks to smart grids. The most widely researched method in the literature involves the use of IEEE test systems (22 articles). These test systems, which are commonly employed in power system analysis research [99], simulate power system behavior under conditions such as power flow, voltage stability and transient stability. They enable researchers to analyze and understand the behavior of power systems and their components in a variety of scenarios. They also offer simplified representations of real power systems for testing and validating power system analysis techniques and algorithms.

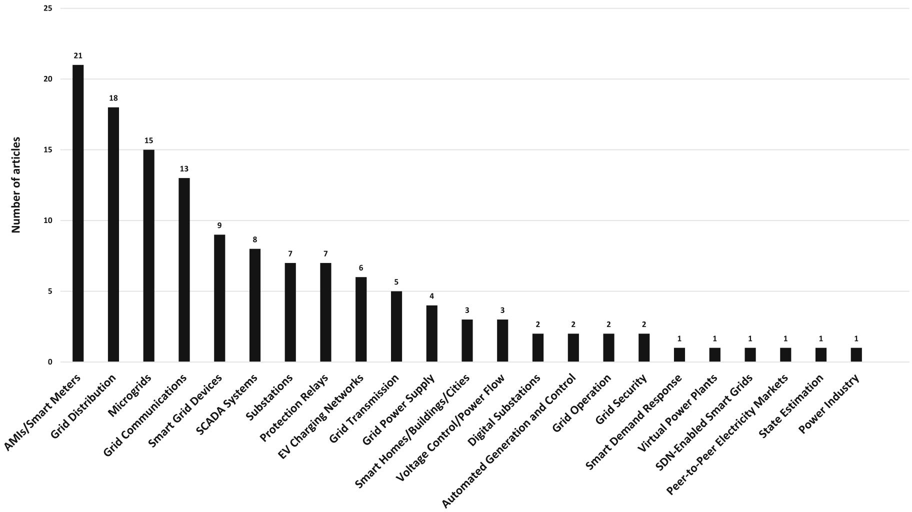


Fig. 3. Overview of impacted business areas.

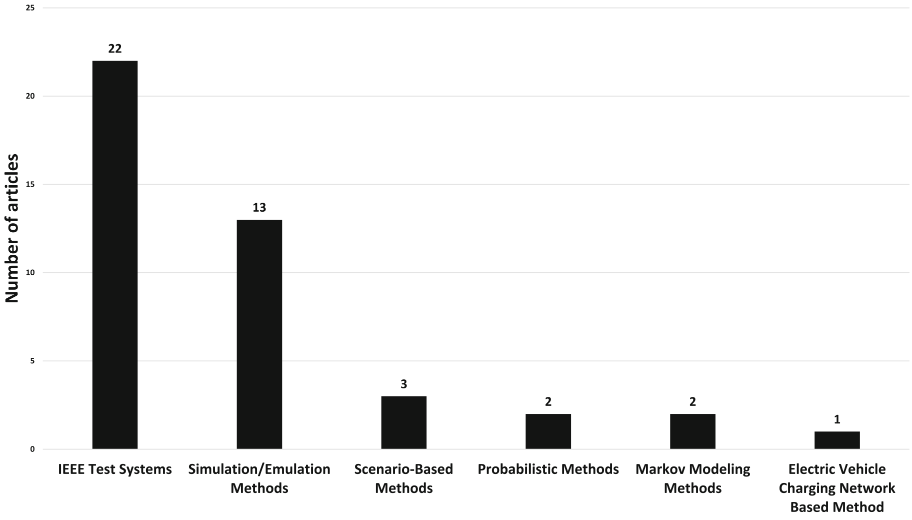


Fig. 4. Methods for verifying real consequences.

The IEEE test systems employ 9-bus, 14-bus, 30-bus and 33-bus models, among others. The IEEE 9-bus system comprises three synchronous generators, nine buses, six transmission lines, three transformers and three real/reactive power loads. The IEEE 14-bus bus system is a simple approximation of the American electric power system with 14 buses, five generators and 11 loads. The IEEE 30-bus system, with 30 buses, 41 branches and six generators, is extensively used for power system analysis. The IEEE 33-bus system, which

Table 2. Business areas impacted by risks.

Impacted Area	Research Articles
Advanced Metering Infratructure/Smart Meters	[4,16,20,22,23,26,30,49,50,52,63,71,78,91,100,103,113,119,123,145,146]
Grid Distribution	[9,15,19,20,22,35,38,53,54,61,62], [81,88,121,127,132,138,147]
Microgrids	[1,3,12,18,24,36,43,58,67,79,80,86,98,102,106]
Grid Communications	[8,13,39,40,64,74,83,89,109,114,124,133,140]
Smart Grid Devices	[7,17,45,68,85,133,143–145]
SCADA Systems	[22,27,60,73,125,133,136,145]
Substations	[36,51,57,83,104,134,142]
Protection Relays	[5,11,37,112,121,130,132]
Electric Vehicle Charging Networks	[10,77,90,105,126,128]
Grid Transmission	[22,35,38,77,121]
Grid Power Supply	[2,48,129,134]
Smart Homes/Buildings/Cities	[63,76,135]
Voltage Control/Power Flow	[28,72,107]
Digital Substations	[93,137]
Automated Generation and Control	[35,75]
Grid Operation	[59,120]
Grid Security	[29,101]
Smart Demand Response	[21]
Virtual Power Plants	[69]
SDN-Enabled Smart Grids	[85]
Peer-to-Peer Electricity Markets	[111]
State Estimation	[97]
Power Industry	[32]

is used as a benchmark test case for power system analysis, has 33 buses, 38 branches and six generators.

Thirteen articles used simulation or emulation methods to verify the real consequences on smart grids. These include hybrid simulation-emulation, Mininet emulation and the use of simulation/emulation tools such as OMNeT++ [94], GridLAB-D [95] and Simulink [84]. Also covered in the research literature were scenario-based methods (three articles), probabilistic methods (two articles), Markov modeling methods (two articles) and electric vehicle charging network based methods (one article).

Table 3 lists the six types of methods used to verify real consequences on smart grids along with the specific articles in the survey pool that cover the methods.

The research reveals that, although several methods demonstrate the real consequences as a result of risk analysis, more research needs to be done to ascertain the degrees of the actual consequences. One reason is that most verification mechanisms focus on specific areas such as advanced metering infrastructures/smart meters, grid distribution and supervisory control and data acquisition (SCADA) systems or specific cyber attacks such as denial-of-service and false data injection.

However, no articles have as yet employed simulation to compare the results of the most serious consequences of cyber attacks on smart grid assets. Simulations may be based on the perceived consequences of cyber attacks on smart-grid-enabled business cases as described in [122]. By basing simulations on business

Table 3. Methods for verifying real consequences.

Real Consequence Verification Methods	Research Articles
IEEE Test Systems	[2,5,7,27,36,39,62,77,78,80,97,103,109,112,117,121,130,133,134,136,143,144]
Simulation/Emulation Methods	[16,37,38,53,54,67,101,107,114,125,127,132,137]
Scenario-Based Methods	[17,74,98]
Probabilistic Methods	[45,110]
Markov Modeling Methods	[79,146]
Electric Vehicle Charging	[105]
Network Based Method	[105]

cases with the highest consequence levels, there is a better chance of identifying, through accurate verification, the business cases that are most crucial to smart grid operation.

6 Discussion

The primary consequences of cyber attacks on smart grids include equipment damage, power outage, economic impact, consumer privacy impact and overall electricity delivery disruption. These consequences may be evaluated as follows:

- **Equipment Damage:** This is evaluated by assessing the vulnerabilities of equipment to cyber attacks and conducting simulations that demonstrate how the equipment would perform under attack conditions.
- **Power Outage:** This is evaluated by analyzing historical data on power outages, conducting simulations of different scenarios and assessing the impacts of outages on consumers.
- **Economic Impact:** This is evaluated by analyzing the costs of power outages, impacts on revenue and business continuity, and recovery costs.
- **Consumer Privacy Impact:** This is evaluated by analyzing the types of information at risk, potential consequences of breaches and identifying the best practices for protecting customer data.
- **Overall Electricity Delivery Disruption:** This is evaluated by assessing the general resilience and robustness of smart grids and identifying potential vulnerabilities and areas for improvement.

It is important to note that as smart grid complexity and interdependencies increase, the impacts should be evaluated using a holistic approach and advanced modeling and simulation tools. The research results indicate that 23 significant business areas are impacted by the cyber attack risks identified by the 120 of the 125 research articles in the survey pool (Table 2). Additionally, 67 of the

120 articles (53.6%) focus on the verification of the real consequences of cyber attacks. Table 3 lists the methods used to evaluate the real consequences of cyber attacks on smart grids. However, detailed analysis of the 67 articles revealed that, although the degrees of cyber attack consequences can be confirmed, their focus is limited to specific portions of smart grids and/or particular types of cyber attacks.

For example, Soykan and Bagriyanik [117] employed the Gridlab-D open-source power system simulation and analysis tool on the IEEE European Low Voltage Feeder test system. Their simulations demonstrated that cyber attack consequences include regional outages that could lead to large-scale blackouts due to cascading effects on the power system. However, they only studied a phishing attack launched via a text message to capture the credentials needed to access a demand response program. Manipulation of demand response program operations enabled the theft of sensitive information as well as electricity supply disruption.

Likewise, Teixeira et al. [121] evaluated the consequences of false data injection attacks on power transmission networks using an IEEE 14-bus benchmark test system, but they only considered attacks on sensor data.

Yan et al. [136] employed the IEEE 39-bus system model, but only to monitor voltage stability during and after cyber intrusions. AlMajal et al. [5] also used the IEEE 39-bus test system to evaluate the consequences of manipulating circuit breakers and the effects of integrating photovoltaic systems on smart grid stability under circuit breaker manipulation scenarios.

Akhtar et al. [2] analyzed the reliability of integrating solar and wind energy resources in a smart grid, but without considering cyber threats. Dogaru and Dumitrache [27] used the IEEE-9 bus benchmark system to simulate the effects of false data injection and message replay attacks on power grid operations. Alrowaili et al. [7] employed a 12-bus power system model using a PowerWorld simulator and launched cyber attacks on critical assets such as circuit breakers and evaluated their impacts on the physical system.

Lanzrath et al. [74] applied scenario-based methods using real devices. However, they only evaluated electromagnetic interference. Yayilgan et al. [137] employed a Mininet emulator with an IEC 61850 library to simulate cyber attacks on digital substations and demonstrate their impacts. This research needs to be moved to real devices via simulation and extended beyond digital substations to verify the real-world consequences of cyber attacks on smart grids.

7 Validity Evaluation

Construct validity [108] reflects the extent to which the contents of the articles analyzed in this research actually represent what was intended and what was assessed according to the research questions. The key point is how closely the consequence verification concepts are understood by the authors of the analyzed articles and the researchers involved in this survey study. Clear criteria and human evaluations were applied to analyze the strength of the consequence evaluation in each article in the survey pool. However, variances in the details of the analyzed research articles rendered the consequence evaluation strengths difficult to assess. Therefore, the interpretations may be characterized as posing threats to the construct validity. Similarly, the need for human interpretation poses a threat to the reliability of the research results.

8 Conclusions

This research has conducted a thorough analysis of the research literature on smart grid cyber risk assessment and consequence verification from 2009 to 2023. A systematic search of prominent research databases covering the technical areas of interest yielded 839 articles. Preliminary culling of articles followed by deep analyses of article content yielded a pool of 125 articles that focused on smart grid risk analysis. A total of 120 (96%) of the articles in the pool also studied the consequences of the assessed risks to some extent. However, the fairly large percentage (46.4%) of smart grid risk assessment articles that ignored real consequence verification demonstrates a key gap in the research literature.

Two key results of this research are the identification of 23 smart-grid-enabled business areas impacted by cyber risks and six methods for verifying the real consequences of cyber attacks on smart grids. Future work will apply real consequence verification techniques to rank the smart-grid-enabled business areas as well as individual business cases based on their potential disruptive impacts. Real consequence verification is important because it helps prioritize security investments for mitigating potential cyber attacks and their negative impacts.

References

1. Abercrombie, R., Ollis, T., Sheldon, F., Jillepalli, A.: Microgrid disaster resiliency analysis: reducing costs in continuity of operations planning. In: Proceedings of the Fifty-Second Hawaii International Conference on System Sciences, pp. 3532–3541 (2019)
2. Akhtar, I., Kirmani, S., Jameel, M.: Reliability assessment of power systems considering the impact of renewable energy sources integration into grid with advanced intelligent strategies. IEEE Access **9**, 32485–32497 (2021)
3. Akula, S., Salehfar, H.: Risk-based classical failure mode and effect analysis of microgrid cyber-physical energy systems. In: Proceedings of the North American Power Symposium (2021)
4. AlMajali, A., Rice, E., Viswanathan, A., Tan, K., Neuman, C.: A systems approach to analyzing cyber-physical threats in the smart grid. In: Proceedings of the IEEE International Conference on Smart Grid Communications, pp. 456–461 (2013)
5. AlMajali, A., Wadhawan, Y., Saadeh, M., Shalalfeh, L., Neuman, C.: Risk assessment of smart grids under cyber-physical attacks using Bayesian networks. Int. J. Electron. Secur. Digit. Forensics **12**(4), 357–385 (2020)

6. Aloul, F., Al-Ali, A., Al-Dalky, R., Al-Mardini, M., El-Hajj, W.: Smart grid security: threats, vulnerabilities and solutions. Int. J. Smart Grid Clean Energy **1**(1), 1–6 (2012)
7. Alrowaili, Y., Saxena, N., Burnap, P.: Determining asset criticality in cyber-physical smart grid. In: Proceedings of the Twenty-Sixth European Symposium on Research in Computer Security, pp. 770–776 (2021)
8. Atat, R., Ismail, M., Refaat, S., Serpedin, E., Overbye, T.: Cascading failure vulnerability analysis in interdependent power communications networks. IEEE Syst. J. **16**(3), 3500–3511 (2021)
9. Atmaja, T., Fitriana: Cyber security strategy for future distributed energy delivery systems. In: Proceedings of the International Conference on Electrical Engineering and Informatics (2011)
10. Ayoub, N., Gabbar, H.: Risk-based lifecycle assessment of hybrid transportation infrastructures as integrated with smart energy grids. In: Gabbar, H. (ed.) Smart Energy Grid Engineering, pp. 399–432. Academic Press, London (2017)
11. Aziz, Q., Sonde, G.: Protection and control analytics for a reliable grid. In: Proceedings of the Sixty-Ninth Annual Conference for Protective Relay Engineers, pp. 1–10 (2016)
12. Azizi, A., Peyghami, S., Wang, H., Blaabjerg, F.: Risk evaluation of hybrid microgrids considering DC-link voltage stability. In: Proceedings of the Thirteenth IEEE International Symposium on Power Electronics for Distributed Generation Systems (2022)
13. Baig, Z., Zeadally, S.: Cyber-security risk assessment framework for critical infrastructures. Intell. Autom. Soft Comput. **25**(1), 121–130 (2019)
14. Bodungen, C., Singer, B., Shbeeb, A., Wilhoit, K., Hilt, S.: Hacking Exposed - Industrial Control Systems: ICS and SCADA Security Secrets and Solutions. McGraw Hill, New York (2016)
15. Bracale, A., Caramia, P., Carpinelli, G., De Falco, P.: Probabilistic management of power delivery based on dynamic transformer rating. In: Proceedings of the International Conference on Probabilistic Methods Applied to Power Systems (2020)
16. Cameron, C., Patsios, C., Taylor, P., Pourmirza, Z.: Using self-organizing architectures to mitigate the impacts of denial-of-service attacks on voltage control schemes. IEEE Trans. Smart Grid **10**(3), 3010–3019 (2019)
17. Cardenas, D., Hahn, A.: IoT threats to the smart grid: a framework for analyzing emerging risks. In: Proceedings of the Northwest Cybersecurity Symposium (2019). Article no. 1
18. Chen, Q., Mili, L.: Assessing the impacts of microgrids on composite power system reliability. In: Proceedings of the IEEE Power and Energy Society General Meeting (2013)
19. Chu, T., Wang, T., Cao, C., Huang, W., Wang, Y.: Self-healing control method in abnormal state of distribution network. In: Proceedings of the Chinese Automation Congress, pp. 6438–6443 (2020)
20. Chumnuan, R., Rerkpreedapong, D.: A practicable framework for risk assessment of distribution transformers using PEA smart meter data. In: Proceedings of the Eighteenth International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 590–594 (2021)
21. Dahleh, M.: Plenary talk: resilience and risk in networked systems. In: Proceedings of the Twenty-Second Mediterranean Conference on Control and Automation, p. 605 (2014)

22. Datta Ray, P., Harnoor, R., Hentea, M.: Smart power grid security: a unified risk management approach. In: Proceedings of the Forty-Fourth Annual IEEE International Carnahan Conference on Security Technology, pp. 276–285 (2010)
23. De, S., Metayer, D.: Privacy harm analysis: a case study on smart grids. In: Proceedings of the IEEE Security and Privacy Workshops, pp. 58–65 (2016)
24. De Vanna, G., Longo, M., Foiadelli, F., Panteli, M., Galeela, M.: Reliability and resilience analysis and comparison of off-grid microgrids. In: Proceedings of the Fifty-Fifth International Universities Power Engineering Conference (2020)
25. Ding, J., Qammar, A., Zhang, Z., Karim, A., Ning, H.: Cyber threats to smart grids: review, taxonomy, potential solutions and future directions. Energies **15**(18), 6799 (2022)
26. Ding, Z., Xiang, Y., Wang, L.: Incorporating unidentifiable cyber attacks in power system reliability assessments. In: Proceedings of the IEEE Power and Energy Society General Meeting (2018)
27. Dogaru, D., Dumitrache, I.: Modeling the dynamic electrical system in the context of cyber attacks. UPB Sci. Bull. Ser. C **80**(2), 3–16 (2018)
28. Dondossola, G., Terruggia, R.: Security of communications in voltage control for grids connecting DER: impact analysis and anomalous behaviors. CIGRE Sci. Eng. J. **CSE–002**, 30–39 (2014)
29. Dong, Z.: Smart grid cyber security. In: Proceedings of the Thirteenth International Conference on Control Automation, Robotics and Vision (2014)
30. Duren, M., Aldridge, H., Abercrombie, R., Sheldon, F.: Designing and operating through compromise: architectural analysis of CKMS for the advanced metering infrastructure. In: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop (2013). Article no. 48
31. Electricity Information Sharing and Analysis Center (E-ISAC), TLP: White - Analysis of the Cyber Attack on the Ukrainian Power Grid, Defense Use Case, Washington, DC (2016)
32. Eroshenko, S., Bramm, A., Zinovieva, E., Vozisova, O.: Power industry risk assessment: current practices. In: Proceedings of the Ural-Siberian Smart Energy Conference, pp. 197–200 (2021)
33. European Parliament and the Council of the European Union, Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 Concerning Common Rules for the Internal Market in Electricity and Repealing Directive 2003/54/EC, Document 32009L0072, Brussels, Belgium (2009)
34. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid - the new and improved power grid: a survey. IEEE Commun. Surv. Tut. **14**(4), 944–980 (2012)
35. Fergus, D.: Advances in cyber risk mitigation for the power generation industry - an integrated approach. In: Proceedings of the Fifty-Third ISA POWID Symposium, pp. 33–49 (2010)
36. Fu, R., Huang, X., Sun, J., Zhou, Z., Chen, D., Wu, Y.: Stability analysis of the cyber-physical microgrid system under intermittent DoS attacks. Energies **10**(5), 680 (2017)
37. Fuchs, J., Jaeger, J.: Integrated analysis of network and protection system in future grids. In: Proceedings of the AFRICON Conference (2013)
38. Fuchs, J., Jaeger, J.: Smart grid study on protection security issues. In: Proceedings of the IEEE Power and Engineering Society Innovative Smart Grid Technologies Conference Europe (2013)
39. Fung, C., Roumani, M., Wong, K.: A proposed study on the economic impacts due to cyber attacks in the smart grid: a risk based assessment. In: Proceedings of the IEEE Power and Energy Society General Meeting (2013)

40. Gehrke, O., Heussen, K., Korman, M.: Integrated multi-domain risk assessment using automated hypothesis testing. In: Proceedings of the Second Workshop on Cyber-Physical Security and Resilience in Smart Grids, pp. 55–60 (2017)
41. Gharavi, H., Hu, B.: Smart Grid Communications, National Institite of Standards and Technology, Gaithersburg, Maryland, 8 July 2023. www.nist.gov/programs-projects/smart-grid-communications-0
42. Ghelani, D.: Cyber security in smart grids, threats and possible solutions. Preprint, Department of Computer Engineering, Gujarat Technological University, Modasa, India, 22 September 2022. www.authorea.com/users/506161/articles/587230-cyber-security-in-smart-grids-threats-and-possible-solutions
43. Ghose, T., Pandey, H., Gadham, K.: Risk assessment of microgrid aggregators considering demand response and uncertain renewable energy sources. J. Mod. Power Syst. Clean Energy **7**(6), 1619–1631 (2019)
44. Gou, B., Zheng, H., Wu, W., Yu, X.: The statistical law of power system blackouts. In: Proceedings of the Thirty-Eighth North American Power Symposium, pp. 495–501 (2006)
45. Goyal, A., Kim, Y., Lavin, M., Basu, C., Kumar, T.: Data-driven risk analysis for poles in the electric grid. In: Proceedings of the IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (2016)
46. Gunduz, M., Das, R.: Analysis of cyber attacks on smart grid applications. In: Proceedings of the International Conference on Artificial Intelligence and Data Processing (2018)
47. Gunduz, M., Das, R.: Cyber-security of the smart grid: threats and potential solutions. Comput. Netw. **169**, 107094 (2020)
48. Gunduz, H., Jayaweera, D.: Reliability assessment of a power system with cyber-physical interactive operation of photovoltaic systems. Int. J. Electr. Power Energy Syst. **101**, 371–384 (2018)
49. Habash, R., Groza, V., Krewski, D., Paoli, G.: A risk assessment framework for the smart grid. In: Proceedings of the IEEE Electrical Power and Energy Conference (2013)
50. Hahn, A.: Smart grid architecture risk optimization through vulnerability scoring. In: Proceedings of the IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply, pp. 36–41 (2010)
51. Han, H., Yan, X., Ma, C.: Security risk assessment of IEC 61850 based substation automation system. In: Proceedings of the Seventh International Conference on Electromechanical Control Technology and Transportation, SPIE Proceedings, vol. 12302, pp. 330–335 (2022)
52. Hansen, A., Staggs, J., Shenoi, S.: Security analysis of an advanced metering infrastructure. Int. J. Crit. Infrastruct. Prot. **18**, 3–19 (2017)
53. Hashemi-Dezaki, H., Agah, S., Askarian-Abyaneh, H., Haeri-Khiavi, H.: Sensitivity analysis of smart grid reliability due to indirect cyber-power interdependencies under various DG technologies, DG penetrations and operation times. Energy Conve. Manage. **108**, 377–391 (2016)
54. Hashemi-Dezaki, H., Askarian-Abyaneh, H., Haeri-Khiavi, H.: Impacts of direct cyber-power interdependencies on smart grid reliability under various penetration levels of microturbine/wind/solar distributed generation. IET Gener. Transm. Distrib. **10**(4), 928–937 (2016)
55. He, H., Yan, J.: Cyber-physical attacks and defenses in the smart grid: a survey. IET Cyber-Phys. Syst. Theor. Appl. **1**(1), 13–27 (2016)

56. Humayun, M., Niazi, M., Jhanjhi, N., Alshayeb, M., Mahmood, S.: Cyber security threats and vulnerabilities: a systematic mapping study. Arab. J. Sci. Eng. **45**, 3171–3189 (2020)
57. Iftimie, I., Huskaj, G.: Strengthening the cybersecurity of smart grids: the role of artificial intelligence in resiliency of substation intelligent electronic devices. In: Proceedings of the Nineteenth European Conference on Cyber Warfare and Security, pp. 143–150 (2020)
58. Jamieson, M., Hong, Q., Han, J., Paladhi, S., Booth, C.: Digital-twin-based real-time assessment of resilience in microgrids. In: Proceedings of the Eleventh International Conference on Renewable Power Generation - Meeting Net Zero Carbon, pp. 213–217 (2022)
59. Jelacic, B., Lendak, I., Stoja, S., Stanojevic, M., Rosic, D.: Security risk assessment based cloud migration methodology for smart grid OT Services. Acta Polytechnica Hungarica **17**(5), 113–134 (2020)
60. Jelacic, B., Rosic, D., Lendak, I., Stanojevic, M., Stoja, S.: STRIDE to a secure smart grid in a hybrid cloud. In: Proceedings of the International Workshops on Computer Security, ESORICS 2017, CyberICPS 2017 and SECPRE 2017, pp. 77–90 (2018)
61. Ji, X., Bai, D., Xu, J., Liu, S., Shan, S.: Research on risk indicator system of smart distribution grid based on LVQ neural network. In: Proceedings of the Chinese Automation Congress, pp. 3070–3075 (2019)
62. Jia, H., Qi, W., Liu, Z., Wang, B., Zeng, Y., Xu, T.: Hierarchical risk assessment of transmission system considering the influence of active distribution network. IEEE Trans. Power Syst. **30**(2), 1084–1093 (2015)
63. Karatzas, S., Chassiakos, A.: System-theoretic process analysis for hazard analysis in complex systems: the case of "demand-side management in a smart grid". Systems **8**(3), 33 (2020)
64. Kelli, V., Radoglou-Grammatikis, P., Lagkas, T., Markakis, E., Sarigiannidis, P.: Risk analysis of DNP3 attacks. In: Proceedings of the IEEE International Conference on Cyber Security and Resilience, pp. 351–356 (2022)
65. Khelifa, B., Abla, S.: Security concerns in smart grids: threats, vulnerabilities and countermeasures. In: Proceedings of the Third International Renewable and Sustainable Energy Conference (2015)
66. Knapp, E., Langill, J.: Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA and Other Industrial Control Systems, Syngress, Waltham, Massachusetts (2015)
67. Koraz, Y., Gabbar, A.: Risk analysis and self-healing approach for resilient interconnect micro energy grids. Sustain. Urban Areas **32**, 638–653 (2017)
68. Kumar, D., Pandey, D., Khan, A., Nayyar, H.: Cyber risk analysis of critical information infrastructure. In: Proceedings of the Sixth International Conference and Exhibition on Smart Grids and Smart Cities, pp. 1–9 (2022)
69. Kumar, V., Narasimhan, V.: Using deep learning for assessing cybersecurity economic risks in virtual power plants. In: Proceedings of the Seventh International Conference on Electrical Energy Systems, pp. 530–537 (2021)
70. Kundur, D., Feng, X., Liu, S., Zourntos, T., Butler-Purry, K.: Towards a framework for cyber attack impact analysis of the electric smart grid. In: Proceedings of the First IEEE International Conference on Smart Grid Communications, pp. 244–249 (2010)
71. Langer, L., Kammerstetter, M.: The evolution of the smart grid threat landscape and cross-domain risk assessment. In: Skopik, F., Smith, P. (eds.) Smart Grid

Security: Innovative Solutions for a Modernized Grid, Syngress, Waltham, Massachusetts, pp. 49–77 (2015)

72. Langer, L., Smith, P., Hutle, M.: Smart grid cybersecurity risk assessment. In: Proceedings of the International Symposium on Smart Electric Distribution Systems and Technologies, pp. 475–482 (2015)
73. Langer, L., Smith, P., Hutle, M., Schaeffer-Filho, A.: Analyzing cyber-physical attacks on a smart grid: a voltage control use case. In: Proceedings of the Power Systems Computation Conference (2016)
74. Lanzrath, M., Suhrke, M., Hirsch, H.: HPEM-based risk assessment of substations enabled for the smart grid. IEEE Trans. Electromagn. Compat. **62**(1), 173–185 (2019)
75. Law, Y., Alpcan, T., Palaniswami, M.: Security games for risk minimization in automatic generation control. IEEE Trans. Power Syst. **30**(1), 223–232 (2014)
76. Li, X., Li, H., Sun, B., Wang, F.: Assessing the information security risk for an evolving smart city based on fuzzy and grey FMEA. J. Intell. Fuzzy Syst. **34**(4), 2491–2501 (2018)
77. Li, Y., Xie, K., Wang, L., Xiang, Y.: The impact of PHEV charging and network topology optimization on bulk power system reliability. Electr. Power Syst. Res. **163**(A), 85–97 (2018)
78. Liu, X., Che, L., Gao, K., Li, Z.: Power system intra-interval operational security under false data injection attacks. IEEE Trans. Industr. Inf. **16**(8), 4997–5008 (2020)
79. Liu, X., Shahidehpour, M., Cao, Y., Wu, L., Wei, W., Liu, X.: Microgrid risk analysis considering the impact of cyber attacks on solar PV and ESS control systems. IEEE Trans. Smart Grid **8**(3), 1330–1339 (2017)
80. Liu, Y., Gu, H.: Research on a risk control system in a regional power grid. In: Proceedings of the China International Conference on Electricity Distribution (2012)
81. Liu, Z., et al.: Hierarchical risk assessment of a transmission network considering the influence of micro-grid. In: Proceedings of the IEEE Power and Energy Society General Meeting (2013)
82. Maas, G., Bial, M., Fijalkowski, J.: Final Report - System Disturbance on 4 November 2006, Union for the Coordination of Transmission of Electricity in Europe, Brussels, Belgium (2007). www.eepublicdownloads.entsoe.eu/clean-documents/pre2015/publications/ce/otherreports/Final-Report-20070130.pdf
83. Martinez, J., Fernandez, M.: Cyber security intrusion detection for electrical stations in power networks. In: Proceedings of the CIRED Porto Workshop: E-Mobility and Power Distribution Systems, pp. 11–14 (2022)
84. Mathworks, Simulink, Natick, Massachusetts (2023). www.mathworks.com/products/simulink.html
85. Maziku, H., Shetty, S., Nicol, D.: Security risk assessment for SDN-enabled smart grids. Comput. Commun. **133**, 1–11 (2019)
86. Mishra, S., Anderson, K., Miller, B., Boyer, K., Warren, A.: Microgrid resilience: a holistic approach for assessing threats, identifying vulnerabilities and designing corresponding mitigation strategies. Appl. Energy **264**, 114726 (2020)
87. Mrabet, Z., Kaabouch, N., El Ghazi, H., El Ghazi, H.: Cyber-security in smart grid: survey and challenges. Comput. Electr. Eng. **67**, 469–482 (2018)
88. Mukherjee, S.: Applying the distribution system in grid restoration/NERC CIP-014 risk assessment. In: Proceedings of the IEEE Rural Electric Power Conference, pp. 103–105 (2015)

89. Muller, S., Harpes, C., Le Traon, Y., Gombault, S., Bonnin, J.-M., Hoffmann, P.: Dynamic risk analyses and dependency-aware root cause model for critical infrastructures. In: Havarneanu, G., Setola, R., Nassopoulos, H., Wolthusen, S. (eds.) CRITIS 2016. LNCS, vol. 10242, pp. 163–175. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71368-7_14
90. Mustafa, M., Zhang, N., Kalogridis, G., Fan, Z.: Smart electric vehicle charging: security analysis. In: Proceedings of the IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (2013)
91. Nateghi, A., Schaarschmidt, M., Fisahn, S., Garbe, H.: Vulnerability of wireless smart meters to electromagnetic interference sweep frequency jamming signals. In: Proceedings of the Joint IEEE International EMC/SI/PI and EMC Europe Symposium, pp. 755–759 (2021)
92. National Initiative for Cybersecurity Careers and Studies, Vocabulary, Cybersecurity and Infrastructure Security Agency, Washington, DC (2023). www.niccs.cisa.gov/cybersecurity-career-resources/glossary#C
93. Omerovic, A., Vefsnmo, H., Gjerde, O., Ravndal, S.T., Kvinnesland, A.: An industrial trial of an approach to identification and modelling of cybersecurity risks in the context of digital secondary substations. In: Kallel, S., Cuppens, F., Cuppens-Boulahia, N., Hadj Kacem, A. (eds.) CRiSIS 2019. LNCS, vol. 12026, pp. 17–33. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41568-6_2
94. OpenSim, OMNeT++ Discrete Event Simulator, Budapest, Hungary (2023). www.omnetpp.org
95. Pacific Northwest National Laboratory: GridLAB-D - A unique tool to design the smart grid, Richland, Washington (2023). www.gridlabd.org
96. Pagani, G., Aiello, M.: The power grid as a complex network: a survey. Phys. A **392**(11), 2688–2700 (2013)
97. Pan, K., Teixeira, A., Cvetkovic, M., Palensky, P.: Cyber risk analysis of combined data attacks against power system state estimation. IEEE Trans. Smart Grid **10**(3), 3044–3056 (2019)
98. Pedroza, G., Le Gall, P., Gaston, C., Bersey, F.: Timed-model-based method for security analysis and testing of smart grid systems. In: Proceedings of the Nineteenth IEEE International Symposium on Real-Time Distributed Computing, pp. 35–42 (2016)
99. Peyghami, S., Davari, P., Fotuhi-Firuzabad, M., Blaabjerg, F.: Standard test systems for modern power system analysis: an overview. IEEE Industr. Electron. **13**(4), 86–105 (2019)
100. Piatkowska, E., Bajraktari, A., Chhajed, D., Smith, P.: Tool support for data protection impact assessment in the smart grid. e & i Elektrotechnik und Informationstechnik **134**(1), 26–29 (2017). https://doi.org/10.1007/s00502-017-0484-4
101. Poudel, S., Ni, Z., Malla, N.: Real-time cyber physical system testbed for power system security and control. Int. J. Electr. Power Energy Syst. **90**, 124–133 (2017)
102. Qin, H., Pan, Z., Huang, L., Yu, M., Lin, X., Tao, Y.: Comprehensive evaluation of microgrid integration based on combination weighting. In: Proceedings of the Seventh IEEE International Conference on Power and Renewable Energy, pp. 331–335 (2022)
103. Rahman, M., Li, Y., Yan, J.: Multi-objective evolutionary optimization for worst-case analysis of false data injection attacks in the smart grid. In: Proceedings of the IEEE Congress on Evolutionary Computation (2020)

104. Rao Alla, R., Pahuja, G., Lather, J.: Importance-measures-based risk and reliability analysis of substation automation systems. In: Proceedings of the Annual IEEE India Conference (2014)
105. Reeh, D., Tapia, F., Chung, Y., Khaki, B., Chu, C., Gadh, R.: Vulnerability analysis and risk assessment of an EV charging system under cyber-physical threats. In: Proceedings of the IEEE Transportation Electrification Conference and Expo (2019)
106. Rekik, M., Chtourou, Z., Gransart, C., Atieh, A.: A cyber-physical threat analysis for microgrids. In: Proceedings of the Fifteenth International Multi-conference on Systems, Signals and Devices, pp. 731–737 (2018)
107. Roy, V., et al.: Design, development and experimental setup of a PMU network for monitoring and anomaly detection. In: Proceedings of the IEEE SoutheastCon (2019)
108. Runeson, P., Host, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131–164 (2009)
109. Sarkar, S., Teo, Y., Chang, E.: A cybersecurity assessment framework for virtual operational technology in power system automation. Simul. Model. Pract. Theor. **117**, 102453 (2022)
110. Sarmiento, H., Pampin, G., Castellanos, R., Ramírez, M., Villa, G., Mirabal, M.: Risk analysis: towards a smarter grid operation. In: Proceedings of the CIGRE 2011 Bologna Symposium - The Electric Power System of the Future: Integrating Supergrids and Microgrids, paper no. 341 (2011)
111. Seyedhossein, S., Moeini-Aghtaie, M.: Risk management framework for peer-to-peer electricity markets. Energy **261**(B), 125264 (2022)
112. Sheela, A., Revathi, S., Iqbal, A.: Cyber risk assessment for intelligent and non-intelligent attacks on power systems. In: Proceedings of the Second International Conference on Power and Embedded Drive Control, pp. 40–45 (2019)
113. Shrestha, M., Johansen, C., Noll, J., Roverso, D.: A methodology for security classification applied to smart grid infrastructures. Int. J. Crit. Infrastruct. Prot. **28**, 100342 (2020)
114. Sierla, S., Hurkala, M., Charitoudi, K., Yang, C., Vyatkin, V.: Security risk analysis for smart grid automation. In: Proceedings of the IEEE International Symposium on Industrial Electronics, pp. 1737–1744 (2014)
115. Smadi, A., Ajao, B., Johnson, B., Lei, H., Chakhchoukh, Y., Al-Haija, Q.: A comprehensive survey on cyber-physical smart grid testbed architectures: requirements and challenges. Electronics **10**(9), 1043 (2021)
116. Sobeslav, V., Horalek, J., Svoboda, T., Svecova, H.: Security consideration of BIA utilization in smart electricity metering systems. In: Proceedings of the Fourteenth International Conference on Computational Collective Intelligence, pp. 585–597 (2022)
117. Soykan, E., Bagriyanik, M.: The effect of SMiShing attack on the security of demand response programs. Energies **13**(17), 4542 (2020)
118. Su, S., Wang, Y., Long, Y., Li, Y., Jiang, Y.: Cyber attack impact on power system blackout. In: Proceedings of the IET Conference on Reliability of Transmission and Distribution Networks (2011)
119. Suleiman, H., Svetinovic, D.: Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure. Requirements Eng. **18**(3), 251–279 (2013)
120. Sun, Y., Ma, T., Huang, B., Xu, W., Yu, B., Zhu, Y.: Risk assessment of power system secondary devices for power grid operation. In: Proceedings of the China International Conference on Electricity Distribution (2012)

121. Teixeira, A., Kupzog, F., Sandberg, H., Johansson, K.: Cyber-secure and resilient architectures for industrial control systems. In: Skopik, F., and P. Smith (eds.) Smart Grid Security: Innovative Solutions for a Modernized Grid, Syngress, Waltham, Massachusetts, pp. 149–183 (2015)
122. Toftegaard, Ø., Abraham, D., Shenoi, S., Hämmerli, B.: Smart-grid-enabled business cases and the consequences of cyber attacks. In: Staggs, J., Shenoi, S. (eds.) Critical Infrastructure Protection XVII, vol. 686, pp. 17–39. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-49585-4_3
123. Tondel, I., Line, M., Johansen, G.: Assessing information security risks of AMI: what makes it so difficult? In: Proceedings of the International Conference on Information Systems Security and Privacy, pp. 56–63 (2015)
124. Tranchita, C., Hadjsaid, N., Viziteu, M., Rozel, B., Caire, R.: ICT and powers systems: an integrated approach. In: Lukszo, Z., Deconinck, G., Weijnen, M. (eds.) Securing Electricity Supply in the Cyber Age. Topics in Safety, Risk, Reliability and Quality, vol. 15, pp. 71–109. Springer, Dordrecht (2010). https://doi.org/10.1007/978-90-481-3594-3_5
125. Urias, V., Van Leeuwen, B., Richardson, B.: Supervisory command and data acquisition system cyber security analysis using a live, virtual and constructive testbed. In: Proceedings of the IEEE Military Communications Conference (2012)
126. VanYe, C., et al.: Trust and security of embedded smart devices in advanced logistics systems. In: Proceedings of the Systems and Information Engineering Design Symposium (2021)
127. Venemans, P., Schreuder, M.: A method for the quantitative assesment of reliability of smart grids. In: Proceedings of the CIRED Workshop: Integration of Renewables in the Distribution Grid (2012)
128. Venkatachalapathy, S.: Common vulnerability considerations as an integral part of the automotive cybersecurity engineering process. In: Proceedings of the Tenth SAE India International Mobility Conference (2022)
129. Wallnerstrom, C., Tjernberg, L.B., Hilber, P., Jurgensen, J.: Framework for system analyses of smart grid solutions with examples from the Gotland case. In: Proceedings of the International Conference on Probabilistic Methods Applied to Power Systems (2016)
130. Wang, P., Ashok, A., Govindarasu, M.: Cyber-physical risk assessment of a smart grid system protection scheme. In: Proceedings of the IEEE Power and Energy Society General Meeting (2015)
131. Wang, Q., Palacios-Trujillo, R., Granelli, F.: A novel architecture for the distribution section of a smart grid with renewable sources and power storage. In: Proceedings of the Twenty-Third International Conference on Computer Communications and Networks (2013)
132. Wang, Y., Qin, L., Guo, Q., Jin, H.: The study of CIM based relay protection model considering distributed generation. In: Proceedings of the International Conference on Advanced Power System Automation and Protection, vol. 3, pp. 1875–1879 (2011)
133. Woo, P., Kim, B.: Methodology of cyber security assessment in the smart grid. J. Electr. Eng. Technol. **12**(2), 495–501 (2017)
134. Xiang, Y., Wang, L., Zhang, Y.: Adequacy evaluation of electric power grids considering substation cyber vulnerabilities. Int. J. Electr. Power Energy Syst. **96**, 368–379 (2018)
135. Xu, P., Gao, X., Agee, P.: Management solutions for cyber-physical security in smart built environments. In: Proceedings of the Construction Research Congress, pp. 1024–1032 (2022)

136. Yan, J., Govindarasu, M., Liu, C., Vaidya, U.: A PMU-based risk assessment framework for power control systems. In: Proceedings of the IEEE Power and Energy Society General Meeting (2013)
137. Yayilgan, S., Holik, F., Abomhara, M., Abraham, D., Gebremedhin, A.: An approach for analyzing cyber security threats and attacks: a case study of digital substations in Norway. Electronics **11**(23), 4006 (2022)
138. Yin, H., Liu, D., Weng, J.: Risk analysis of a cyber-physical distribution system considering cyber attacks on a V2G system. In: Proceedings of the Tenth Renewable Power Generation Conference, pp. 841–846 (2021)
139. Yu, C., Zhang, H., Zhao, L.: Architecture design for smart grid. Energy Procedia **17**, 1524–1528 (2012)
140. Yu, W., Xue, Y., Luo, J., Ni, M., Tong, H., Huang, T.: A UHV grid security and stability defense system: considering the risk of power system communications. IEEE Trans. Smart Grid **7**(1), 491–500 (2016)
141. Yu, X., Cecati, C., Dillon, T., Simoes, M.: The new frontier of smart grids. IEEE Industr. Electron. **5**(3), 49–63 (2011)
142. Zhang, H., Zhang, J., Liu, N., Wu, X.: Function-oriented information asset identification in substation automation systems. In: Proceedings of the Asia-Pacific Power and Energy Engineering Conference (2009)
143. Zhao, T., Wang, D., Lu, D., Zeng, Y., Liu, Y.: A risk assessment method for cascading failures caused by electric cyber-physical systems. In: Proceedings of the Fifth IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, pp. 787–791 (2016)
144. Zhu, B., Deng, S., Xu, Y., Yuan, X., Zhang, Z.: Information security risk propagation model based on the SEIR infectious disease model for smart grid. Information **10**(10), 323 (2019)
145. Zografopoulos, I., Ospina, J., Liu, X., Konstantinou, C.: Cyber-physical energy systems security: threat modeling, risk assessment, resources, metrics and case studies. IEEE Access **9**, 29775–29818 (2021)
146. Zonouz, S., Berthier, R., Haghani, P.: A fuzzy Markov model for scalable reliability analysis of advanced metering infrastructure. In: Proceedings of the IEEE Power and Engineering Society Conference on Innovative Smart Grid Technologies (2012)
147. Zuniga, A., Baleia, A., Fernandes, J., Da Costa Branco, P: Classical failure modes and effects analysis in the context of smart grid cyber-physical systems. Energies **13**(5), 1215 (2020)

Measuring the Impacts of Power Outages on Internet Hosts in the United States

Scott Anderson(✉), Tucker Bell, Patrick Egan, Nathan Weinshenker, and Paul Barford

University of Wisconsin, Madison, Madison, WI, USA
standerson4@wisc.edu

Abstract. Power outages are a well-known threat to Internet communications systems. While Internet service providers address this threat via backup power systems in datacenters and points-of-presence, office buildings and private homes may not have similar capabilities.

This chapter describes an empirical study that assesses how power outages in the United States impact end-host access to the Internet. To conduct this study, the PowerPing system was created to monitor a power outage reporting website and measure end-host responsiveness in the impacted areas. PowerPing collected power outage and end-host responsiveness data over 14 months from June 2020 through July 2021.

The results reveal that power outages affecting 10% or more customers in U.S. counties occur at a rate of about 50 events/day. The outages typically impact about 3,000 customers and services are restored in just under two hours. The end-host responsiveness characteristics for typical power outage events are also reported. Surprisingly, only a weak correlation exists between power outage impacts and service restoration periods versus end-host responsiveness. This suggests that improving backup power for network devices in office buildings and private homes may enable end-hosts to maintain access to Internet service during typical power outages.

Keywords: Power Outages · Internet End-Hosts · Impacts

1 Introduction

The robust availability of Internet service to end-hosts in office buildings and private homes is essential to day-to-day activities. This was highlighted when people moved from offices and classrooms to their homes during the COVID-19 pandemic. Disruptions to service are not merely irksome or inconvenient; they can have real consequences in terms of lost work time and missed opportunities. The importance of connectivity is directly reflected in service level agreements (SLAs) between Internet service providers (ISPs) and their customers, which typically include specific guarantees on service availability [37]. However, several factors determine the realized availability of service to end-hosts.

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 62–90, 2024.
https://doi.org/10.1007/978-3-031-49585-4_4

Access to the Internet can be impaired by endogenous and exogenous events that affect single users and groups of users in geographic areas. Endogenous events include misconfigurations and equipment failures. Well-understood best practices exist for minimizing the durations and impacts of such events. Exogenous events include natural disasters, infrastructure failures, accidents and attacks. By definition, these events are outside the direct control of Internet service providers and often require other entities to make repairs before service can be restored. Understanding the causes and effects of exogenous events is essential to improving end-to-end network reliability.

Previous studies on the availability of communications systems in the face of exogenous events have focused on retrospective studies of natural disasters such as hurricanes [11], earthquakes [6] and severe weather events [28,34]. The studies present detailed data about the numbers of end-hosts that lost service and the time required to restore service, and also provide road maps for understanding other types of outage events.

This chapter considers the problem of how power outages impact the availability of Internet service to end-hosts. The focus is on wireline Internet services that are typically delivered to end-hosts via cable, digital subscriber line or fiber, and excludes cellular service. Three principal questions are considered: How do power outages impact Internet service to end-hosts on a day-to-day basis? What are the scopes and durations of typical power outages versus service availability? How can understanding typical power outage events inform new techniques and practices to improve network reliability?

This research differs from previous studies on exogenous events that impact communications systems because power outages are common events that occur daily in the United States [7]. There is also a simple solution to power outages – backup power supplies – assuming that the outage durations are relatively modest.

To conduct the study, a measurement system called PowerPing was developed to monitor the `PowerOutage.us` website that publishes current power outage reports by county in the United States [29]. The data was employed to identify U.S. counties to target with active probe-based measurements of end-host responsiveness. This was accomplished using a ZMap-based probing system that operated in two modes. The first mode conducted background probing of IP addresses geolocated to counties across the United States to establish baselines for responsiveness (i.e., end-hosts that respond to probes). The second mode, conducted when outages were identified, sent probe packets to the IP addresses in the targeted areas until power was restored.

Certain challenges were encountered during the development, configuration and deployment of PowerPing. First, timely data about power outages was required to enable active probing of the affected areas to begin as soon as possible after the outages. The `PowerOutage.us` website was leveraged to obtain outage data at 12-min collection intervals. Second, a database containing IP addresses mapped to U.S. counties as probe targets was required. The database was constructed using Esri's ArcGIS [15] to assign IP addresses to counties based on latitude-longitude coordinates provided by MaxMind [24]. The term "end-hosts" used in this work refers to the IP addresses geolocated by MaxMind that, in many cases, may be home routers instead of computers. Third, PowerPing had to be configured to ensure that ZMap would effectively send and receive probes without biasing results. This was accomplished by deploying PowerPing at three CloudLab sites [8] to evaluate vantage point location bias, probe scaling and consistency. Upon conducting a series of tests, it was discovered that deploying PowerPing in a single location with a maximum probe limit of 60,000 packets/s was adequate to obtain consistent results. Finally, a baseline for responsive IP addresses in each U.S. county was established while minimizing the overall probe load on the network.

PowerPing was deployed to collect data over a 14-month period from June 2020 through July 2021. During this period, there were about 330,000 reported power outages with more than 14,000 outages affecting 10% or more customers in the counties. The power outages varied from impacting fewer than 100 customers to impacting 3.7 million customers in Harris County, Texas on February 16, 2021. The power outage durations varied from less than 24 min to an outage in Linn, Iowa that lasted for ten days starting on August 10, 2020. It was discovered that power outages across the United States follow strong diurnal cycles with the largest numbers of events taking place around midday. This can be explained, in part, by power company reports that outages are typically caused by humans through scheduled maintenance, vehicle accidents and high demand [3,10]. Also, power outages that are relatively significant in their impacts are not uncommon. Outages impacting 10% or more customers in a county occur at a rate of about 50 events per day with service restoration typically completed in under two hours.

Active probing of IP addresses conducted after outage reports reveals a wide range of impacts on service availability. The vast majority of Internet service outages impacted fewer than 1,000 end-hosts in the target areas and the service restoration periods were similar to the power restoration periods of about two hours. In aggregate, across all the power outages at a given time, a strong correlation ($R^2 = 0.99$) exists between the numbers of customers impacted by power outages and the numbers of unresponsive end-hosts. However, at the county level, the correlation is not as strong ($R^2 = 0.66$). Possible explanations for these and other results are discussed later in this chapter.

Ethical considerations related to web scraping and active measurements conducted in this research deserve mention. Low-rate scraping of publicly-available data was conducted with the goal of contributing to the public good; no financial benefits were sought or received. No laws were broken to obtain data [41,44] and ethical principles promulgated by major computing organizations such as the Electronic Frontier Foundation (EFF) [23] and Association for Computing Machinery (ACM) [36] were followed. Active measurements followed established methodologies [14,20,34] and the probing methodology limited the impacts to end-hosts and Internet service providers.

2 Related Work

Several techniques have been developed to measure Internet events and outages. These include active probe-based methods [28,31,34], measuring Border Gateway Protocol (BGP) advertisements [12], measuring changes in Network Time Protocol (NTP) traffic [39], passive techniques such as Chocolatine that leverage Internet background radiation [19], combinations of passive and active measurements such as Disco [35] and analyses of Internet service provider logs [32]. These techniques differ from the work described in this chapter because they mainly focus on network outages without considering their causes.

Of particular relevance to this work are two studies on the impacts of weather events on residential Internet service [28,34]. These studies developed and employed ThunderPing to measure end-host responsiveness in areas affected by severe weather events. The methodology described in this chapter was inspired by ThunderPing, but the objective of understanding end-host responsiveness in areas affected by power outages is different. The distinction is significant because severe weather is just one of several causes of power outages, which also include routine maintenance, human operator error, accidents and overload. Unlike the rare weather events studied using ThunderPing, the following sections demonstrate that power outages are common events, with hundreds of outages occurring every day. Additionally, forecasting is an established science for predicting weather whereas power outages are announced publicly only after they occur. Due to these differences, a completely new code base was developed to study how power outages impact end-host Internet service, helping enhance the understanding of the relationships between the two critical infrastructure sectors.

Tools and techniques for conducting active measurements of Internet hosts have evolved significantly over the years and this research was enabled by the advances. Due to hardware and network limitations in sending and processing active network probes, many early active probing studies focused on small sets of representative IP addresses in their regions of interest [16,20]. In contrast, this research actively probes as many IP addresses as possible in select geographic areas during specific events.

Several tools are available for conducting active Internet surveys, including Nmap and Scamper [22]. However, after evaluating the tools, ZMap was selected for its ability to rapidly and accurately scan large numbers of IP addresses in targeted IP subnets [14]. This study has benefitted from the open release of tools to the research community.

3 Datasets

This section describes the three datasets used in the research that cover current power outages, geographic information on U.S. counties and geographic distributions of IP subnets.

3.1 Power Outages

The two primary sources of data on U.S. power outages are the U.S. government and private power generation and distribution utilities. At the federal government level, the U.S. Energy Information Administration (EIA) publishes data about U.S. energy grid operations, including electricity supply, demand, generation and major disturbances and unusual occurrences [42,43]. However, the EIA data suffers from two major drawbacks that made it inappropriate for this research. First, the data is restricted to very large and/or very long duration outages. Second, there are delays of hours to days before data is published.

Private power utility companies are the primary source of U.S. outage data. The United States has more than 1,000 power utility companies that collectively serve more than 140 million customers (households). Many of the power utilities maintain online systems that track the occurrences and current status of power outages for their customers [17,26]. The online systems typically present maps of service areas along with pins showing the geographic locations, numbers of customers without power, reasons for the outages and expected resolution times. However, the maps only display current outage data, not historical outage data. Constantly collecting, parsing and storing current data from numerous power utilities to create a dataset of historical data are most challenging.

The `PowerOutage.us` website aggregates data from major U.S. utilities and presents a consolidated national view [30]. More than 680 power utility companies that serve more than 135 million customers across the United States are monitored to provide data about the numbers and percentages of customers without power in most U.S. counties. The data is updated every ten minutes to accommodate updates posted on utility websites. `PowerOutage.us` lists more than 20 companies and government organizations that use its outage data. The website is frequently quoted in news media reports on major power outages [5,25,29,40,45].

This study has leveraged consolidated data from `PowerOutage.us`. However, certain limitations exist compared with the data provided directly by utilities. Power utilities provide accurate and timely information to support their customers whereas `PowerOutage.us` outage data is likely delayed and can be incomplete. Additionally, `PowerOutage.us` does not track about 500 smaller power utilities with a total of 5.5 million customers, so the results of this study would not reflect all outages in the United States. Nevertheless, it is posited that the large data sample is representative of the power outage conditions experienced by most of the U.S. population.

3.2 U.S County Data

This study has sought to measure the impacts of power outages on end-host responsiveness in U.S. counties in the 48 conterminous states. The U.S. Census Bureau identifies the geographic boundaries of 3,108 counties in the conterminous states [13] and Esri ArcGIS [15] was employed to process this data. The Census Bureau also provides county area, population and population density

Table 1. Top ISPs by subnet count in MaxMind data for U.S. counties.

ISP	ASN	Subnets	Network Type
CHARTER	20115	277,471	Cable/Fiber
TWC-MIDWEST	10796	142,494	Cable
TWC-TEXAS	11427	118,322	Cable
BHN	33363	115,763	cable
TWC-PACWEST	20001	92,052	Cable
COMCAST	7922	81,287	Cable
TWC-CAROLINAS	11426	79,053	Cable
TWC-NORTHEAST	11351	67,616	Cable
TWC-NYC	12271	53,692	Cable
ATT-INTERNET4	7018	45,640	Cable/Fiber
UUNET	701	27,327	DSL/Fiber
CENTURYLINK-US-LEGACY-QWEST	209	23,289	DSL/Fiber
ASN-CXA-ALL-CCI-RDC	22773	16,247	Cable
WINDSTREAM	7029	9,600	DSL/Cable/Fiber
FRONTIER-FRTR	5650	8,615	DSL/Fiber

data that was used in the study. The PowerPing tool developed in this study was designed to employ counties as geographical units because they correspond to the smallest geographic resolution considered by `PowerOutage.us`.

3.3 End-Host IP Subnets

An objective of this study was to probe as many IPv4 addresses as possible in target counties during power outages to measure their impacts and durations. The MaxMind database [24] that provides (approximate) geographic locations (latitudes/longitudes) of variable-sized IP subnets worldwide was leveraged for this purpose. ArcGIS was employed to spatially connect the location data of each IPv4 subnet in the MaxMind database with the U.S. Census Bureau county shapefiles to identify subnets in the counties.

The study considered 1,377,238 variable-sized subnets from MaxMind in U.S. counties that were located in power utility service areas tracked by `PowerOutage.us`. The subnets are owned by 9,441 Internet service providers identified by their autonomous system numbers (ASNs); 44 service providers operated more than 1,000 subnets each.

Table 1 shows the top Internet service providers, dominated by large fixed service residential service providers. This study frequently refers to the responsiveness of "Internet hosts" or "end-hosts." Given the representation of Internet service providers listed in the table, the IP addresses used as probe targets in the study would most likely be home routers. Therefore, if they were responsive during power outages, it was assumed that service was available at the corresponding locations.

The MaxMind dataset limitations include inaccuracies in geolocation information, the incompleteness of the identified subnets, the use of subnet address space by Internet service providers in multiple geographic locations and the understanding of baseline end-host responsiveness in subnets. Additionally, Dynamic Host Configuration Protocol (DHCP) churn, i.e., the rate at which hosts change IP addresses, must be considered. North American Internet service providers do not change IP addresses assigned to end-hosts as frequently as providers elsewhere in the world; most U.S. IP addresses are consistently assigned to the same end-hosts for at least several weeks [27]. To account for IP subnet geographic relocation, the IP subnets from MaxMind were updated three times during the course of the study.

4 PowerPing

The PowerPing system developed for the study has two major functions – identifying the numbers of customers without power in 2,987 U.S. counties and conducting active measurements of end-hosts in counties experiencing outages and those not experiencing outages. PowerPing was written in Python 3.6 and is packaged in a GitHub repository for deployment on an Ubuntu 18.04 server in a cloud-based infrastructure.

During the research, PowerPing was deployed on CloudLab nodes [8]. CloudLab is a distributed computing infrastructure deployed from data centers in Utah, Wisconsin and South Carolina that supports experimental research.

4.1 Power Outage Identification

After a power outage occurs, several steps are taken by a power utility and by `PowerOutage.us` to post information online about the outage event. The power utility identifies the occurrence of the outage and posts the location and number of customers affected on its website. `PowerOutage.us` scrapes the power utility website, identifies the new outage and updates its website. The duration between the occurrence of an outage and its posting on `PowerOutage.us` is uncertain. However, the utility and `PowerOutage.us` have incentives to post outage information as soon as possible.

PowerPing scraped the `PowerOutage.us` website to harvest the total number of customers tracked and the number of customers without power in each of the 2,987 U.S. counties. Since power outages are unpredictable, other than scheduled maintenance, data on all counties was collected in 12-min intervals (epochs) to identify changes. The percentages of customers without power were computed during each epoch for three categories of counties – those experiencing outages impacting 10% or more customers, those in which outages were resolved within four hours, and those experiencing outages impacting less than 10% of the customers.

The start of an outage was set to the first epoch when 10% or more customers in a county experienced an outage. An outage was considered to be resolved when less than 2% of the customers in a county were without power. A county with a resolved outage was maintained as a "county of interest" for four hours after resolution, after which the county was removed from the list of counties of interest. The counties of interest list was maintained to accommodate situations where Internet service was unavailable even after power was restored. The county power outage status during each epoch was passed to the active measurement component of PowerPing.

Three issues must be noted with regard to the outage identification component of PowerPing. First, there were inherent delays between the start of a power outage in a county and PowerPing's identification of a power outage in the county. The delays were mostly external to PowerPing – delays in utilities identifying outages and delays in posting outage information on their public-facing websites. However, there also were delays in `PowerOutage.us` posting outage information on its website. Overall, the delays were due to automated processes, except for situations where customers manually informed utilities of outages. These delays are acknowledged, but it was not possible to reduce them any further. In any case, it is posited that the impact is a modest reduction in outage duration measurements. PowerPing was configured to employ a 12-min interval between harvesting outage information. This interval was identified during initial experimentation because it provided a good balance between the load on `PowerOutage.us`, timeliness of outage update reporting and end-host responsiveness probing (described in Sect. 4.2).

The second issue was that `PowerOutage.us` changed its format during the research, which prevented the harvesting of outage information until the code was adapted to process the reported outages. Future changes to `PowerOutage.us` will require additional PowerPing code updates.

The third issue is that only counties with 10% or more customers without power were considered. This convention was adopted for three reasons – it improved system efficiency by limiting the number of active probes sent during an epoch, it reduced the impact of probe traffic on the network and it helped differentiate the impact of an outage on responsiveness versus IP response churn for outages that affected small numbers of customers. However, there is the risk that outages in some of the largest U.S. counties could have been excluded. Nevertheless, the study identified power outages affecting 10% or more customers in five of the ten largest counties as well as in 13 of the 20 largest counties. End-host responsiveness measurements were performed successfully during the power outages in all 13 counties.

4.2 Active Measurement

The active measurement components of PowerPing implement Pre-Processing and IP address probing to assess end-host responsiveness.

Pre-processing. Efficiency was a key PowerPing design requirement due to the frequency of probing and the large numbers of IP addresses in target areas. Certain pre-processing tasks were implemented to address these issues. The tasks included classifying each IP subnet by county, identifying counties with IP subnets tracked by `PowerOutage.us` and specifying optimal system parameters for data collection, storage and processing.

The MaxMind dataset provides the latitudes and longitudes of IP subnets. The ArcGIS system was leveraged to associate each IP subnet with a state and county from the U.S. Census Bureau shapefiles covering all U.S. counties. Of the 3,108 counties in the conterminous United States, 3,093 counties were identified with subnets from MaxMind within their geographic perimeters.

During each active probe period, up to tens of megabytes of compressed and archived data on ongoing outages and ICMP responses were collected. A standard directory structure, file naming convention and file organization were created for storing and processing the results of each probe period.

End-Host Responsiveness Probing. The IP probing component of PowerPing was informed by prior studies that measured end-host responsiveness [14,20,34]. During each epoch, after U.S. counties were classified according to their power outage status (experiencing outages, recently resolved outages or not experiencing outages), PowerPing identified all the IP subnets in counties with outages, all the IP subnets in counties with outages that were resolved within four hours and all the IP subnets in a select set of counties without outages. Following this, PowerPing sent probes to all the IP addresses in the selected subnets and processed the responses. Finally, it stored the measurement and log data.

All the targeted IP subnets in the three classes of interest were saved in a single "allow list" file for input to ZMap. In accordance with previous research [20], ICMP echo requests were employed as probes. Although ZMap can send probes at a rate of up to 1 Gbps [14], tests of probe rates conducted with network administrators determined that the highest effective rate supported without overwhelming other network traffic was 60 packets/s. When ZMap received a response to a probe, it recorded the responding IP address. Each iteration completed within a variable amount of time, typically five to ten minutes, depending primarily on the numbers of probes sent during an epoch.

Using active probing to identify unresponsive end-hosts required careful consideration. IP address responsiveness is a complex, moving target because end-hosts are naturally cycled on and off the Internet as the devices to which they are attached are moved, and their exact locations are unknown [2]. Therefore, it was difficult to assess how many IP addresses actually existed in a county, how many were typically responsive, how many were responsive prior to an outage, how many were impacted by the outage and how many became responsive after the outage was resolved. To account for these dynamic changes, the end-hosts that responded to all the probes over one-hour each week during a non-outage period were recorded. The corresponding IP addresses were deemed as candidate end-hosts for outages that occurred the same week. If, during an outage period, a response was received from one of the IP addresses, the end-host was considered to be responsive; no response from the IP address led to the end-host being deemed unresponsive.

Another issue was that the probes could be deemed unwanted or even malicious because the packets were sent to IP addresses without the express consent of the administrators. In fact, over more than one year of active probing, only 20 requests to cease the probing of specific IP addresses were received. All the requests were accommodated using ZMap blocklists.

4.3 Deployment

Two important considerations when deploying PowerPing were the selection of measurement vantage points and numbers of probes sent to target IP addresses. Some previous studies have considered these issues [14,20,28,31]. In particular, Wan et al. [46] found that scanning from two vantage points with a single probe increased the network coverage from 95.5% to 98.3%. Additionally, sending two probes instead of one probe increased network coverage from 95.5% to 96.9%.

PowerPing was configured to send one probe from one vantage point to each target IP address during an epoch. This decision could result in false negative responses, but it was made for four reasons. First, since power outages are common events, it is important to limit the impacts of PowerPing probing on the networks. Second, severe power outages that impact wide geographic areas could involve ten million or more end-hosts. Probing such large numbers of end-hosts would push PowerPing up against the 12-min intervals of collection epochs; sending multiple probes would certainly exceed the 12-min collection epochs. Third, there is very little information gain from sending multiple probes instead of a single probe; specifically, network coverage increases from 95.5% for one probe to just 96.9% for two probes. Fourth, Wan et al. [46] observed that vantage points located in the same country as end-hosts have marginally better coverage than vantage points located outside the country and the study described in this chapter only considered end-hosts in the United States.

To verify the design choices, a single server was set up at each of the three CloudLab nodes located at the University of Wisconsin, University of Utah and Clemson University. The servers ran PowerPing to identify power outages and conduct active probing of IP addresses in the impacted U.S. counties. The servers were configured with the same list of IP subnets for each county and were employed simultaneously for one week.

During the testing, differences in the numbers of probe replies received by the servers were observed. Experimentation with different configuration parameters revealed that reducing the ZMap probe rate yielded consistent response rates between the Wisconsin and Utah nodes, but the Clemson node had a consistently lower response rate. However, reducing the ZMap probe rate would increase the time to complete a round of sending probes and processing the responses, limiting the number of IP addresses that could be actively probed during the 12-min epochs.

The difference in active probe network coverage between the CloudLab servers in Wisconsin and Utah was investigated from October 16 through Octo-

Table 2. Network coverage and percentage measurements during power outages.

Network Coverage	Cumulative Percentage (Wisconsin)	Cumulative Percentage (Utah)
99%	90.52%	81.24%
98%	95.66%	93.66%
97%	97.37%	95.64%
96%	97.94%	96.58%
95%	98.24%	97.56%
90%	98.69%	99.74%

ber 25, 2020. During the ten days, each server conducted 10,414 active probe measurements during power outages in 179 counties across 37 states. Consistent with the probing methodology, each server sent a single ICMP probe to each targeted IP address. Three metrics were computed for each county during a measurement period. These included the numbers of IP addresses that responded to each server (R_{wisc} and R_{utah}), total numbers of discrete end-hosts that responded to either server ($R_{total} = R_{wisc} \cup R_{utah}$) and the percentages of end-hosts observed from each server for various network coverage values ($C_{server} = (R_{server}/R_{total}) \times 100$).

Table 2 shows the cumulative percentage measurements taken during outages with indicated network coverage from vantage points at CloudLab sites in Wisconsin and Utah from October 16 through October 25, 2020. In particular, the percentages of end-hosts observed for a network coverage of 97% were $C_{wisc} = 97.37\%$ for CloudLab Wisconsin and $C_{utah} = 95.64\%$ for CloudLab Utah.

Figure 1 shows the total numbers of responses to servers at CloudLab Wisconsin and CloudLab Utah from end-hosts in target counties during power outages from October 16 through October 25, 2020. Specifically, the responses to CloudLab Wisconsin (R_{wisc}) versus the responses to CloudLab Utah (R_{utah}) are plotted for each county for each measurement period to show the consistency across measurements for the two servers. The results demonstrate that less than 4.36% of end-hosts would be expected to be improperly identified as unreachable during more than 97% of measurement periods from a single vantage point. It was posited that this was an acceptable level of uncertainty that would not bias the results significantly because power outages are a common daily occurrence and the study was conducted over a period of 14 months. Furthermore, given the minor differences in response rates, employing multiple vantage points or sending multiple probes to each end-host would be an unnecessary use of Internet resources. As a result, the remaining measurements were conducted using a single server at CloudLab Wisconsin.

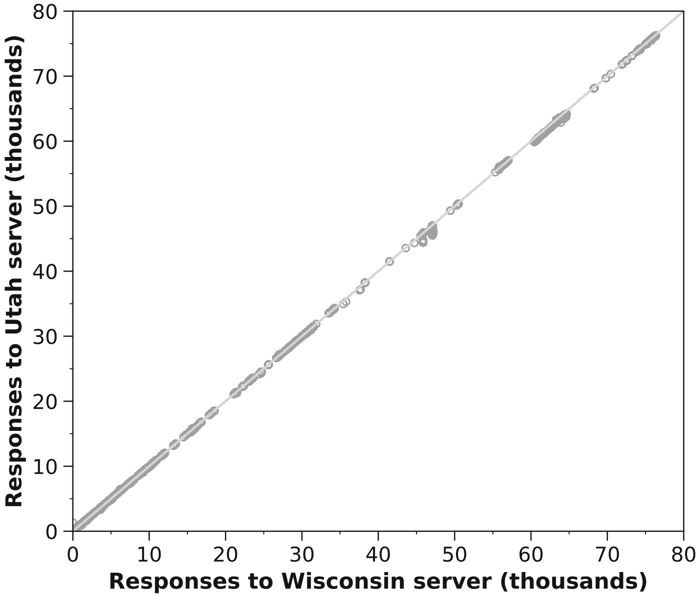


Fig. 1. Total numbers of responses from end-hosts during outages.

4.4 System Design Considerations

While end-hosts require power for operation, there are several reasons why hosts may be reported as responsive during power outages. One reason is the delays between outage occurrences and outage reports on `PowerOutage.us`. Since most power outages are short-lived, they may have already been resolved before they were recognized by PowerPing. Also, the reported numbers of customers affected may not accurately reflect the actual numbers and locations of customers impacted by power outages. For some outages, it was observed that power companies do not update the numbers of customers impacted frequently enough. Instances were routinely observed where the numbers of reported customers with outages did not change, but PowerPing probes had varying response rates. In these instances, the numbers of end-hosts responding to probes may provide more accurate indicators of the numbers of customers without power.

Another reason is that most U.S. counties have multiple power utilities. Although PowerPing determined the number and percentage of customers without power at a given time in a county, it did not distinguish between customers served by different utility providers. Additionally, it was not possible to match individual IP addresses to the utilities that provided power to customers. Active measurements were limited to IP subnets located in counties experiencing power outages, but it was not possible to ascertain that the probed IP addresses belonged to customers impacted by the outages.

Finally, some customers may have used backup power devices such as uninterruptible power supplies for their Internet routers. Internet service providers also maintain redundant power devices and/or backup generators for their network equipment. When customers and Internet service providers utilize backup power during outages, the end-hosts may maintain Internet connectivity during power outages.

While the factors discussed in this section lead to measurement uncertainty, it can be argued that the findings are statistically meaningful because power outages are common events and measurements were collected and analyzed over 14 months. During this time, more than 330,000 outages in 2,495 counties across 48 states were posted on `PowerOutage.us`. Also, by focusing on about 14,000 outages impacting 10% or more customers in counties, nearly all the events with the factors discussed in this section were eliminated. As a result, the negative impact on the findings of this study is expected to be minimal.

5 Results

This section presents the study results that include the characteristics of end-host responsiveness in the absence of power outages (baseline), characteristics of power outages and characteristics of end-host responsiveness during power outages.

5.1 Baseline End-Host Responsiveness

Establishing a baseline of end-host responsiveness in the absence of outages for each U.S. county was essential to the study. The baseline indicates the number of IP addresses as well as the specific IP addresses in each county expected to respond to PowerPing probes. The baseline is employed in the impact and recovery analyses discussed in Sect. 5.3.

The possibility of using existing datasets to identify live end-hosts in subnets was considered. One measurement dataset provides an "IP address space hitlist" upon selecting a single IP address for any /24 subnet to represent all the end-hosts in the subnet [1]. Another dataset provides responsiveness data for hosts running specific services such as HTTP, HTTPS and SSH, but it only collects measurements once a day and does not test responsiveness using ICMP probes [4]. Although these datasets are useful for understanding Internet characteristics at the network subnet and service levels, baseline data was collected during the study due to its focus on individual end-host responsiveness.

Baseline measurements were performed periodically to quantify the responsiveness of end-hosts in each county during non-outage periods. A separate server was set up in the same CloudLab site as the PowerPing server. ZMap was used with the same configuration as the PowerPing server to periodically probe all the IP addresses in each county every ten minutes during a 24-h period. In order to complete probing rounds within ten minutes at a rate of 60,000 packets/s, all the subnets in 100 to 150 counties were selected for probing in each 24-h period. The measurement campaign was conducted for all the counties in the study from August 21 through October 9, 2020.

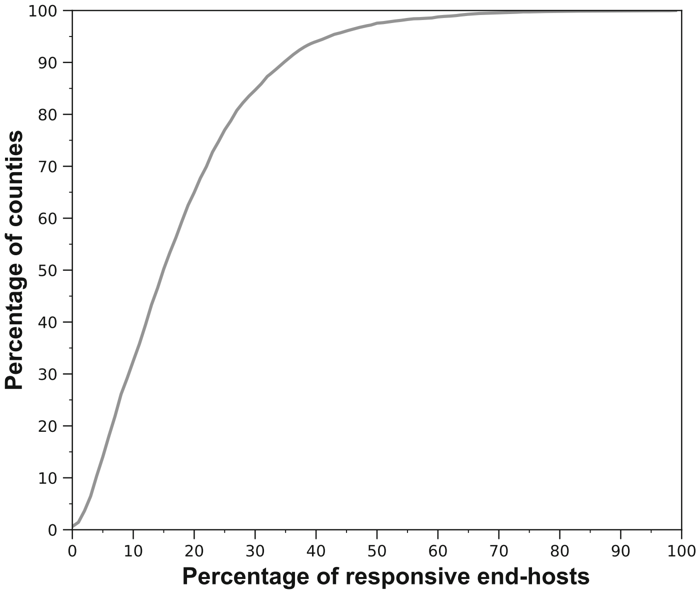


Fig. 2. Cumulative distribution of responsive end-hosts (no outages).

The probing campaign during non-outage periods revealed that the response rate was relatively low in most counties. In an average county, 18.6% of the IP addresses from MaxMind responded to probes. Figure 2 shows the cumulative distribution of the percentages of responsive end-hosts from all the targeted IP addresses in each county during non-outage periods. This is the distribution of responses expected to be received from all the IP addresses in MaxMind. Despite the low response rate, more than 100,000 end-hosts in 183 counties (6% of counties) responded, more than 10,000 end-hosts in 911 counties (29% of counties) responded and more than 1,000 end-hosts in 2,171 counties (70% of counties) responded.

Figure 3 shows a plot of county population from the U.S. Census Bureau versus the number of expected responses from end-hosts for each county during non-outage periods. As expected, the most responses were received from counties with the largest populations: Los Angeles County, California (3.4 million), Cook County, Illinois (1.5 million) and Maricopa County, Arizona (1.2 million). However, the counties with the largest fractions of responses were not associated with the largest metropolitan areas.

Figure 4 (left) shows the distribution of IP addresses in MaxMind by county. Figure 4 (center) shows the numbers of end-host ping responses received. Figure 4 (right) shows the percentages of hosts in MaxMind that responded to target pings.

The numbers of responses received from the counties were consistent over the 24-h measurement periods. ZMap was configured to send one probe to each tar-

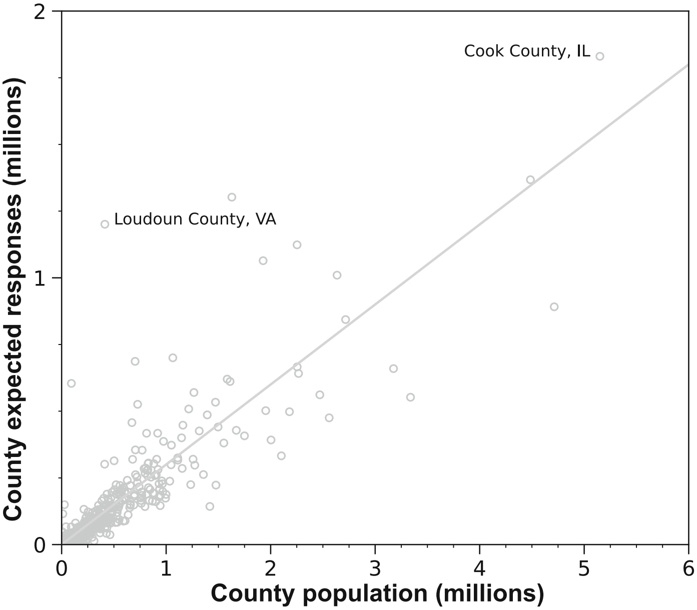


Fig. 3. County population versus expected end-host responses (no outages).

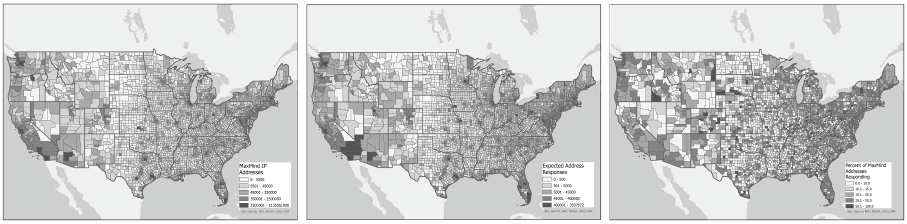


Fig. 4. Geographic distribution of responses by county (no outages).

geted IP address. For this configuration, the ZMap authors measured a 2% single packet loss rate [14]. Figure 5 shows the cumulative distribution of the percentage differences between the maximum and minimum numbers of responses from counties without power outages over the 24-h measurement periods. In 2,766 of 2,987 targeted counties, a difference of 10% or less was measured in the maximum number of responses compared with the minimum number of responses. The differences were less than 2% in 1,391 counties. Diurnal variations in the numbers of responses were not observed.

The baseline of IP address responsiveness was re-evaluated by selecting a uniform random sample of subnets from each county and conducting an additional measurement campaign over a one-month period from March 13 to April 13, 2021. Three to five subnets were selected from the MaxMind dataset for each

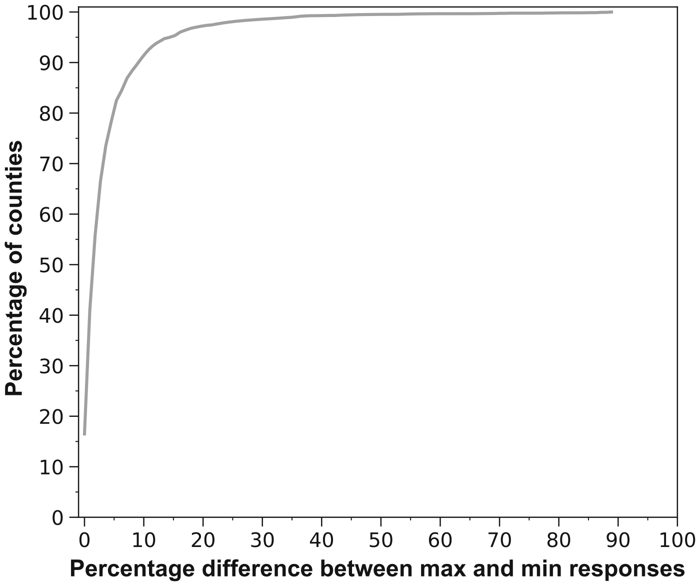


Fig. 5. Distribution of percentage differences in max-min responses (no outages).

county. ZMap was configured to send one ICMP probe to every IP address in the selected subnets every 12 min.

The results obtained during the week of March 14, 2021 were typical of those seen during the additional measurement campaign. During that week, 2,709 counties did not have any power outages that impacted 10% or more customers and 840 active polling iterations were conducted. Although only 5.8% of the end-hosts responding to at least one probe responded to every probe that week, as many as 76.2% of the end-hosts responded to 99% of the probes and 91.3% of end-hosts responded to at least 90% of the probes. Only 7.3% of end-hosts responded to less than 80% of the probes over the entire week. These results indicate consistently high responsiveness levels from IP addresses during non-outage periods.

5.2 Power Outage Characteristics

Power outages are relatively common occurrences and most outages follow distinct cycles. An outage begins with an event that interrupts normal service. Power utilities identify several events that cause outages, the most common being severe weather and motor vehicle accidents. Other events include equipment failures, wildlife interference, high demand, damage from construction work and maintenance [3,10]. An outage is detected by a utility via automated means or customer reports. The utility then deploys the necessary assets to restore power. The outage may be resolved simultaneously for all impacted customers or it may be resolved incrementally for groups of customers.

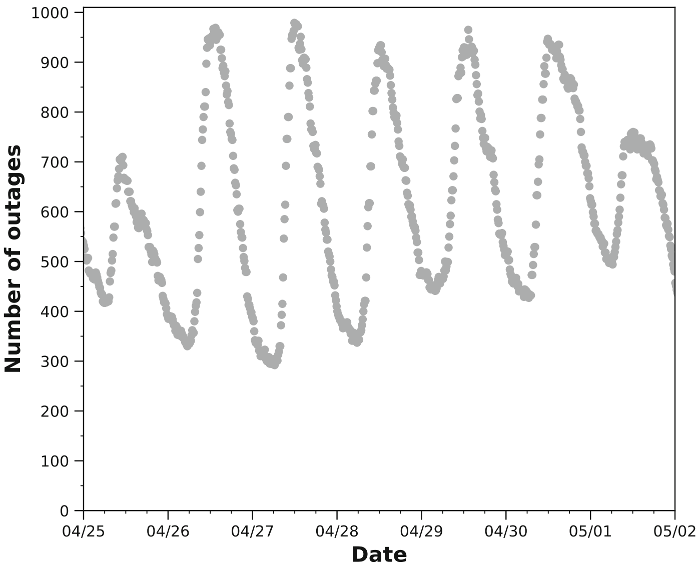


Fig. 6. Power outages detected during measurement epochs over one week.

Most power companies maintain online trackers of known power outages. The online trackers are updated after outages are detected. Additional updates to outages track changing conditions on the ground. Complete resolution of an outage may not be updated on the tracker at the same time there is resolution on the ground.

Figure 6 shows the number of power outages detected in each measurement epoch during the week of April 25, 2021. As shown in the figure, power outages in the United States typically have a strong diurnal pattern. Most outages occur during the early afternoon. A steady increase in the number of reported power outages is seen from early morning until early afternoon. From early afternoon to late evening, a steady decrease is seen in the number of reported outages. The fewest outages occur late at night. This is consistent with previous observations that the majority of power outages are caused by maintenance or operational disturbances, which are more likely to occur during business hours [21]. During the study, fewer power outages were observed on weekends and major holidays. Typically, there were about 50 power outage events per day across the 48 conterminous states that impacted 10% or more customers in a county.

Most outages were short lived – 80% were resolved in under one hour and 90% were resolved in under two hours. A small number of long-duration outages pushed the average outage duration to just under two hours. Figure 7 shows the cumulative distributions of outage durations during each week from September 27 to October 18, 2020. Each outage duration was computed from the time of first report on `PowerOutage.us` to the time the outage was removed from the site.

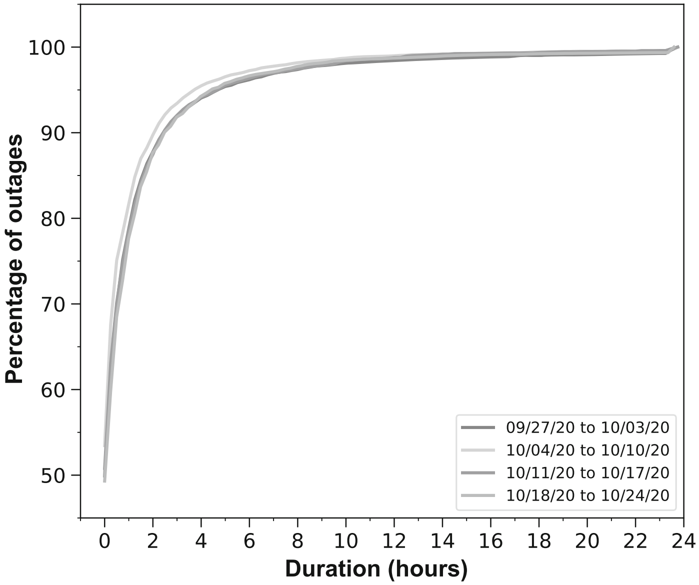


Fig. 7. Cumulative distributions of outage durations over a four-week period.

While the number of power outages follows a consistent diurnal pattern, the study revealed that during most weeks there were strikingly different patterns in the numbers of impacted customers. Figure 8 plots the numbers of customers without power in large counties (top plot), medium counties (middle plot) and small counties (bottom plot) during the week of April 25, 2021. As expected, counties with the largest populations had the most customers without power. The sharp spike in the number of customers without power on the night of April 30, 2021 was due to strong winds and rain that caused power outages along the East Coast [9]. During the study, numerous instances of spikes in the numbers of impacted customers were observed that did not follow diurnal patterns. Also, many counties had small numbers of customers without power (typically fewer than 10) during most probing epochs.

In summary, the study revealed that power outages follow strong diurnal patterns, with most outages occurring on weekday afternoons. Nearly all outages are resolved within an hour. Additionally, the daily numbers of impacted customers have more variations than the daily numbers of outages.

5.3 End-Host Responsiveness During Outages

Two key metrics were identified to assess the impacts of power outages on end-host responsiveness. The first metric is impacts – the percentages of end-host IP addresses (versus the background response rates for counties) that are unresponsive to probes during a power outage. The second is durations – the lengths of

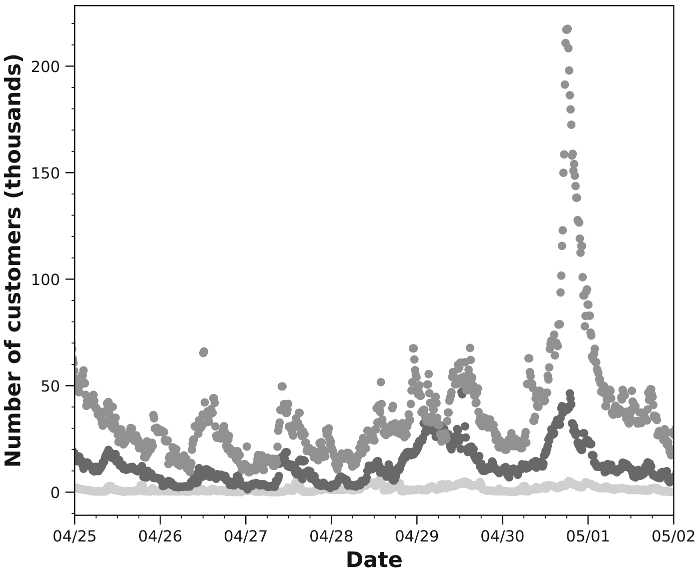


Fig. 8. Customers without power in large, medium and small counties.

time end-host IP addresses in counties are unresponsive during and after power outages.

As far as impacts are concerned, the study revealed that most power outages affect fewer than 1,000 end-hosts. Figure 9 shows the distributions of the numbers of unresponsive end-hosts in counties experiencing power outages for four typical weeks during the study. In most weeks, 80% of the outages affected less than 1,000 end-hosts. The week of October 4, 2020 is a clear outlier. The reason was Hurricane Delta, which struck the Gulf Coast on October 9, 2020, leading to power outages and large numbers of unresponsive end-hosts [33].

A positive correlation was observed between the aggregate numbers of customers without power and aggregate numbers of unresponsive end-hosts during power outages across all counties during each measurement epoch. Specifically, Fig. 10 shows the scatter plot for total customers without power versus total unresponsive end-hosts from February through July 2021 with a correlation $R^2 = 0.99$. However, the correlation results are skewed by the Texas power outages that occurred over four days in February 2021 and impacted up to 4.5 million customers [38].

On shorter timescales (month-long periods), R^2 correlations ranging from 0.19 (April 2021) to 0.99 (February 2021) were obtained. For comparison, a correlation $R^2 = 0.76$ over the same six-month period was obtained when the week of the Texas power outages was excluded. Figure 11 shows the numbers of customers without power (lighter shade) versus numbers of unresponsive end-hosts (darker shade) in counties with major power outages during the week of April 25, 2021. The graph shows an example of temporal variations across all the

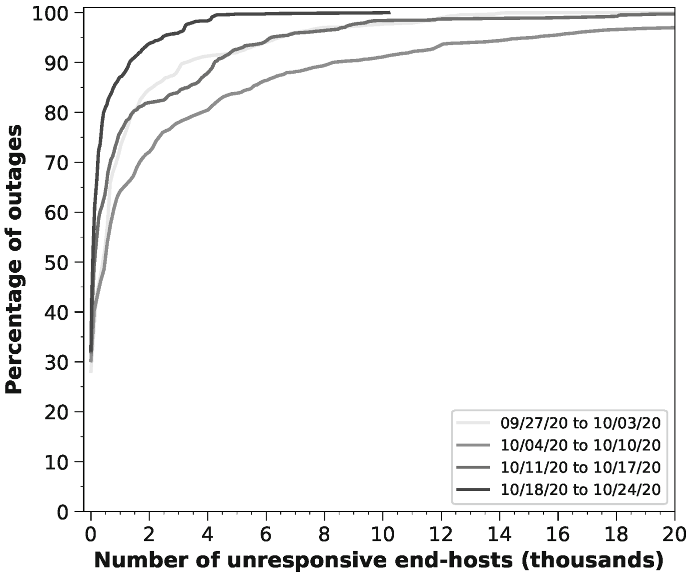


Fig. 9. Cumulative distributions of unresponsive end-hosts in counties.

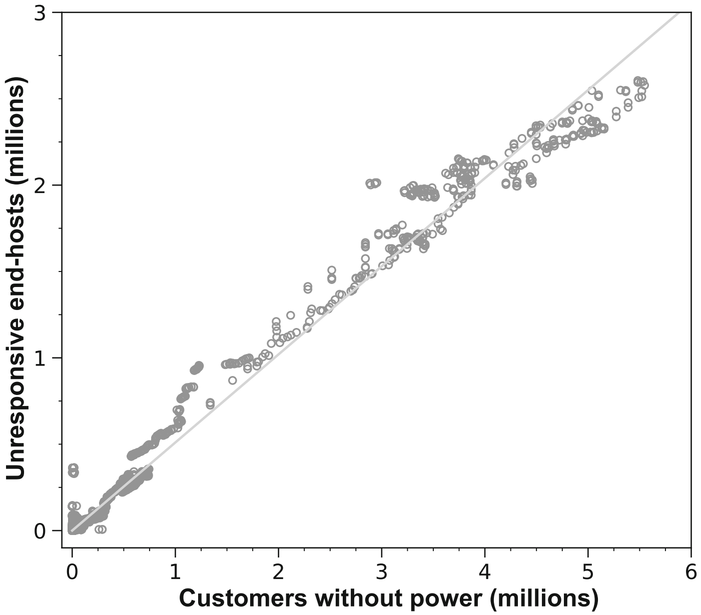


Fig. 10. Customers without power versus unresponsive end-hosts.

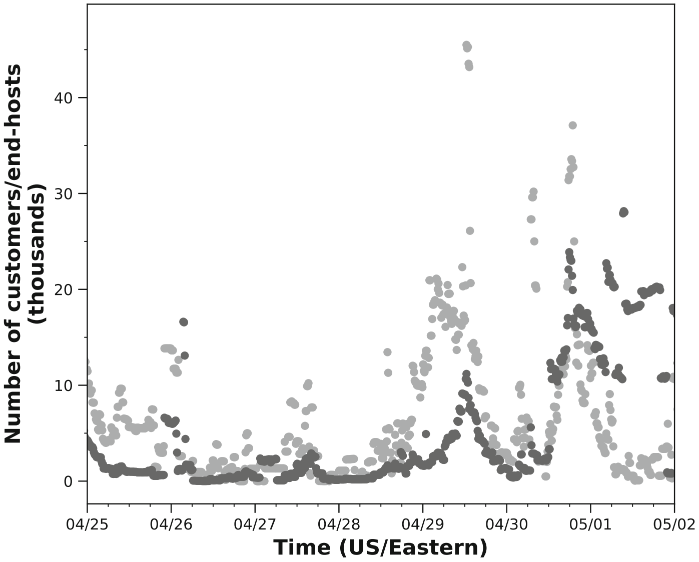


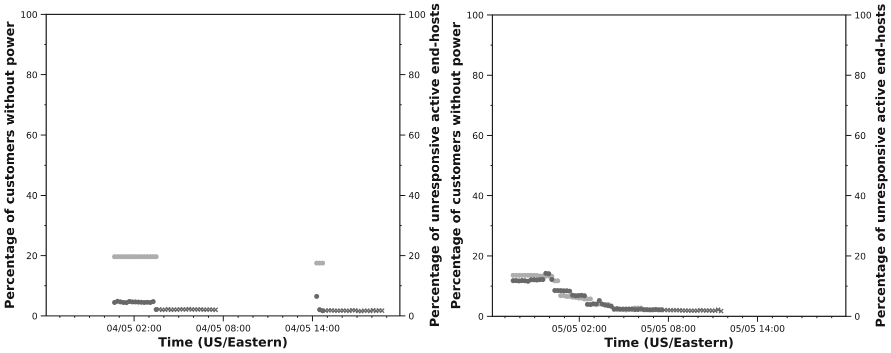
Fig. 11. Customers without power and unresponsive end-hosts (major outages).

major power outages during that week. The number of customers without power corresponds closely with number of unresponsive end-hosts, with the exception of May 1, 2021 due to the Texas power outages.

Unlike at the national aggregate level, it was observed that power outages at the county level often impacted customers without affecting the responsiveness of end-hosts. Across all the power outages from February to July 2021, a correlation $R^2 = 0.66$ was computed for the numbers of customers without power in counties versus the numbers of unresponsive hosts in the corresponding counties. For example, over the week of April 25, 2021, 118 power outages were observed to have increased end-host unresponsiveness during the outages and 98 power outages were observed to have no increase in end-host unresponsiveness.

At the county level, distinct patterns were observed when comparing the percentages of customers without power with the percentages of unresponsive end-hosts. The patterns were placed in four outage classification categories:

- **Category 1:** The percentages of unresponsive end-hosts roughly follow the percentages of customers without power throughout the outages.
- **Category 2:** The percentages of unresponsive end-hosts remain largely unchanged throughout the outages.
- **Category 3:** The percentages of unresponsive end-hosts change smoothly during the collection periods throughout the outages whereas the percentages of customers without power remain constant or undergo frequent large changes.
- **Category 4:** The percentages of unresponsive end-hosts diverge considerably from the percentages of customers without power.



(a) Dawson County, Montana.

(b) Barbour County, Alabama.

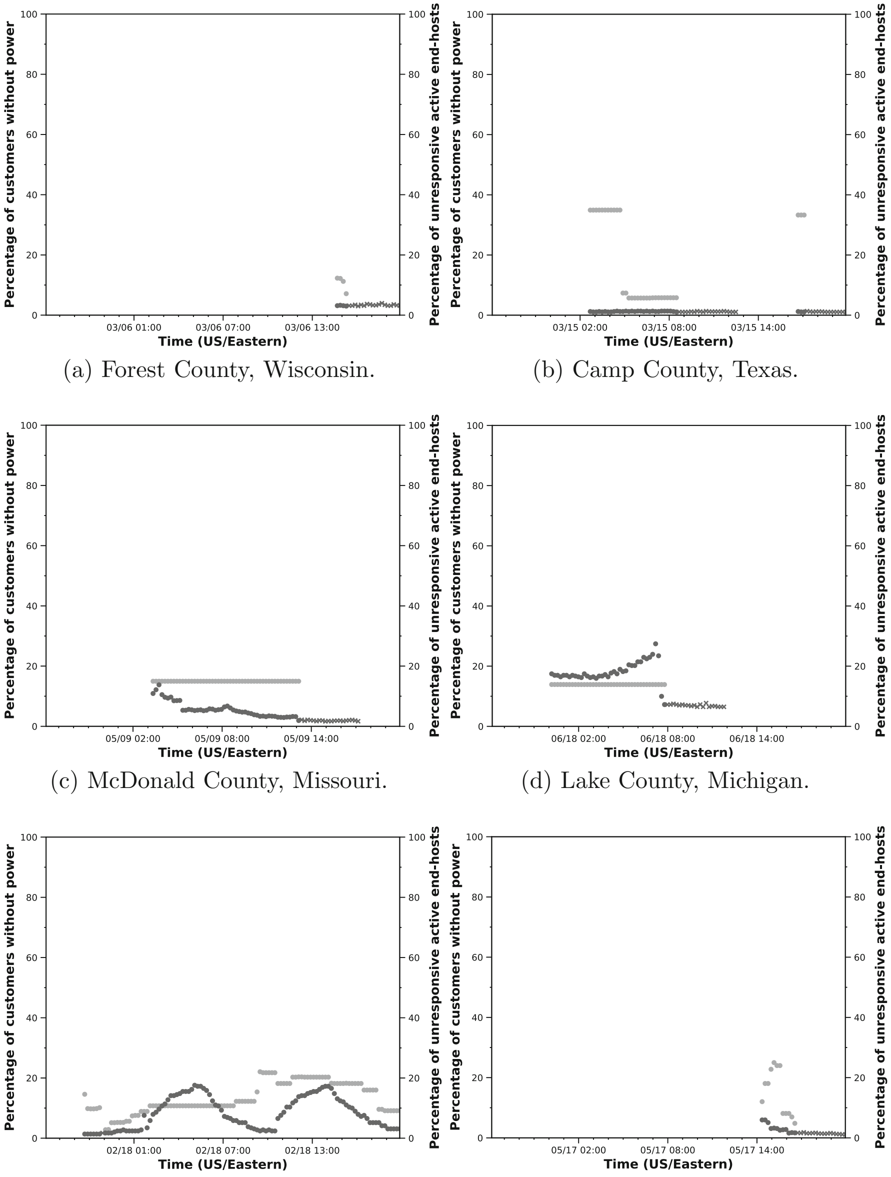
Fig. 12. Category 1 power outages.

The four categories of power outages and their frequencies of occurrence are discussed in the remainder of this section.

Figure 12 compares the percentages of customers without power against the percentages of unresponsive end-hosts when the unresponsive end-hosts closely track customers without power. Specifically, it presents data for two Category 1 outages in Dawson County, Montana on April 5, 2021 and in Barbour, Alabama on May 5, 2021. As the percentage of customers without power in a county increases, the percentage of unresponsive end-hosts increases, and vice versa. This behavior was observed in geographically-distinct areas for counties of various sizes (by area and population) for outages of varying durations and intensities, as well as for counties with different numbers of subnets and expected numbers of end-hosts that respond to active probing.

Figure 13 shows the behaviors of Category 2, 3 and 4 power outages that do not align with the intuitive behavior of Category 1 power outages. Figures 13(a) and (b) present data for Category 2 outages in Forest County, Wisconsin on March 6, 2021 and in Camp County, Texas on March 15, 2021. In the Forest County outage, the percentage of customers without power decreased smoothly from about 15% to 5% over about one hour, but the percentage of unresponsive end-hosts did not vary during or after the outage. Similar behavior is seen in the Camp County outage, where two different outages impacted almost 40% of the power utility customers. However, no effects on the responsiveness of end-hosts were measured during either outage.

Figures 13(c) and (d) present data for Category 3 outages in McDonald County, Missouri on May 6, 2021 and in Lake County, Michigan on June 18, 2021 where the percentages of customers without power stayed almost constant throughout the outages, but the percentages of unresponsive end-hosts varied. The McDonald County outage shown in Fig. 13(c) lasted about ten hours with a constant 18% of customers without power. Towards the beginning of the outage, approximately 12% of end-hosts were unresponsive; the percentage of unresponsive end-hosts decreased to about 5% approximately two hours into the out-



(a) Forest County, Wisconsin.

(b) Camp County, Texas.

(c) McDonald County, Missouri.

(d) Lake County, Michigan.

(e) Carroll County, Mississippi.

(f) Seminole County, Oklahoma.

Fig. 13. Category 2, 3 and 4 power outages.

age, remained near-constant for three hours and then decreased slowly over the remaining five hours of the outage. After the outage was resolved, a consistent percentage of unresponsive end-hosts remained.

Table 3. Occurrences of the four categories of outages in March 2021.

Category	Outages	Percentage
1	402	38.8%
2	423	40.8%
3	34	3.3%
4	177	17.1%

The Lake County outage in Fig. 13(d) was similar to the McDonald County outage, but it impacted an increasing number of end-hosts from 20% gradually up to nearly 30% at the end of the outage. When the outage was reported as resolved, an immediate drop in the percentage of unresponsive end-hosts occurred. In both the Category 2 situations, it is surmised that active probing was a better predictor of customers with outages than what was reported by the utilities. However, as shown in Table 3, Category 3 outages are the least common of the four categories of outages.

Figures 13(e) and (f) present data for Category 4 outages in Carroll County, Mississippi on February 18, 2021 and in Seminole County, Oklahoma on May 17, 2021 where the percentages of customers without power and the percentages of unresponsive end-hosts varied differently, but the metric that most accurately describes the ground situation could not be established definitively. The Carroll County outage had a slowly-changing percentage of customers without power, ranging from hours with a consistent percentage of customers without power, which increased or decreased in distinct steps from 5% to 20% of customers without power. In contrast, the percentage of unresponsive end-hosts rose and fell in two distinct hills that peaked at 5 am and 2 pm.

The Seminole County Category 4 outage in Fig. 13(f) lasted approximately three hours during which the percentage of customers without power slowly rose to almost 30% and then dropped distinctly to less than 10% of customers without power towards the end of the outage. The percentage of unresponsive hosts conveys a different story. A peak of nearly 10% of unresponsive end-hosts occurred at the beginning of the outage, which dropped steadily over the duration of the outage until nearly all the end-hosts became responsive at outage resolution.

To quantify the frequencies of occurrence of the four categories of outages, measurements were conducted for all the outages in all the counties during the month of March 2021. Table 3 lists the occurrences of the four categories of outages during the month of March 2021. Category 1 and 2 outages were most common whereas Category 3 were rare. Although this could not be confirmed, it appears that Category 2 outages are manifested when backup power at the Internet service providers and customer locations helps maintain connectivity during power outages.

The durations of end-host unresponsiveness were also computed during and after power outages. It was determined that more than 80% of end-hosts became responsive to active probing within one hour of power outage resolution and 90%

recovered within two hours. For several long power outages where power was restored to customers incrementally over hours or days, similar increases in the numbers of responsive end-hosts were observed as power was restored.

In summary, the study revealed that most power outages impact the responsiveness of less than 1,000 end-hosts. In aggregate, the numbers of unresponsive end-hosts are closely correlated with the numbers of customers without power. However, the correlation is not as strong at the county level. Additionally, unresponsive end-hosts typically became responsive within two hours of outage resolution.

6 Maintaining Communications During Outages

The study results reveal that power outages are frequent events and often last less than two hours. A natural question is whether or not it is possible for customers to maintain Internet connectivity during power outages.

In order to maintain Internet connectivity, three types of devices or equipment must have alternate power sources: end-host devices (computers, televisions and smartphones), home network equipment (modems and routers) and Internet service provider network equipment. Disruptions of one or more device/equipment types would result in disruptions of Internet connectivity.

Some customer devices, such as laptops and smartphones, have batteries that provide hours of service during power outages. Other customer devices, such as printers, game consoles, smart speakers and televisions, do not. Customers may install their own battery backups for many of these devices.

At this time, no U.S. Government regulations require customer devices and network equipment to have built-in battery backups. However, situations arise where voice (telephone) service continues during power outages while Internet service is lost. This can occur when customer modems and routers have battery backups. Some models provide battery backup for voice service but not Internet service. Customers may take steps to ensure uninterrupted Internet connectivity by installing batteries internal to devices when the options are available or by plugging modems/routers into uninterruptible power supplies. Less common customer solutions involve the installation of residence-level batteries, power generators or solar panels.

Private communication with Internet service providers via `nanog.org` revealed that it is standard practice to deploy various levels of backup power for their equipment. These include battery backups that provide uninterrupted service for short-term outages and backup power generators at their aggregation centers and points of presence. Additionally, Internet service providers may provide short-term (several hours) battery backups for local nodes in residential neighborhoods.

Interruption of power supply to devices or equipment at any of the levels would interrupt Internet service. The disruptions would be inconvenient (e.g., loss of access to online gaming and streaming video), problematic (e.g., inability to conduct online banking, shopping and business communications) or critical (e.g., disrupting access to emergency communications services, news and weather reports and medical devices that require Internet access) [18]. Given the ubiquity of laptops and other consumer devices with batteries, the study findings suggest that the availability of backup power for network devices is not geographically uniform across the United States and end-host connectivity during power outages could be improved with backup power for network devices at Internet service providers as well as at customer residences.

7 Future Work

This empirical study has clarified the relationships existing between power outages and availability of Internet service to end-hosts in the United States. Several opportunities are available for future research. The PowerPing system may be deployed in other geographic areas to assess regional variations in end-host responsiveness. However, the challenge to an expanded geographic scope is that power outage information is not always reported accurately or in a timely manner.

The study indicates that power utilities may not always update outage status in a timely manner. However, given the correlations existing between power outages and Internet service outages, active measurements of end-host service availability is an alternative to obtain more accurate pictures of the prevalence and extent of power outages. This approach would require ground truth power outage data from a source such as `PowerOutage.us` and a careful probing strategy that minimizes network impact.

Important next steps are conducting similar studies for cellular service interruptions during power outages and to include end-hosts with IPv6 addresses. One challenge is that PowerPing could not be directly adapted to these studies. In fact, different techniques and tools would be required to measure outages involving these technologies.

This study has focused on the complete loss of power, but it is important to consider situations where power utilities reduce electricity supply to customers. One type of situation is brownouts, which occur when electricity demand exceeds generation capacity. This study did not measure periods of power brownouts. With an adequate real-time dataset on brownouts, it would be worthwhile evaluate the impacts of brownouts on Internet service.

8 Conclusions

This chapter describes an empirical study on how power outages impact Internet service availability to end-hosts. The PowerPing system was developed to monitor active power outages in the conterminous United States and probe end-hosts in IPv4 subnets geolocated in counties with power outages. During the 14-month study period, more than 330,000 power outages were monitored, including almost 14,000 outages – approximately 50 outages per day – that impacted 10% or more customers in U.S. counties. Most power outages were observed to last less than two hours. In the aggregate, a strong correlation was determined to exist between power outage impact and duration and end-host responsiveness; however, the correlations were found to be weak at the county level. The findings highlight the diverse impacts on Internet connectivity at the county level. The results suggest that providing improved backup power sources for network devices, especially for modems and routers in customer residences, may be adequate for end-hosts to maintain uninterrupted Internet service during typical power outages.

All the code and data described in this chapter are available to the research community upon request. The views and conclusions in this chapter are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, expressed or implied, of the National Science Foundation or the U.S. Government.

Acknowledgments. This research was supported by the National Science Foundation under Grant nos. CNS 1703592 and CNS 2039146. The authors also wish to thank the operators of CloudLab for their support in this research.

References

1. ANT Lab: IP Address Space Hitlists, University of Southern California Information Sciences Institute, Marina del Rey, California (2021). https://ant.isi.edu/datasets/ip_hitlists/format.html
2. Bano, S., et al.: Scanning the Internet for liveness. ACM SIGCOMM Comput. Commun. Rev. **48**(2), 2–9 (2018)
3. Bowen, C.: 8 common causes of outages, Edison International, Rosemead, California, 27 June 2016. https://energized.edison.com/stories/8-common-causes-of-outages
4. Censys: Censys, Ann Arbor, Michigan (2021). https://search.censys.io/
5. Chappell, B.: Zeta causes 2 million power outages, speeds its way into Virginia, National Public Radio, 29 October 2020
6. Cho, K., Pelsser, C., Bush, R., Won, Y.: The Japan earthquake: the impact on traffic and routing observed by a local ISP. In: Proceedings of the Special Workshop on Internet and Disasters (2011). Article no. 2
7. Chrobak, U.: The U.S. has more power outages than any other developed country. Here's why, Popular Science, 17 August 2020
8. CloudLab: CloudLab, University of Utah, Salt Lake City, Utah (2023). https://cloudlab.us/
9. Constantino, A., Small, M.: Downed wires, trees, outages as strong winds sweep through DC area, WTOP News, 1 May 2021
10. Constellation Energy: 10 common causes of power outages, Houston, Texas, 24 September 2021. https://blog.constellation.com/2020/08/21/10-common-causes-of-power-outages
11. Cowie, J., Popescu, A., Underwood, T.: Impact of Hurricane Katrina on Internet Infrastructure. Renesys, Manchester, New Hampshire (2005)
12. Dainotti, A., et al.: Analysis of country-wide Internet outages caused by censorship. IEEE/ACM Trans. Netw. **22**(6), 1964–1977 (2014)

13. Data.gov: U.S. Census Bureau TIGER Dataset, U.S. General Services Administration, Washington, DC (2021). https://catalog.data.gov/dataset
14. Durumeric, Z., Wustrow, E., Halderman, J.: ZMap: fast internet-wide scanning and its security applications. In: Proceedings of the Twenty-Second USENIX Security Symposium, pp. 605–619 (2013)
15. ESRI, ArcGIS, Redlands, California (2023). www.esri.com/en-us/arcgis/about-arcgis/overview
16. Fan, X., Heidemann, J.: Selecting representative IP addresses for Internet topology studies. In: Proceedings of the Tenth ACM SIGCOMM Conference on Internet Measurement, pp. 411–423 (2010)
17. Georgia Power, GPC Outage Map, Atlanta, Georgia (2021). https://outagemap.georgiapower.com
18. Grandhi, S., Plotnick, L., Hiltz, S.: An Internet-less world? Expected impacts of a complete Internet outage with implications for preparedness and design. In: Proceedings of the ACM on Human-Computer Interaction, vol. 4(GROUP) (2020). Article no. 3
19. Guillot, A., et al.: Chocolatine: outage detection for Internet background radiation. In: Proceedings of the Network Traffic Measurement and Analysis Conference, pp. 1–8 (2019)
20. Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., Bannister, J.: Census and survey of the visible Internet. In: Proceedings of the Eighth ACM SIGCOMM Conference on Internet Measurement, pp. 169–182 (2008)
21. Hines, P., Apt, J., Talukdar, S.: Trends in the history of large blackouts in the United States. In: Proceedings of the IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century (2008)
22. Luckie, M.: Scamper: a scalable and extensible packet prober for active measurement of the Internet. In: Proceedings of the Tenth ACM SIGCOMM Conference on Internet Measurement, pp. 239–245 (2010)
23. Mackey, A., Opsahl, K.: Van Buren is a victory against overbroad interpretations of the CFAA and protects security researchers. Electronic Frontier Foundation, San Francisco, California, 17 July 2021
24. MaxMind, GeoLite2 Free Geolocation Data, Massachusetts (2020). https://dev.maxmind.com/geoip/geolite2-free-geolocation-data#accessing-geolite2-free-geolocation-data
25. McWhirter, C.: Delta leaves hundreds of thousands without power. Wall Street J., 10 October 2020
26. Pacific Gas and Electric, PGE Emergency Site - Outage Center, San Francisco, California (2023). https://m.pge.com/#outages
27. Padmanabhan, R., Dhamdhere, A., Aben, E., Claffy, K., Spring, N.: Reasons dynamic addresses change. In: Proceedings of the Internet Measurement Conference, pp. 183–198 (2016)
28. Padmanabhan, R., Schulman, A., Levin, D., Spring, N.: Residential links under the weather. In: Proceedings of the ACM Special Interest Group on Data Communications, pp. 145–158 (2019)
29. PowerOutage.us: About PowerOutage.us, Bluefire Studios, South Portland, Maine (2023). https://poweroutage.us/about
30. PowerOutage.us: United States Power Outage Map, Bluefire Studios, South Portland, Maine (2023). https://poweroutage.us
31. Quan, L., Heidemann, J., Pradkin, Y.: Trinocular: understanding Internet reliability through adaptive probing. SIGCOMM Comput. Commun. Rev. **43**(4), 255–266 (2013)

32. Richter, P., Padmanabhan, R., Spring, N., Berger, A., Clark, D.: Advancing the art of Internet edge outage detection. In: Proceedings of the Internet Measurement Conference, pp. 350–363 (2018)
33. Samenow, J., Livingston, I.: Hurricane Delta by the numbers: 101 mph winds and 9.3-foot surge in coastal Louisiana, Washington Post, 12 October 2020
34. Schulman, A., Spring, N.: Pingin' in the rain. In: Proceedings of the ACM SIGCOMM Conference on Internet Measurement, pp. 19–28 (2011)
35. Shah, A., Fontugne, R., Aben, E., Pelsser, C., Bush, R.: Disco: fast, good and cheap outage detection. In: Proceedings of the Network Traffic Measurement and Analysis Conference (2017)
36. Siegel, A., Grosso, A., Rasch, M., Jarvis, R.: Brief for Amicus Curiae United States Technology Policy Committee of the ACM in Support of Neither Party. Association for Computing Machinery, New York (2020). www.acm.org/binaries/content/assets/public-policy/ustpc-amicus-brief-vanburen-v-us.pdf
37. Sommers, J., Barford, P., Duffield, N., Ron, A.: Multiobjective monitoring for SLA compliance. IEEE/ACM Trans. Netw. **18**(2), 652–665 (2010)
38. Subcommittee on Oversight and Investigations, Power Struggle: Examining the 2021 Texas Grid Failure, Virtual Hearing, Committee on Energy and Commerce, U.S. House of Representatives, One Hundred and Seventeenth Congress, Washington, DC, 24 March 2021. www.congress.gov/117/chrg/CHRG-117hhrg46582/CHRG-117hhrg46582.pdf
39. Syamkumar, M., Mani, S., Durairajan, R., Barford, P., Sommers, J.: Wrinkles in time: detecting Internet-wide events via NTP. In: Proceedings of the IFIP Networking Conference and Workshops, pp. 91–99 (2018)
40. Taylor, D., Diaz, J.: Hundreds of thousands of people are without power, New York Times, 30 August 2021
41. U.S. Court of Appeals for the Ninth Circuit, LinkedIn Corporation, Petitioner v. hiQ Labs Inc, Case no. 17–16783, San Francisco, California (2019). www.supremecourt.gov/docket/docketfiles/html/public/19-1116.html
42. U.S. Energy Information Administration: Electric Power Monthly Washington, DC (2023). www.eia.gov/electricity/monthly
43. U.S. Energy Information Administration: Hourly Electric Grid Monitor, Washington, DC (2023). www.eia.gov/electricity/gridmonitor/dashboard/electric_overview/US48/US48
44. U.S. Supreme Court: Van Buren v. United States, Certiorari to the U.S. Circuit of Appeals of the Eleventh Circuit, Washington, DC, 3 June 2021. www.supremecourt.gov/opinions/20pdf/19-783_k53l.pdf
45. Vigdor, N.: More than 3 million homes and businesses have lost power, New York Times, 15 February 2021
46. Wan, G., et al.: On the origin of scanning: the impact of location on Internet-wide scans. In: Proceedings of the ACM Internet Measurement Conference, pp. 662–679 (2020)

Network and Telecommunications Systems Security

Analyzing Discrepancies in Whole-Network Provenance

Raza Ahmad[1], Aniket Modi[2], Eunjin Jung[3](✉), Carolina de Senne Garcia[4], Hassaan Irshad[5], and Ashish Gehani[6]

[1] DePaul University, Chicago, IL, USA
[2] Indian Institute of Technology Delhi, New Delhi, India
[3] University of San Francisco, San Francisco, CA, USA
ejung@cs.usfca.edu
[4] Google, Zurich, Switzerland
[5] CrowdStrike, Sunnyvale, CA, USA
[6] SRI International, Menlo Park, CA, USA

Abstract. Data provenance describes the origins of a digital object. This information is particularly useful when analyzing distributed workflows because extant tools, such as debuggers and application profilers, do not support tracing through heterogeneous executions that span multiple hosts. In a decentralized system, each host maintains the authoritative record of its own activity in the form of a dependency graph. Reconstructing the provenance of an object may involve the assembly of subgraphs from multiple, independently-administered hosts. The collection of host-specific dependencies coupled with cross-host flows comprise the whole-network provenance, which can grow to terabytes for a small network.

Critical infrastructure assets face constant attacks and despite best efforts, some attacks, such as those leveraging zero-day exploits, succeed. Whole-network provenance has become a common basis for post-attack forensic analyses with the creation of DARPA's Transparent Computing Program. This chapter describes and analyzes aspects of distributed querying, caching and response discrepancy detection used in forensic analyses that are specific to provenance.

Keywords: Distributed Provenance · Data Provenance · Discrepancy Detection

1 Introduction

Provenance collection and analysis are useful for studying distributed applications. These applications may coordinate workflows across multiple interconnected hosts and combine the results [19]. This is important for consortia of institutions that share data and resources for large-scale tasks such as TeraGrid [3] and XSEDE [20]. Provenance metadata from these systems may span multiple administrative domains. These records collected from a single host are

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 93–114, 2024.
https://doi.org/10.1007/978-3-031-49585-4_5

termed whole-system provenance [17]. "Whole-network provenance" is defined as metadata that describes the relationships between whole-system provenance on individual hosts coupled with the set of distributed data flows connecting processes on the hosts.

Whole-network provenance became a common basis for detecting stealthy advanced persistent threats with the creation of DARPA's Transparent Computing Program [5]. Critical infrastructure assets in the form of network-facing services, such as access to code repositories and domain name resolution, may come under attack. Despite best efforts to secure critical infrastructure assets, attacks often succeed and subsequent forensic analyses are of utmost importance to identify the attack vectors and the scopes of the attacks. One aspect of forensic analysis involves querying provenance agents on hosts in a distributed system such as an enterprise or government organization. Systems that collect and analyze whole-network provenance are now being deployed at scale. For example, DISTDET has been installed on more than 22,000 hosts at over 50 industrial customers [6].

In these settings, individual hosts can send queries to other hosts to obtain the full provenance data of an item such as a file downloaded from a remote host. In a decentralized querying approach, each host receives responses from remote hosts to its own queries, but also forwards responses to queries from other hosts as well. Any subset of these responses can be stored in local storage to build a host cache. When a network is too slow or expensive, the host may run a provenance query on its own cache to obtain a preliminary query result.

Provenance metadata collected from remote hosts is not necessarily reliable and trustworthy. Some hosts may have buggy software, some may send outdated data, some may suffer from network fluctuations and some may be malicious. Provenance discrepancy is defined as the difference between truthful provenance and a response received by the querying or intermediate host. Since provenance is a record of the history of computation, the later metadata from a host can have more elements and relationships between the elements than before, but not less. This "append-only" nature of provenance metadata is leveraged to detect and report a discrepancy whenever a query response is missing an element from the previously-known provenance metadata in the cache.

The ability to detect discrepancies from missing graph elements is important in several real-world applications. Four example scenarios include a product failure that exposes a company to legal liabilities in case of forensic analysis, a legal battle over patent infringement by a company to deny prior possession of references, an accident as a result of a computational error, and a claim of credit for a discovery after learning about a competitor's result [8]. These scenarios motivate the alteration of provenance data after an incident has occurred. Data modifications manifest themselves as deletions of old elements and insertions of new elements, which cause discrepancies in provenance data.

2 Background

The open-source SPADE middleware [12] is employed in this study. SPADE supports a number of operating systems for provenance management. In particular, it supports the use of the Linux Audit framework as a source to derive whole-system provenance [17]. However, the ideas in this research apply to any provenance management framework that supports decentralized operation.

A provenance graph $G(V, E)$ contains a set of vertices V and a set of edges E, where edges in E connect vertices in V. Each vertex $v \in V$ corresponds to an agent, process or artifact that is the subject or object of an operation. Each vertex is characterized by a unique key-value set of annotations $A(v)$: $A(v) = \{a_1, a_2, \ldots, a_n\}$ where $a_i = \langle key_i : value_i \rangle$. For example, a vertex representing an operating system process would contain annotations such as $\langle pid : 2 \rangle$, $\langle user : root \rangle$, $\langle time : 1345012 \rangle$. The annotation set is unique because there is only one process with a certain *pid* at a given *time*. Hence, to uniquely identify vertex v with a single attribute, a content-based hash identifier id_v is constructed by hashing the concatenation of all the key-value pairs: $id_v = hash(a_1 \parallel a_2 \parallel \cdots \parallel a_n)$.

Note that any change to a key-value pair results in changing the vertex to a different vertex. For example, if a malicious host changes the time in vertex $v = \{pid : 2, time : t_1\}$ to $\{pid : 2, time : t_2\}$, then the hash identifier would change and v would become a different vertex $v' = \{pid : 2, time : t_2\}$ and the provenance graph $G(V, E)$ would change to $G(V', E)$ where $V' = V \setminus \{v\} \cup \{v'\}$.

An edge in E is an operation on a pair of vertices and corresponds to a directed edge between them, specifying a data dependency. For example, a system `read()` call results in an edge from a process vertex to a file vertex and contains annotations such as $\langle size : 1024 \rangle, \langle time : 1345121 \rangle$. Each edge $e \in E$ is defined by the two vertices, X and Y, on which it is incident, and a set of annotations $A(e)$: $e = \{X, Y, A(e)\}$. Each edge is uniquely identified by a content-based identifier id_e by hashing the concatenation of the identifiers of the incident vertices id_X and id_Y and the elements of the annotation set $A(e)$: $id_e = hash(id_X \parallel id_Y \parallel a_1 \parallel a_2 \parallel \cdots \parallel a_n)$. As with a change to a vertex, any change to an annotation in $A(e)$ results in changing the edge by deleting the original edge and adding a new edge to E.

3 Whole-Network Provenance

Whole-network provenance is formally defined as the metadata that describes the intra-host whole-system provenance of each host in the network coupled with the inter-host flows between pairs of hosts. Using whole-network provenance graphs, the provenance of an object can be reconstructed by starting from one host and tracking back through other relevant hosts.

The provenance graph on a host H_i is defined as $G_{H_i} = (V_{H_i}, E_{H_i})$. The inter-host flow created between two hosts H_i and H_j is given by the tuple of network artifacts connecting them:

$$F_{i,j} = (n_i, n_j) : n_i \in G_{H_i}, n_j \in G_{H_j}, i \neq j, n_i = n_j$$

where n_i is the network artifact vertex on host H_i.

The whole-network provenance graph is defined as:

$$G_{network} = \bigcup_{i} G_{H_i} \cup \bigcup_{i,j,i \neq j} F_{i,j}$$

where H_i is a host on the network and $F_{i,j}$ is a flow between two hosts H_i and H_j on the network.

In a centralized strategy, each host uploads its own provenance metadata periodically to a single repository that handles all provenance queries. This approach simplifies the coordination between hosts, but suffers from three limitations. First, all hosts in a network are required to periodically send all their provenance metadata to the central repository, although other hosts may not need much of it. Second, the central repository may become a performance bottleneck, especially in terms of bandwidth because simultaneous uploads from multiple hosts may render it unavailable for processing queries. Third, the reliability of the entire system decreases because the central repository becomes a single point of failure. Note that a data integrity compromise at the repository can affect the provenance metadata of the entire network.

The proposed approach employs a decentralized, peer-to-peer architecture. Each connected host in the network is independently responsible for collecting and storing its own metadata. Individual hosts can completely satisfy all local queries. They may also collect provenance metadata by querying other hosts in the network. The querying host then combines all the responses from the remote hosts.

This mechanism provides a scalable approach for whole-network provenance collection because it does not have the aforementioned limitations of a centralized approach. The mechanism also has four benefits. First, less resources are required per host – no single host is required to have sufficient resources to maintain complete copies of provenance from all hosts. Second, there is no wasted data transfer – all the transferred data is necessary to respond to specific queries. Third, there is resilience to network fluctuations – individual hosts can use their own caches to answer queries in the case of network instability. Fourth, individual hosts have the freedom to implement their own data management policies, such as the database to use and the retention period of archival copies.

At the heart of this decentralized metadata collection is a construct called the network artifact [9,12]. Its key property is that it can be constructed without any explicit coordination at independent endpoints. In the context of a distributed system, a pair of network artifacts indicates a data flow between two hosts. For operating system provenance, network artifacts are constructed using the IP addresses and ports of the endpoints, combined with the times when the connections were established.

4 Distributed Querying

In a distributed, decentralized environment, the host that originates a query is responsible for collecting its responses. After resolving the query locally, the host contacts remote hosts through network artifacts that subsequently return their results and contact other hosts if required. The responses are stitched together at the originating host to create a single connected provenance graph. This approach enables remote hosts located the same distance away in the network to be contacted in parallel. Thus, the distributed querying time increases linearly with the height of the network topology tree regardless of the number of remote hosts.

A provenance management system that operates in a distributed environment may collect provenance metadata across several hosts. Two of the most common operations in collecting provenance are lineage and path queries. The lineage of an item traces its past (ancestors) or future impact (descendants). The response to a lineage query is a directed graph. Lineage queries are sent with a maximum depth d to limit the retrieved provenance because the size of a provenance graph could grow rapidly over multiple hosts.

To formally define a lineage ancestor query from a vertex v for depth d, it is necessary to first define the parent graph of v: $G_P(v) = (P, E)$, where P is a set of vertices such that $\forall p \in P$, an edge $e \in E$ exists and $e = (v, p)$. The lineage of v is given by:

$$\begin{aligned} &l(v,d) = G_P(v) \cup l(p, d-1) \;\; \forall p \in P \\ &l(v,0) = v \end{aligned}$$

The response to a lineage query is always a connected graph in which the directions of edges represent the information flow. Thus, given a graph $G_{response}$ sent in response to a lineage query q from vertex v, $\forall u \in G_{response}$, a path exists between any two vertices:

$$\begin{aligned} &\exists\; u \rightsquigarrow v && \text{(descendant query)} \\ &\exists\; v \rightsquigarrow u && \text{(ancestor query)} \end{aligned}$$

Also, $\forall e = (x, y) \in G_{response}$:

$$\begin{aligned} &x, y \in G_{response} \;\; \wedge \;\; \exists\; y \rightsquigarrow v && \text{(descendant query)} \\ &x, y \in G_{response} \;\; \wedge \;\; \exists\; v \rightsquigarrow x && \text{(ancestor query)} \end{aligned}$$

A path query requests the provenance between two objects. Its response is a set of chains from one element to another. The response to a path query is constructed by finding the intersection of lineage ancestor queries from the sink and lineage descendant queries from the source when obtaining all the paths from a particular source to a sink.

When a host needs to see the history of an artifact (e.g., downloaded file) – specifically, where the artifact originated and when and how it was changed before arriving at the host – the host may send a lineage ancestors query to its upstream hosts. The term query host refers to the host from which the lineage inquiry originates.

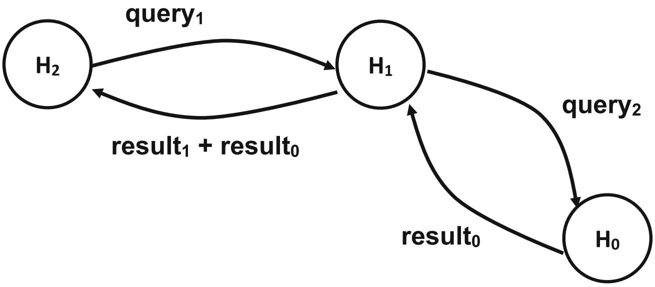


Fig. 1. Interconnected hosts querying provenance in a distributed manner.

Figure 1 shows a network of three interconnected hosts where H_2 is the query host, H_1 is the intermediate host and H_0 is the source host. In this case, H_2 wishes to find the lineage of file f_2 on H_2 and learns that the file was downloaded from H_1. H_2 becomes the query host and sends $query_1$ to the upstream host H_1 requesting for provenance metadata of file f_2. H_1 observes that the provenance of f_2 on H_1 continues to H_0. This could happen in one of two cases – f_2 could have been downloaded from H_0 or the process that modified f_2 could have been involved in a network connection between H_1 and H_0. At this point, H_1 becomes the intermediate host and sends $query_2$ to the next upstream host H_0 requesting the provenance metadata of file f_2. If f_2 originated from H_0, then H_0 is the source host and it responds with $result_0$.

The origin and type of a query implicitly define whether one host is upstream or downstream of another. When a query is performed at H_2 about metadata that originated from H_1, H_1 is upstream of H_2 in the context of a lineage ancestors query (and its response). Similarly, H_2 is downstream of H_1 in this context.

However, the converse holds for a lineage descendants query. Specifically, if the query is targeted at host H_1 about metadata that flowed from the host to H_2, then H_2 would be upstream of H_1. Of course, the same pair of hosts could be upstream of each other in the context of different queries. In the rest of this chapter, lineage query is used as shorthand for a lineage ancestors query or a lineage descendants query, where the precise meaning is determined by the context.

5 Caching

It is assumed that each host manages its own cache of provenance metadata from remote hosts. Using cached data to save bandwidth and reduce latency is a common practice in distributed systems. Provenance metadata benefits from

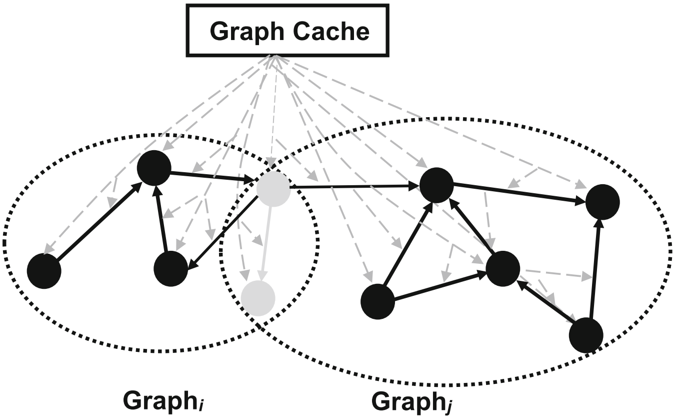


Fig. 2. Cache containing responses to two queries with partial overlap.

similar approaches [11]. When a host receives a response from an upstream host – as a querying host or intermediate host – the host adds the response to its cache. Each response is stored as a directed graph, so the cache is essentially a set of directed graphs.

When a host has a lineage or path query that involves remote hosts, the cache can be also used to obtain a (potentially outdated) local response when communications between the network and other hosts are not reliable or too expensive, and also when low latency is more important than freshness. This cache is denoted as G_{cache} because it contains provenance graphs created from previously-received query responses from other hosts.

Figure 2 shows an example graph cache containing two previously-received query responses, $Graph_i$ and $Graph_j$. The shaded vertices and edges are shared by both graphs and stored only once to save memory. When the response to a query overlaps with the existing cache (even if the query is sent for the first time), the G_{cache} of the host is used to detect discrepancies. The cache has pointers to all the vertices and edges in the graphs it contains. This enables searches of the union of all the graphs in the cache.

Merging a new response $G_{response}$ with the existing cache G_{cache} without redundancy starts by identifying the intersection of sets G_{cache} and $G_{response}$. One approach for computing $G_{cache} \cap G_{response}$ is to construct a bijection between the graphs using McKay's algorithm [15]. However, this requires the construction of a canonical form that requires $O(2^n)$ time, where $n = |G_{cache} \cup G_{response}|$. Therefore, an alternative approach that leverages provenance metadata represented as a property graph is employed.

All vertices and edges have content-based identifiers as described in Sect. 2. Specifically, the identifier of a vertex is computed by hashing the catenation of the sorted set of annotations associated with the vertex. In the case of an edge, the hash takes as input the identifiers of the two endpoint vertices and the annotations associated with the edge; the resulting hash is the identifier of

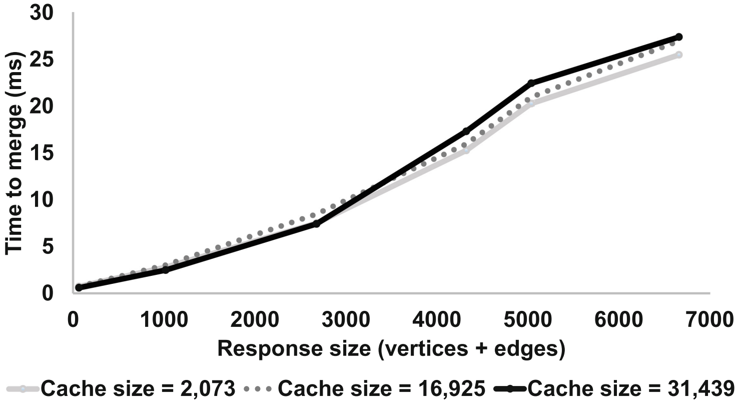


Fig. 3. Impact of response size on merging time in the graph cache.

the edge. In this setting, the problem is reduced to sorting the identifiers of the vertices and edges of each graph. The intersection of the two graphs contains the elements present in both sorted sets. The operations can be performed in linear time by traversing the two sorted sets in lockstep.

Figure 3 shows the linear relation between response size and time taken to merge responses into a fixed-size cache for varying cache sizes (numbers of vertices and edges). This is significant because larger cache sizes do not increase the merge time significantly.

Figure 4 shows the querying and discrepancy detection workflow. The analyzer module in host H_2 acts as a query manager:

- The analyzer module receives a query from a user, sends it to the local query module and receives the response G_{local}.
- If the local query module indicates that a remote host needs to be consulted, the analyzer prepares a remote query and sends it to H_1, which responds with a provenance graph $G_{response}$.
- The analyzer checks the signature of $G_{response}$. If the signature is valid, it forwards $G_{response}$ to the discrepancy checker, which returns the discrepancy count *dc*.
- If the discrepancy *dc* is zero, $G_{response}$ is added to the graph cache G_{cache} and is shown to the user along with G_{local}. Otherwise, the discrepancy checker reports *dc* to the analyzer. It is important to note that the discrepancy count is proportional to the number of different discrepancies detected.

5.1 Eviction Policy

The cumulative metadata can grow very large in an environment when whole-network provenance is being collected, For example, during the DARPA Transparent Computing engagements [5], terabytes of provenance records were col-

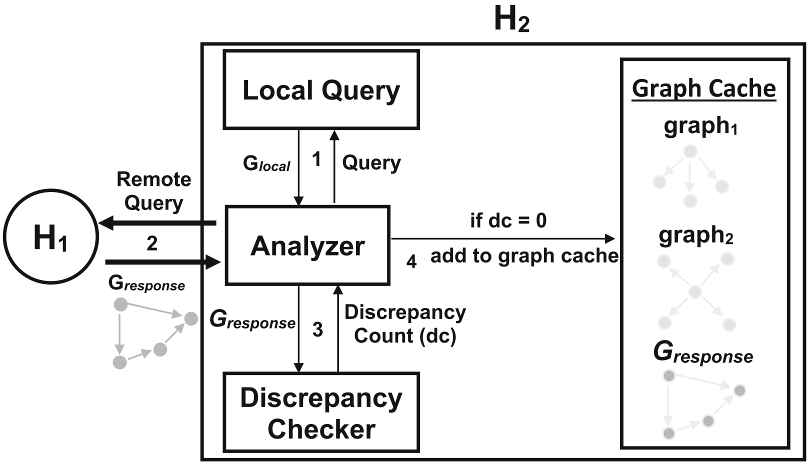


Fig. 4. Querying and discrepancy detection workflow.

lected from a small network. If all provenance queries are resolved across a distributed system and their responses are cached at the intermediate and original querying hosts, the metadata would increase monotonically with a large storage overhead.

One way to keep the cache size from growing arbitrarily is to implement an eviction policy. Such a policy can be framed at the granularity of individual graph elements, similar to previous approaches for distributed provenance cache management [10]. However, this leads to two shortcomings. First, if individual vertices and edges are removed from a provenance graph, the graph may become disconnected. This would violate the property that a provenance graph obtained from a lineage query is a single, connected graph (as described in Sect. 4). Second, evicting an element from the intersection of a new response and previously-cached responses is indistinguishable from the case where the response contains a discrepancy.

If an old response G exists such that $G \subset G_{cache}$, then the host can discard G without loss of information. However this requires old responses to be evaluated periodically, which would increase the time complexity of cache management. Instead, a provenance-aware first-in first-out (FIFO) eviction policy is employed that removes the complete response graph components from the cache instead of individual graph elements.

Measurements of the impact of the eviction policy on the number of detected discrepancies shows a clear trade-off between the cache size and effectiveness of discrepancy detection. This was accomplished by executing a series of queries $q_1, q_2, \ldots, q_n$ and adding their responses $r_1, r_2, \ldots, r_n$ to the cache in the same order. The responses were removed one by one to reduce the cache size. However, before and after removing a response r_i, query q_i was sent again and a fixed number of edges and vertices in the response was deleted. This enabled the

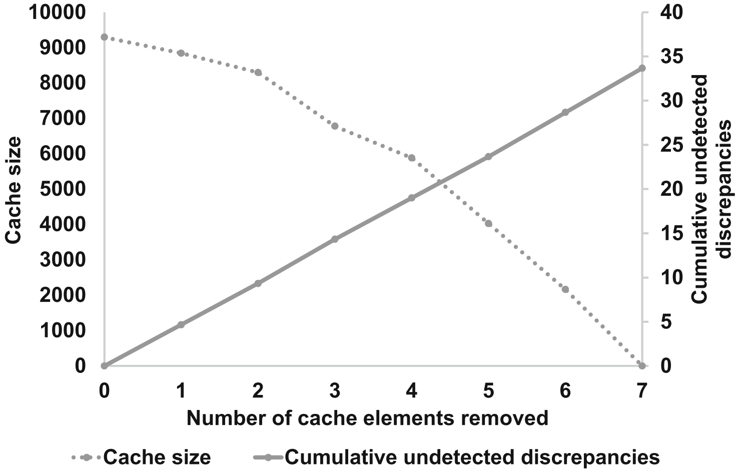


Fig. 5. Eviction policy impact on number of undetected discrepancies.

measurement of the number of discrepancies that went undetected when r_i was absent from the cache.

Figure 5 shows the impact of the FIFO eviction policy on cache size (number of vertices and edges) and the number of discrepancies that go undetected. In the beginning, the cache contains seven graph responses and there is no eviction. As a result, the number of detected discrepancies at the time is also the maximum. As cache elements are removed one by one, the cache size decreases and the number of discrepancies that go undetected increase. When all seven graphs in the cache are removed, no discrepancy is detected by the algorithm because there is nothing left in the cache to compare with the new query response.

5.2 Graph Storage

A provenance graph can be stored in any way that a directed graph with annotations is stored. For example, SPADE [12] provides the Postgres relational database, Neo4j graph database and Apache Kafka streams as storage options. While storing the entire graph provides the most information to detect a discrepancy, the storage required grows rapidly. In fact, when using TRACE data sets, the storage required grew by approximately 1 GB per hour [13].

The rapid growth not only consumes storage, but also network bandwidth. If the cache is built by periodically circulating the provenance graph from each host, the rapid metadata growth would burden the storage of every host and every connection between hosts in the network.

Instead of storing the entire graph, a Bloom filter may be used to store the vertex and edge identifiers. Discrepancy detection relies on membership tests, that is, checking if a certain vertex or edge is in a particular provenance graph.

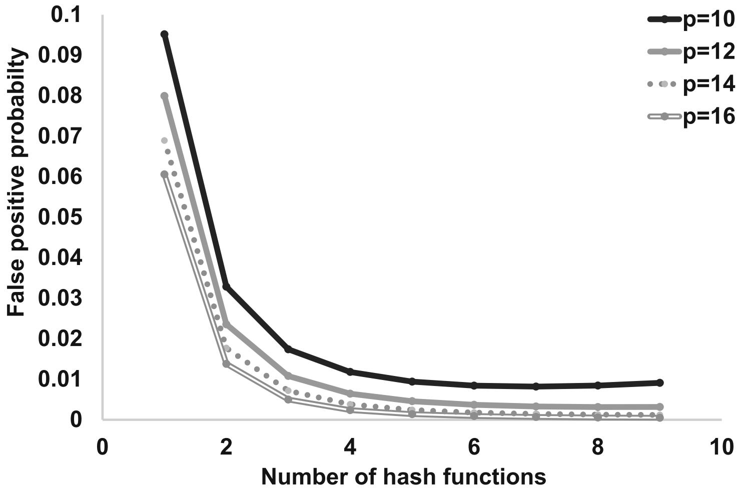


Fig. 6. False positive rates for varying numbers of hash functions.

A Bloom filter offers a trade-off between space (and bandwidth) and the false positive response rate.

Figure 6 shows how the probability of returning a false positive in the membership test changes as more hash functions are employed for varying p, which is the ratio between the size of the Bloom filter m and the number of elements (vertices or edges) n. As more hash functions are used, the false positive rate quickly decreases and then plateaus.

If the host periodically circulates the changes in the provenance graph to update the whole-network provenance stored at each host, the Bloom filter could contain only the newly added vertices and edges created since the last Bloom filter was sent. Each host could keep the Bloom filters separately in its cache or merge a subset. Merging the Bloom filters saves space and also reduces the time complexity of the membership test in discrepancy detection.

Figure 7 shows that merging Bloom filters increases the false positive rate. When the ratio of the Bloom filter size to the number of elements p is 100 and nine hash functions are used, merging ten Bloom filters resulted in a 1% false positive rate. The ratio p and number of hash functions k can be selected to minimize the false positive rate in the merged Bloom filters.

6 Discrepancy Detection

A provenance discrepancy is defined as the difference between truthful provenance and a response received by a querying or intermediate host. A host may have experienced an overwhelming workload and omitted some provenance metadata or it may have replayed an old response from another host. Upon getting such a response, the receiving host could detect a discrepancy if the discrepancy occurred in any of the previously-received responses.

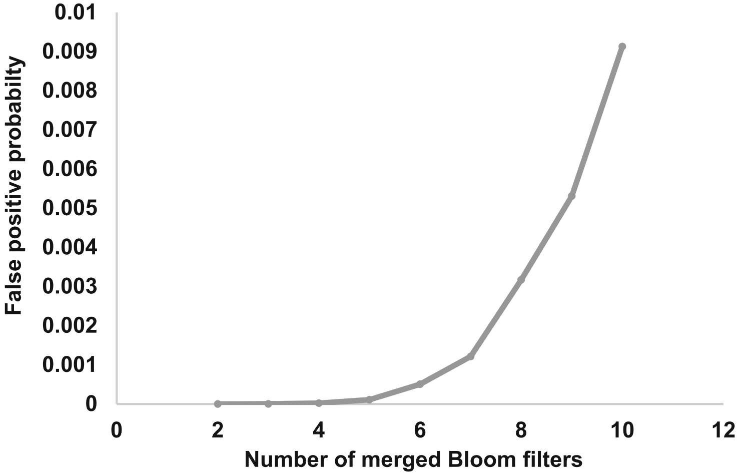


Fig. 7. False positive rate using merged Bloom filters.

Before the query host uses the provenance metadata it received from upstream hosts, it has to verify the authenticity and integrity of the received data. It is assumed that every host has the public keys of other remote hosts and that the response from each host is digitally signed using the private key of the host. The query nodes can check cryptographic signatures to detect if the intermediate nodes modified the metadata from upstream nodes before forwarding them to the downstream nodes. However, when any host fabricates its own provenance metadata, it can also provide a proper signature for the fraudulent metadata. The query host would not be able to detect this attack using the cryptographic signatures. Similarly, when an intermediate host replays a previously-received response from its upstream hosts, the cryptographic signature would still verify normally and the query host would not be able to detect that the response is outdated.

Whole-network provenance is typically inferred based on records originating from the kernel; this is due to multiple reasons, including the global view available and the higher bar for tampering. Consequently, in practice, the primary threat to the soundness of the provenance being reported is the loss of records along the data path from the occurrence of the relevant event to persistent storage. A missing record can translate to a variety of effects in the provenance stream, the simplest of which is a missing instance of a relation.

6.1 Threat Model

The threat model comprises two attacks on the desired properties. Note that any provenance metadata given as a response to a remote query could be affected by one or more of these attacks.

Omission Attack on Integrity. In this attack, a source or intermediate host provides fabricated metadata by deleting or modifying its own provenance metadata. The fabrication may be intentional or it may be due to network fluctuations, errors or software bugs.

As an example, assume that H_1 has experienced an overwhelming workload and failed to record some of its own provenance metadata in persistent storage. Also, H_1 may have previously provided a truthful response to a query from H_2. The result would be equivalent to modifying or deleting an element from a truthful provenance graph. The discrepancy detection approach does not require that the same query that gave rise to the fraudulent response had to be performed earlier. The discrepancy would be detected as long as the deletion in the fraudulent response is in the portion that overlaps with an earlier truthful response to a query.

Replay Attack on Freshness. In this attack, an intermediate host resends (replays) a previously-received response to a downstream host containing outdated provenance metadata from an upstream host. For example, H_1 in Fig. 1 may not forward $query_2$ to H_0 and repeat an old response from H_0 to H_2 to save computing and network resources. Note that H_1 cannot modify or produce a fraudulent $result_0$ without H_2 detecting it because of the cryptographic signature.

The threat model does not include the case where a remote host only adds fraudulent data to the authentic provenance metadata in a monotonically increasing manner. Consider a case where a remote host adds the same fraudulent provenance metadata in addition to the authentic data to all the responses it generates. In this case, all the other hosts would not be able to tell if the remote host is lying because the cryptographic signature would be valid and all the responses would be consistent with each other. From a user's standpoint, there is no difference between such an addition and a valid insertion to the provenance graph.

6.2 Omission Attack Detection

A discrepancy in a whole-network provenance graph G' is defined as an invalid modification of the topology (modifying or deleting a vertex or edge) or schema specifications (changing the annotations of a vertex or edge) of G, where G is a truthful response to a provenance query. It is important to note that, when an adversary changes the schema specifications, it appears as if the adversary deleted a vertex or an edge in G and added a new one to it. In other words, all discrepancies appear as deletions and/or additions of vertices and edges in a whole-network provenance graph.

The proposed scheme detects if any vertices and/or edges present in the previous responses (i.e., G_{cache}) are deleted in a later response (i.e., $G_{response}$). More specifically, Algorithm 1 computes the discrepancy count dc defined as the number of vertices and edges missing from $G_{response}$, number of dangling edges

Algorithm 1: Discrepancy detection algorithm.

```
Data: $G_{cache} = \cup_{\forall t < t_r} G(t)$, $G_{response} = G(t_r)$: Provenance graphs;
$d_{max}(G_{response})$: Maximum lineage query depth of $G_{response}$ computed via
breadth-first search
Result: $dc$: Discrepancy count in $G_{response}$
$C \leftarrow 0$
/* Count missing vertices                                                    */
for each vertex $X \in G_{cache}$ and $X \notin G_{response}$ do
    if $d(X) < d_{max}(G_{response})$ then
        $C \leftarrow C + 1$
    end
end
/* Count missing edges                                                       */
for each edge $e = (X, Y)$ such that $e \in G_{cache}$ and $e \notin G_{response}$ do
    if $d(X) < d_{max}(G_{response})$ then
        $C \leftarrow C + 1$
    end
end
/* Count dangling edges                                                      */
for each edge $e = (X, Y) \in G_{response}$ do
    if $X \notin G_{response}$ or $Y \notin G_{response}$ then
        $C \leftarrow C + 1$
    end
end
/* Count dangling vertices                                                   */
for each vertex $X \in G_{response}$ do
    if $\nexists\ e = (A, B)$ such that $X = B$ then
        $C \leftarrow C + 1$
    end
end
return $dc$
```

(both incident vertices are not in $G_{response}$) and number of dangling vertices (no incoming edges in $G_{response}$). More formally, if $G_{cache} = (V_c, E_c)$ and $G_{response} = (V_r, E_r)$, then the discrepancy count is given by:

$$dc = |V_c \setminus V_r| + |E_c \setminus E_r| + |\{e = (X, Y) \in E_r | (X \notin V_r) \vee (Y \notin V_r)\}| \\ + |\{X \in V_r | \nexists (Y, X) \in E_r, \forall Y \in V_r\}|$$

6.3 Empirical Analysis

The empirical analysis employed a small experimental network comprising two hosts, H_1 and H_2, with H_1 the source host of file f. File f was transferred via `scp` to H_2, which generated a provenance trace. During the transfer, both the hosts constructed provenance graphs of their internal system activity. H_1's provenance graph comprised 36,612 vertices and 126,999 edges whereas H_2's provenance graph comprised 128,119 vertices and 446,098 edges.

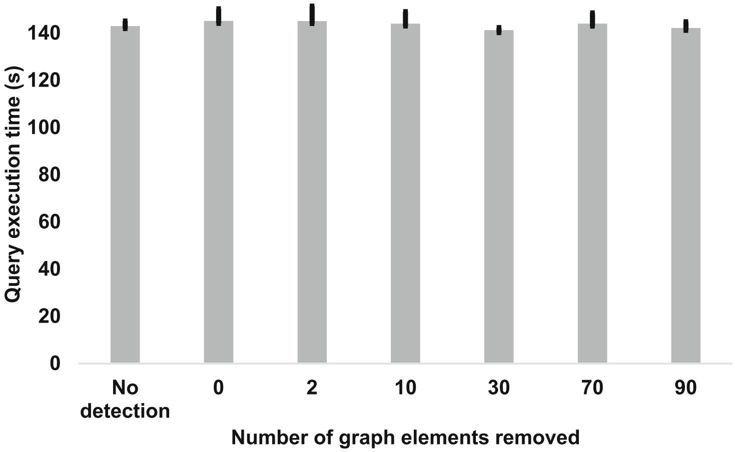


Fig. 8. Query execution time with discrepancy detection for a lineage query.

To track the descendants of file f, H_1 sent a lineage query q with a maximum lineage depth of eight. It originated from f and traveled to H_2, which returned the response graph $G_{response}$. Next, the algorithm executed on H_1 and returned the discrepancy count by comparing $G_{response}$ with G_{cache}. The final result to the query q included graphs G_{local} and $G_{response}$. G_{local} comprised 2,283 vertices and 3,740 edges from H_1 whereas $G_{response}$ comprised 327 vertices and 404 edges from H_2.

The query execution time was measured as starting when H_1 sent q until H_1 completely executed the discrepancy detection algorithm. To evaluate the algorithm overhead at the query host H_1, the baseline performance was first established by measuring the query execution time for q without the detection algorithm in place. Several independent iterations of the query q were executed with the detection algorithm, each with varying numbers of modifications to the response graph. The modifications were induced by dropping the same number of vertices and edges from $G_{response}$, where the number of dropped vertices ranged from zero to 90.

Figure 8 shows that the query execution time did not change significantly with the number of eliminated graph elements. No detection refers to the case when discrepancy checking was not executed. The algorithm imposed less than 0.4% overhead over the baseline. In fact, the query execution time without the algorithm (no detection) is comparable to the case where 90 vertices and 90 edges were removed. This is because most of the query execution time is attributed to the network latency between hosts.

6.4 Replay Attack Detection

The query and response structures were modified to include unique, unpredictable nonces chosen by the query host. When the query host issues a remote query, it sends a new nonce along with the query. Malicious intermediate hosts may choose not to forward the entire query and cause the query host to time out, but they cannot fabricate a response from upstream hosts with the matching nonce. The upstream and source hosts respond with their own provenance metadata along with a nonce and signature computed over their provenance metadata and nonces. The downstream and query hosts discard responses that do not contain valid cryptographic signatures for the (query, nonce) pairs. This can increase the overhead at the query host because it needs to keep track of the (query, nonce) pair until it receives all the responses. However, a timeout was introduced at the query host so it would discard the (query, nonce) pair after waiting for a certain time period.

Note that this mechanism does not interfere with the ability of the query host to use its own cache to answer a remote query, but it clearly does not allow an intermediate host to reuse responses from its own cache because the nonce would not match. The querying host may decide to send a remote query with a lower depth value to check if there is a change in the provenance metadata before it sends a remote query with the maximum depth necessary. If there is no change in the provenance metadata in nearby hosts, the querying host may use its own cache to answer the lineage or path query.

6.5 Correctness Proofs

The correctness of the discrepancy detection algorithm is proved using induction over the size of an isolated discrepancy. An isolated discrepancy is defined as a maximal connected subgraph of vertices and edges contained in the previous response G_{cache} but missing in $G_{response}$. In general, there may be multiple isolated discrepancies in $G_{response}$.

Theorem 1: *Algorithm 1 detects an isolated discrepancy of any size.*

Proof: *Proof by induction on the size of discrepancy k.*

- *Base Step: $k = 1$. If the discrepancy is a single vertex from G_{cache} missing from $G_{response}$, then Lines 2–6 would detect the discrepancy. If the discrepancy is a single edge from G_{cache} missing from $G_{response}$, then Lines 7–11 would detect the discrepancy. Thus, any discrepancy of size $k = 1$ is detected by Algorithm 1.*
- *Inductive Step: Assume that Algorithm 1 detects an isolated discrepancy of size up to k. An isolated discrepancy of size $k + 1$ is the union of an isolated discrepancy of size k and an additional vertex/edge connected from the discrepancy of size k being deleted from $G_{response}$.*

  There are three possible cases for the additionally-deleted vertex or edge – vertex, incoming edge and outgoing edge:
  - *Vertex: This is the case where an additional vertex connected to an edge in the discrepancy of size k is deleted. If the vertex has an incident edge in $G_{response}$, then Lines 12–16 of the algorithm would detect that the vertex is missing. If the vertex does not have an incident edge in $G_{response}$,*

by the definition of an isolated discrepancy, all its incident edges are in the discrepancy of size k. If an incident edge of the vertex in G_{cache} is not in the discrepancy of size k and not in $G_{response}$, then the size of the isolated discrepancy would be of size $k+2$ (= k + missing vertex + missing incident edge), not $k+1$. While the newly deleted vertex does not increase dc, the discrepancy of size k is detected due to the inductive hypothesis that the algorithm detects any isolated discrepancy of size k. Thus, the algorithm detects the discrepancy of size $k+1$.

- *Incoming Edge: This is the case where an additional incoming edge to a vertex in the discrepancy of size k is deleted. The other vertex x associated with the edge must be in $G_{response}$ and in G_{cache}, so Lines 2–6 of the algorithm would detect the discrepancy and increase dc.*
- *Outgoing Edge: This is the case where an additional outgoing edge from a vertex in the discrepancy of size k is deleted. The other vertex y associated with this edge must be in $G_{response}$, and is detected as a discrepancy in Lines 17–21 unless there is another edge that goes to vertex y. If there is another edge to y, then the algorithm would still detect the discrepancy based on the discrepancy of size k, but would not return a higher discrepancy count dc.* □

Theorem 2: *Algorithm 1 detects any number of isolated discrepancies of any size.*

Proof: *Each isolated discrepancy is connected to a legitimate vertex or edge in the dependency graph. If it is a vertex, then the vertex would miss a path from/to other parts of the graph and the algorithm would detect it. If it is an edge, then the edge would miss a vertex and become a dangling edge. Lines 12–16 in the algorithm specifically detect this discrepancy.* □

6.6 Probabilistic Analysis

Algorithm 1 detects any discrepancy that occurs in $G_{response} \cap G_{cache}$ and rejects $G_{response}$. Thus, for any $G_{response}$ with a discrepancy to bypass the detection algorithm, all the discrepancies such as missing vertices and edges should occur in $G_{response} \setminus G_{cache}$. Assume that the size of $G_{response}$ is s (equal to the number of vertices and edges in $G_{response}$), and the probability that any vertex or edge is removed from $G_{response}$ is p_{Ω}. Then, the expected number of missing vertices and edges from $G_{response}$ is $p_{\Omega} \times s$. When the probability that any vertex or edge in $G_{response}$ is already in G_{cache} is equal to p_c, the probability p_f of all the missing vertices and edges occurring in $G_{response} \setminus G_{cache}$ is given by:

$$p_f = \frac{\binom{(1-p_c)*s}{p_{\Omega}*s}}{\binom{s}{p_{\Omega}*s}}$$

The probability p_f is the upper bound of the algorithm not detecting any discrepancy and accepting $G_{response}$ with missing vertices and/or edges. The

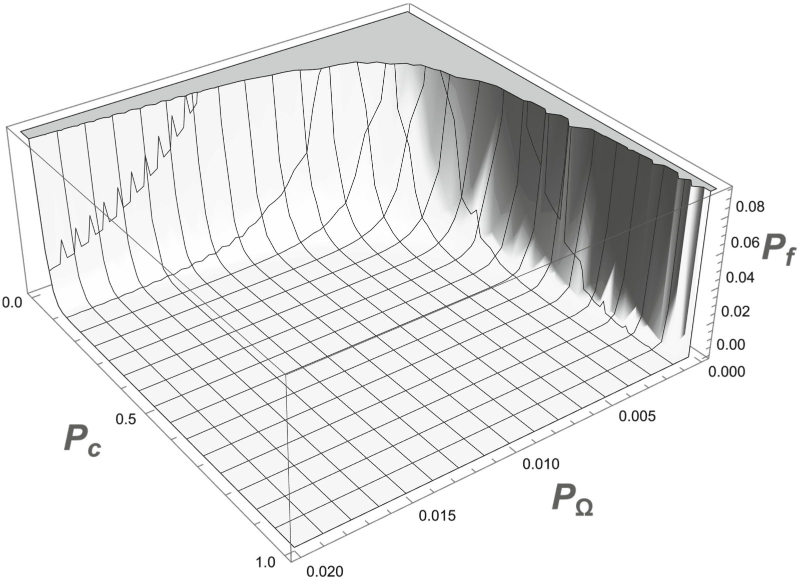


Fig. 9. Probability of discrepancy detection failure.

algorithm would detect that $G_{response}$ is missing vertices or edges if there are any dangling vertices and edges, and the probability of all the missing vertices and edges being arranged such that there are no dangling vertices and edges is strictly less than one.

Figure 9 shows how the probability p_f changes when p_Ω ranges from 0 to 0.2 and p_c ranges from 0 to 1. When the system launches, there is little overlap between G_{cache} and $G_{response}$, and p_c is close to zero. As G_{cache} builds up, the overlap increases and p_f decreases as well.

Figure 9 also shows that p_f quickly decreases as p_c increases. Also, as p_Ω increases, it is less likely that all the missing vertices and edges would be in $G_{response} \setminus G_{cache}$; thus, p_f decreases.

Figure 9 also shows that probability p_f quickly decreases as p_Ω increases. For example, when p_Ω is 0.005 and p_c is 0.5, p_f is 0. In other words, when there is 50% overlap between G_{cache} and $G_{response}$, Algorithm 1 would detect that $G_{response}$ is missing 0.5% or more vertices and/or edges. Once the overlap increases to 90%, the algorithm would detect $G_{response}$ is missing 0.1% or more vertices and/or edges.

7 Related Work

Several systems offer metadata or provenance management in distributed environments. FusionFS [25] implements distributed file metadata management based on distributed hash tables. ExSPAN [28] is a generic framework for provenance management that employs the distributed query processing capabilities of declarative networks. It extends a traditional relational database management system for provenance collection and retrieval.

Several systems have been used to track the provenance of scientific applications. The open-source workflow management system Taverna [24] enables biologists to add application-level annotations of data provenance. CMCS [16] applies an informatics-based approach for synthesizing multi-scale chemistry information. ESSW [7] is a metadata storage system for earth scientists.

None of the systems mentioned above address the problem of discrepancy detection in distributed environments. In many cases, they are customized to specific application domains. In contrast, SPADE adopts a domain-agnostic approach. This enables the enhancements described in this chapter to be utilized in a wide range of settings.

Providing security for data provenance in distributed environments has also been discussed in the literature. Wang et al. [22] proposed a public-key linked chain provenance framework to protect provenance metadata. The Mendel protocol incorporates a three-pronged strategy that combines signature verification and cryptographic ordering witnesses to perform provenance verification in distributed environments [8]. In decentralized settings, where each host signs its own responses, such cryptographic protections cannot address the concerns raised in this chapter.

Some systems focus on specific security aspects that relate to their target domains. Cheney [4] outlined a formal model of security properties for provenance. The Trio system enables the source of uncertainty to be traced after tracking the provenance of database elements [23]. TAP [26] and DTaP [27] are time-aware provenance models that explicitly represent time, distributed state and state change in order to secure queries in the absence of trusted nodes in a network. Liao and Squicciarini [14] developed a system that identifies anomalies in the MapReduce framework based on provenance information collected from within the framework.

Other systems have used provenance metadata in critical infrastructure. Sultana et al. [18] demonstrated that provenance can be used for data integrity in large-scale sensor networks, where the collected data supports decision making in critical infrastructure assets. When a base station knows the communication paths in the network, the complete path of any data sent from a source sensor to the base station can be encoded in a Bloom filter. This enables the base station to compare the provenance to the known path. Each datum from the source comes with a sequence number. The base station can tell if a packet is missing from the skipped sequence number and identify malicious node(s) using the path information of the next packet.

Provenance has also been used in intrusion detection. Hassan et al. [21] employed a provenance graph in cluster auditing to process system audit information in an efficient manner. The provenance graph generated from system audit information is used to monitor hosts in a cluster during normal operation and also to reconstruct attacks in forensic investigations. Berrada et al. [2] evaluated five categories of unsupervised anomaly detection algorithms on provenance data collected via DARPA's Transparent Computing Program, which includes advanced persistent threats.

However, none of the above approaches detect the types of discrepancies addressed by Algorithm 1 in this chapter. The approach is prototyped in cyber infrastructure that is available for researchers to modify and deploy in their own environments. Additionally, code for the core functionality, such as caching and discrepancy detection, is available at the SPADE open-source repository. In contrast, the implementations of many other systems for securing provenance have not been released to the research community.

8 Conclusions

This chapter has introduced the notion of whole-network provenance that represents dependency metadata within and across hosts in distributed systems. First, it shows how the slice of whole-network provenance related to a local artifact or process is reconstructed by issuing specific distributed queries. Next, it demonstrates how each host can build a cache of provenance records received in response to queries made to remote hosts. Finally, it describes an approach that detects discrepancies in provenance metadata distributed across several hosts by comparing previously-cached responses against new responses. The fact that provenance grows monotonically is leveraged to detect a discrepancy in the event that a later response is missing an element in an earlier response.

The DISTDET provenance-based attack detection system has been installed on more than 22,000 hosts at over 50 industrial customers [6]. Future research will focus on deploying the proposed system in real network environments.

Note that a preliminary version of this work appeared in [1]. This chapter extends the previous work with empirical analysis, graph storage analysis, algorithm formalization and probabilistic analysis. Note also that the views and conclusions in this chapter are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, expressed or implied, of the National Science Foundation or the U.S. Government.

Acknowledgment. This research was supported by the National Science Foundation under Grant no. ACI 1547467.

References

1. Ahmad, R., Jung, E., de Senne Garcia, C., Irshad, H., Gehani, A.: Discrepancy detection in whole-network provenance. In: Proceedings of the Twelfth USENIX Conference on Theory and Practice of Provenance, article no. 5 (2020)
2. Berrada, G., et al.: A baseline for unsupervised advanced persistent threat detection in system-level provenance. Futur. Gener. Comput. Syst. **108**, 401–413 (2020)
3. Catlett, C.: The philosophy of TeraGrid: building an open, extensible, distributed terascale facility. In: Proceedings of the Second IEEE/ACM International Symposium on Cluster Computing and the Grid (2002)

4. Cheney, J.: A formal framework for provenance security. In: Proceedings of the Twenty-Fourth IEEE Computer Security Foundations Symposium, pp. 281–293 (2011)
5. Defense Advanced Reseach Projects Agency, Transparent Computing (archived), Arlington, Virginia (darpa.mil/program/transparent-computing) (2023)
6. Dong, F., et al.: DISTDET: a cost-effective distributed cyber threat detection system. In: Proceedings of the Thirty-Second USENIX Security Symposium, pp. 6575–6592 (2023)
7. Frew, J., Bose, R.: Earth system science workbench: a data management infrastructure for earth science products. In: Proceedings of the Thirteenth International Conference on Scientific and Statistical Database Management, pp. 180–189 (2001)
8. Gehani, A., Kim, M.: Mendel: efficiently verifying the lineage of data modified in multiple trust domains. In: Proceedings of the Nineteenth ACM International Symposium on High Performance Distributed Computing, pp. 227–239 (2010)
9. Gehani, A., Kim, M., Malik, T.: Efficient querying of distributed provenance stores. In: Proceedings of the Nineteenth ACM International Symposium on High Performance Distributed Computing, pp. 613–621 (2010)
10. Gehani, A., Kim, M., Zhang, J.: Steps toward managing lineage metadata in grid clusters. In: Proceedings of the First Workshop on the Theory and Practice of Provenance, article no. 7 (2009)
11. Gehani, A., Lindqvist, U.: Bonsai: balanced lineage authentication. In: Proceedings of the Twenty-Third Annual Computer Security Applications Conference, pp. 363–373 (2007)
12. Gehani, A., Tariq, D.: SPADE: support for provenance auditing in distributed environments. In: Proceedings of the ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, pp. 101–120 (2012)
13. Irshad, H., et al.: TRACE: enterprise-wide provenance tracking for real-time APT detection. IEEE Trans. Inf. Forensics Secur. **16**, 4363–4376 (2021)
14. Liao, C., Squicciarini, A.: Towards provenance-based anomaly detection in MapReduce. In: Proceedings of the Fifteenth IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 647–656 (2015)
15. McKay, B.D.: Computing automorphisms and canonical labellings of graphs. In: Holton, D.A., Seberry, J. (eds.) Combinatorial Mathematics. LNM, vol. 686, pp. 223–232. Springer, Heidelberg (1978). https://doi.org/10.1007/BFb0062536
16. Pancerella, C., et al.: Metadata in the collaboratory for multi-scale chemical sciences. In: Proceedings of the International Conference on Dublin Core and Metadata Applications, pp. 121–129 (2003)
17. Pohly, D., McLaughlin, S., McDaniel, P., Butler, K.: Hi-Fi: collecting high-fidelity whole-system provenance. In: Proceedings of the Twenty-Eighth Annual Computer Security Applications Conference, pp. 259–268 (2012)
18. Sultana, S., Ghinita, G., Bertino, E., Shehab, M.: A lightweight secure scheme for detecting provenance forgery and packet drop attacks in wireless sensor networks. IEEE Trans. Dependable Secure Comput. **12**(3), 256–269 (2015)
19. Tan, Y., Ko, R., Holmes, G.: Security and data accountability in distributed systems: a provenance survey. In: Proceedings of the Tenth IEEE International Conference on Embedded and Ubiquitous Computing, pp. 1571–1578 (2013)
20. Towns, J., et al.: XSEDE: accelerating scientific discovery. Comput. Sci. Eng. **16**(5), 62–74 (2014)

21. Hassan, W.U., Aguse, L., Aguse, N., Bates, A., Moyer, T.: Towards scalable cluster auditing through grammatical inference over provenance graphs. In: Proceedings of the Twenty-Fifth Network and Distributed Systems Security Symposium (2018)
22. Wang, X., Zeng, K., Govindan, K., Mohapatra, P.: Chaining for securing data provenance in distributed information networks. In: Proceedings of the IEEE Military Communications Conference (2012)
23. Widom, J.: Trio: a system for integrated management of data, accuracy and lineage. In: Proceedings of the Second Biennial Conference on Innovative Data Systems Research, pp. 262–276 (2005)
24. Wolstencroft, K., et al.: The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. Nucleic Acids Res. **41**(WS), W557–W561 (2013)
25. Zhao, D., et al.: FusionFS: toward supporting data-intensive scientific applications on extreme-scale high-performance computing systems. In: Proceedings of the Second IEEE International Conference on Big Data, pp. 61–70 (2014)
26. Zhou, W., Ding, L., Haeberlen, A., Ives, Z., Loo, B.: TAP: time-aware provenance for distributed systems. In: Proceedings of the Third USENIX Workshop on the Theory and Practice of Provenance (2011)
27. Zhou, W., et al.: Distributed time-aware provenance. Proc. VLDB Endow. **6**(2), 49–60 (2012)
28. Zhou, W., Sherr, M., Tao, T., Li, X., Loo, B., Mao, Y.: Efficient querying and maintenance of network provenance at internet-scale. In: Proceedings of the Twenty-Ninth ACM SIGMOD International Conference on Management of Data, pp. 615–626 (2010)

A Contextual Integrity Property to Impede Privacy Violations in 5G Networks

James Wright(✉) and Stephen Wolthusen

Norwegian University of Science and Technology, Gjøvik, Norway
james.g.wright@ntnu.no

Abstract. The privacy of information transmitted between user equipment and radio nodes in 5G networks is preserved using encrypted channels. However, this single point of failure would expose the identities and, potentially, locations of network users if a vulnerability were to be discovered and exploited.

This chapter presents a consensus algorithm that adds an additional layer of defense in the 5G standard. The algorithm leverages access to the 5G control network by multiple radio nodes in an administrative area to control the mobility of agents that can connect with user equipment. The algorithm is designed to decrease the likelihood of privacy violations by an international mobile subscriber identity catcher should a vulnerability be found in the 5G-AKA protocol. The algorithm is formalized using the π-calculus to create a contextual integrity property, and is verified using π-calculus equivalence relations.

Keywords: 5G Security · Defense in Depth · IMSI Catcher · π-Calculus

1 Introduction

A consistent threat to the various generations of cellular communications technologies is the deployment of malicious radio nodes called international mobile subscriber identity (IMSI) catchers to undermine the identities and locational privacy of user equipment. As the technology has developed over several generations, the cellular communications trust model has matured by moving as much of the information passed between user equipment and legitimate radio nodes to after they have mutually authenticated their identities [9]. However, an adversary always has an advantage because the standards require user equipment to connect to the strongest signal in a network. Specifically, the adversary can jam legitimate signals to force user equipment in the vicinity to connect to it.

In fact, adversaries have several ways of circumventing 5G network security controls because implementations are required to serve previous generations of cellular technologies that have weaker security controls, and security vulnerabilities have been discovered during the development of the 5G standard. Relying on

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 115–139, 2024.
https://doi.org/10.1007/978-3-031-49585-4_6

encryption to protect transmitted information creates a security bottleneck – if encryption fails, the 5G security guarantees are undermined. Building additional security protections in the 5G standard to prevent or respond to the deployment of IMSI catchers in an administrative area would increase the privacy protection offered to user equipment.

Research has been undertaken to detect IMSI catcher deployments. Several signatures have been proposed to identify their malicious activity and the 5G standard mandates the use of information from user equipment to detect if an IMSI catcher is operating in an administrative area. However, these are weak responses to attacks because IMSI catchers are detected only after they have successfully connected to user equipment. Responding to adversaries before they connect to user equipment decreases the likelihood that the privacy guarantees of the 5G standard would be violated.

This chapter presents a distributed consensus algorithm that builds an additional layer of defense in the 5G standard that would restrict the deployment of IMSI catchers in an administrative area. The solution requires that specific network semantics be maintained to create a contextual integrity property. Although this property is weaker than cryptographic integrity, it enables intrusion detection to be focused by reducing the semantic space that must be surveyed. The property depends on the fact that a 5G control network has access to a greater number of legitimate radio nodes in a particular administrative area than an adversary, and will be able to form a consensus on whether or not a new radio node is malicious. This enables a 5G system operator to control the mobility of radio nodes in an administrative area.

The consensus processes leverage the 5G broadcast network semantics so that radio nodes can interrogate a new radio node in an administrative area. The goal is to warn user equipment in the administrative area not to connect to an IMSI catcher in advance of it exploiting vulnerabilities. The algorithm can detect adversaries that present themselves as new radio nodes or masquerade as legitimate radio nodes, as well as adversaries that have conducted passive reconnaissance to detect legitimate radio nodes. The algorithm uses existing infrastructure and communications models in the 5G standard to perform its interrogation.

The algorithm is presented as a π-calculus process algebra that expresses how mobile multiagent distributed systems concurrently pass messages and communications channels between themselves. The π-calculus also models modifications to radio node and control network agents to enable them to interrogate and form a consensus on adversary presence, as well as the ability of user equipment to form a consensus on whether or not to act on alerts. The algorithm is verified using π-calculus equivalence relations to demonstrate that the added consensus messages are distinguishable from those sent by an adversary simulating a radio node and that deadlocks are not created during the user equipment registration procedure.

The algorithm reduces the likelihood of privacy violations [13] should the 5G encryption model be compromised because it enables the network to automatically instruct user equipment not to establish connections with suspicious radio nodes. Incorporating the automatic response in the 5G standard would enable IMSI catchers to be ejected from a network before they connect with legitimate user equipment, independent of the organizational policies of 5G implementations. Implementing the consensus algorithm also benefits 5G security controls because they can leverage the anomalies detected by sensor-based and network-based IMSI catcher detection solutions directly in the standard. The consensus algorithm also increases the amount and the uncertainty of reconnaissance conducted by adversaries because they are more likely to be detected as they attempt to identify legitimate radio nodes.

2 Related Work

This section reviews the different kinds of IMSI catchers and their countermeasures. The goal is to extract adversary capabilities that can be used to formulate the adversary model in this research. The primary security concern of the cellular communications infrastructure is to prevent breaches of identity and locational privacy of user equipment.

Park et al. [14] analyzed the capabilities of IMSI catchers that were being sold to governments in 2019. Some of the devices they studied passively listened on unencrypted public channels to discern the temporary identifiers of user equipment during the registration procedure. Active IMSI catchers act as radio nodes. By exploiting various protocol vulnerabilities, they are able to intercept permanent identifiers and track user equipment, and potentially eavesdrop on communications if they can compromise the control network. An IMSI catcher can force a connection in various ways. Some devices jam the frequency bands of legitimate radio nodes in an attempt to force connections with user equipment in an administrative area. Other devices can automatically forge the identities of nearby radio nodes before attempting to intercept user equipment. Another way to undermine privacy is to force a downgrade to a more insecure communications standard during the negotiation of the ciphertext suite. The option to downgrade is possible due to the backward compatibility requirements that include LTE and earlier standards. Shaik et al. [16] leveraged a 4G protocol vulnerability that enabled the modification of the stated capabilities of user equipment to cap the data transmission rate and drain battery power.

Several vulnerabilities have been discovered in the 5G-AKA key protocol during its development. Cremers and Dehnel-Wild [3] employed a Tamarin symbolic protocol tester to assess a protocol draft, discovering a replay attack that enables an attacker to receive user equipment keys by swapping uplink control information (UCI). Borgaonkar et al. [2] demonstrated that an adversary could steal the IMSI or subscription permanent identifier of user equipment in a replay attack because the user equipment portion of the encryption protocol does not employ randomness [2]. Basin et al. [1] also used Tamarin to demonstrate that the 5G standard had various under-defined notions of authentication that enable an adversary to replay keys and violate forward secrecy.

Several methods have been developed for assessing the risks posed by IMSI catchers. Yocam et al. [19] developed a risk assessment methodology for analyzing the trade-offs between IMSI catcher security and risk, enabling 5G implementers to select cost-effective security controls. They assessed the effects of incorrect security implementations as well as the ability of security controls to combat legacy downgrade, denial-of-service and message interception attacks. Khan et al. [10] proposed a pseudonym scheme for 4G networks, like the 5G subscription permanent identifier, to prevent identity leaks during user equipment registration. This would also increase the security of the 5G standard, because if a backward compatibility attack is discovered, the 4G standard would have additional cleartext security.

Sensor-based methods identify anomalies in publicly-broadcasted channels. The methods cannot survey dedicated channels between user equipment and radio nodes [14]; instead, they look for misconfigured or new radio nodes. Examples of sensor solutions are sICC [5] and SITCH [20].

Network-based IMSI catcher detection schemes attempt to identify adversaries by the artifacts they leave in communications networks. However, they require the cooperation and information sharing from network operators, which is not always guaranteed [14]. Steig et al. [17] proposed a detection scheme for identifying active and passive IMSI catchers in 4G networks. The scheme measures the locations of known radio nodes in an area; since legitimate radio nodes rarely change their locations, alarms are raised if their positions have changed. Dabrowski et al. [4] developed a rubric for identifying signs of IMSI catchers. Their scheme detects anomalies in user equipment registration when an attacked device reconnects with a legitimate network. They also propose the use of radio nodes and sensors to listen to registration procedures in order to detect malicious cipher downgrades.

A number of IMSI catchers have been developed or theorized in the literature. Passive IMSI catchers are sensors that gather publicly-broadcasted information before targeted user equipment has been authenticated whereas active IMSI catchers exploit vulnerabilities to trick user equipment into believing they are legitimate radio nodes. Regardless of the type of IMSI catcher, the primary objective of the adversary is to extract the unique identities of user equipment to undermine their privacy. Secondary objectives include tracking user equipment and eavesdropping on communications. Various means for forcing connections with user equipment are possible, including jamming the signals of legitimate radio nodes. While the 5G standalone mode resolves the historical attack vectors, vulnerabilities exist that can be exploited by a dedicated adversary [9].

Inherent to all the countermeasures is the requirement that 5G network operators have procedures to eject IMSI catchers from their networks after they are detected. However, none of the methods automatically remove adversaries from networks; they only inform the operators of the presence of IMSI catchers.

3 5G Defense in Depth

This section describes the 5G registration procedure and specifies an adversary model that is employed as the foundation of the contextual integrity property. The section also describes the boundaries that must be maintained for the consensus algorithm to successfully mitigate IMSI catchers and provide tighter mobility control in 5G networks. The consensus algorithm, which is constructed from pre-existing communications models used during the user equipment registration procedure, is described along with certain modifications.

3.1 5G Device Registration Procedure

Figure 1 shows the 5G registration procedure [6]. The procedure, which involves three participants across multiple communications models, has more than 100 steps. This research models Step 2 in the procedure up to the establishment of an encrypted communications channel between user equipment (UE) and a radio node (gNB). All the communications between the agents from the new access and mobility management function (AMF) onwards are treated as a single agent because they constitute various parts of the control network (CN). Steps 12 to 20 are beyond the scope of the model because neither the user equipment nor the radio node can see their transitions in the network.

The first stage is to establish radio communications between the user equipment and radio node. The radio resource control (RRC) procedure establishes the connection between the two devices. The user equipment messages the radio node and begins a timer to complete the handshake. The radio node then sends the frequency of a direct communications channel and requests a temporary identifier from the user equipment. After switching to the new channel, the radio node sets up the signal radio bearer (SRB_1) and sends it to the user equipment in a RRCSetup message. This message contains the MAC address of the radio node. Following this, the radio node informs the new access and mobility management function that the user equipment wishes to connect to its administrative area. This function negotiates the transfer of the security context of user equipment from the access and mobility management function of the old radio node to the access and mobility management function of the new radio node. This function is omitted from the model because it does not contribute to channel encryption.

The new access and mobility management function begins a series of security checks via the radio node to authenticate the user equipment. First, it requests a subscription permanent identifier (SUPI) from the user equipment, which is a concatenation of the mobile country code (MCC), mobile network code (MNC) and IMSI number of the user equipment. Next, the 5G-AKA protocol [8] is executed to enable the user equipment and radio node to authenticate each other's keys. The two authentication checks are condensed to one step in the formal model presented in the next section.

At this point, the new access and mobility management function performs a series of checks to ensure that the user equipment is permitted to operate in the administrative area. These are beyond the scope of this work. After the

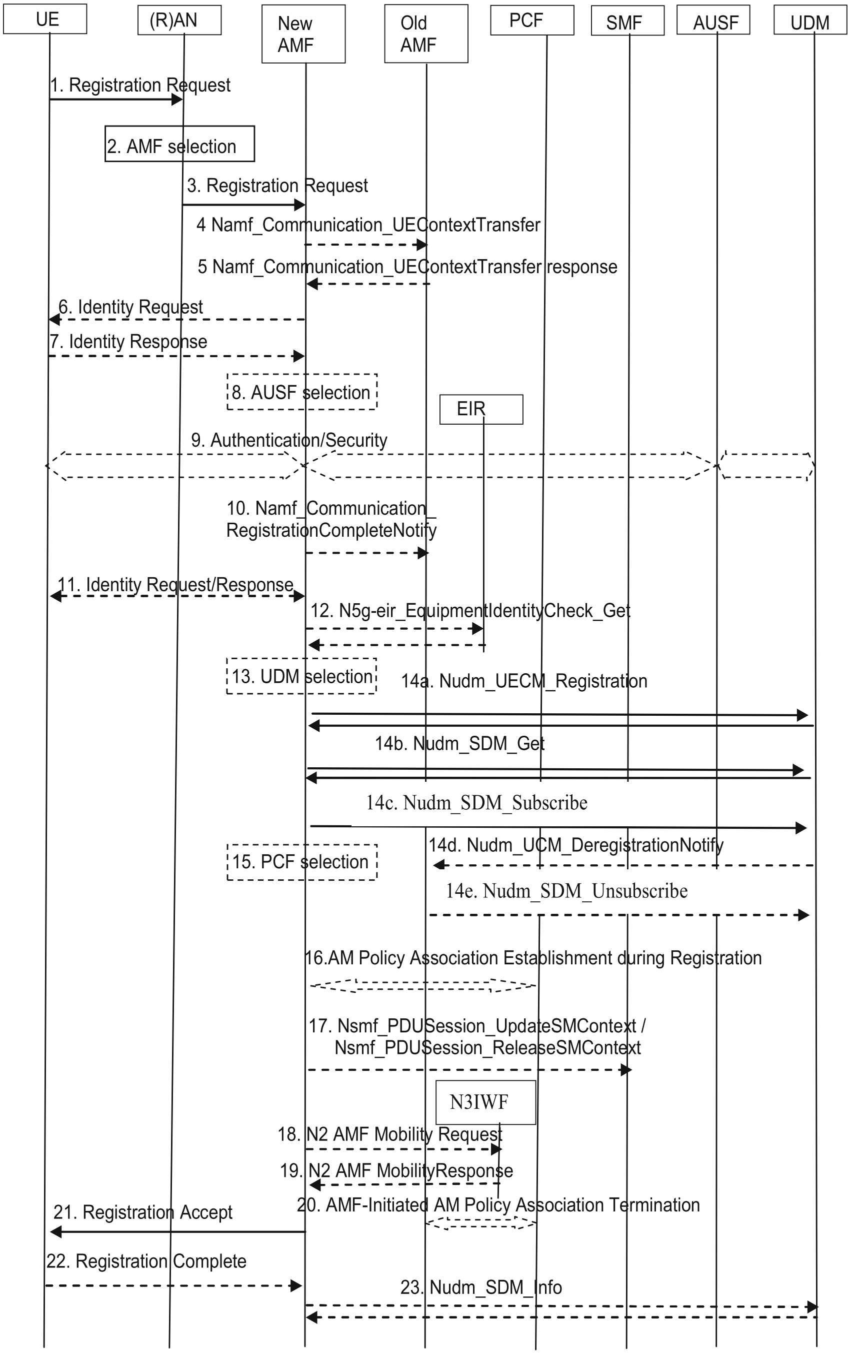

UE
(R)AN
New AMF
Old AMF
PCF
SMF
AUSF
UDM
1. Registration Request
2. AMF selection
3. Registration Request
4 Namf_Communication_UEContextTransfer
5 Namf_Communication_UEContextTransfer response
6. Identity Request
7. Identity Response
8. AUSF selection
EIR
9. Authentication/Security
10. Namf_Communication_RegistrationCompleteNotify
11. Identity Request/Response
12. N5g-eir_EquipmentIdentityCheck_Get
13. UDM selection
14a. Nudm_UECM_Registration
14b. Nudm_SDM_Get
14c. Nudm_SDM_Subscribe
14d. Nudm_UCM_DeregistrationNotify
15. PCF selection
14e. Nudm_SDM_Unsubscribe
16.AM Policy Association Establishment during Registration
17. Nsmf_PDUSession_UpdateSMContext / Nsmf_PDUSession_ReleaseSMContext
N3IWF
18. N2 AMF Mobility Request
19. N2 AMF MobilityResponse
20. AMF-Initiated AM Policy Association Termination
21. Registration Accept
22. Registration Complete
23. Nudm_SDM_Info

Fig. 1. Session diagram of the 5G registration procedure [6].

access and mobility management function confirms that the user equipment is allowed to operate, it informs the radio node to proceed with the negotiation of the security mode (SM) function to create the encrypted channel. The radio node sends a session key and requests a ciphertext suite. The user equipment then responds and the encrypted channel is established.

3.2 Active Adversary Model

IMSI catchers are passive or active. Since passive devices do not participate in a network, this work focuses on an adversary with an active IMSI catcher. The goal of the algorithm is to prevent the adversary from completing a registration procedure with user equipment in an administrative area. Therefore, the modeled adversary does not have the ability to jam legitimate radio node signals. Since the proposed defense-in-depth mechanism is intended to provide additional security to 5G systems via the contextual integrity property, it is explicitly assumed that the adversary has cryptanalytic functions that have broken the 5G-AKA protocol. In order to intercept the identity of user equipment, it is assumed that the adversary can simulate messages sent by the radio node and control network during the registration procedure. The security model assumes that the adversary has not compromised any of the other radio nodes in the network and, therefore, cannot manipulate the internal state of other agents except via the messages they transmit. The model also assumes that the adversary has conducted passive network reconnaissance to identify legitimate radio nodes.

3.3 Contextual Integrity Property

The consensus algorithm described in the next section will not prevent all attempts to undermine the privacy promises of the 5G standard. Its purpose is to delay privacy violations by adding an additional defense-in-depth layer should the 5G-AKA protocol be broken. It also makes the reconnaissance phase of an attack more limited and precarious by forcing the adversary to only listen to traffic in a passive manner. The algorithm provides a mechanism for the control network and legitimate radio nodes to control radio node mobility in an administrative area. All this is achieved, through the monitoring and control of network semantics that remain immutable via intrusion detection systems, in order to construct the contextual integrity property.

Since the devices operate in a wireless network, they can only see messages sent and received by other devices in the coverage area provided by the radio node. There is no guarantee that user equipment will observe the state transition of any device with which it communicates. But this does mean that it can observe messages of agents with which it is not in direct communication. Additional model simplifications are that only the sending and receiving of messages cause state transitions in a device and all messages are assumed to be sent on a reliable channel.

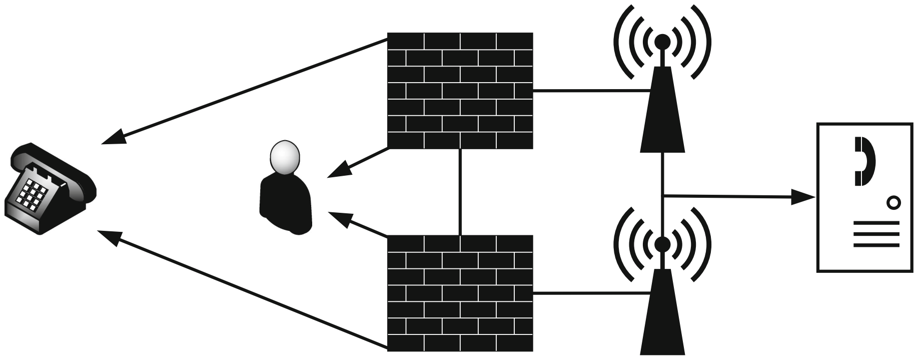


Fig. 2. Network topology facilitating the consensus algorithm.

As stated in Sect. 3.2, it is assumed that the adversary has broken the 5G-AKA protocol, but does not have access to the radio node/control network communications public key infrastructure (PKI). This will be proved in Sect. 6. The representation of the adversary's cryptanalytic capabilities is expressed in Sect. 4 as $K_m = K_{AC1}$. Since the adversary has compromised the 5G-AKA protocol, the consensus algorithm only works if the number of legitimate radio nodes N_{gNB} is greater than the number of malicious/imitated radio nodes set up by the adversary N_A, i.e., $N_{gNB} > N_A$. If the adversary employs more radio nodes than the legitimate network, it would be able to use the consensus algorithm to establish connections with user equipment. However, this requires more resources and coordination than merely deploying a jammer.

An implicit race condition exists in the consensus algorithm. This is borne from the real-time requirements of some 5G communications models involved in the user equipment registration procedure [7]. Computing the overhead and timing of a successful consensus is beyond the scope of the proposed model.

3.4 Defense-in-Depth Consensus Algorithm

The consensus algorithm depends on multiple radio nodes to control radio node mobility in an administrative area via an interrogation process. The interrogation process begins when a new radio node cannot be reached via the public key infrastructure of the radio node. Figure 2 shows the relationships between the agents in the interrogation process. Upon detecting an unreachable radio node, the legitimate radio nodes interrogate it by pretending to be user equipment. They begin their own radio resource control procedures to gain the adversary's MAC address from its SRB_1 messages. If the unreachable radio node is an IMSI catcher, two things can occur. If all the legitimate radio nodes do not recognize the adversary's MAC address, they form a consensus between themselves and page warnings to user equipment in their administrative area. However, if only one legitimate radio node recognizes the adversary, then the radio nodes that do not recognize it query the control network to resolve the deadlock. If the control network does not recognize the adversary, it informs the querying radio node, which proceeds to page its warning and instruct other legitimate radio nodes to do this as well.

Another possibility described in the adversary model is to impersonate the MAC address of a legitimate radio node. If the radio node recognizes this during an interrogation, it immediately pages its warning and messages the control network to coordinate with another radio node in the area to do this as well.

A final possibility is when the adversary has deduced which user equipment are actually radio nodes. In this case, when the adversary refuses to connect, the radio node waits for the adversary to use its 5G-AKA key during an observed radio resource control procedure and intercept the key. The radio node then pages its warning containing this key instead of the MAC address and tells the control network to coordinate another warning page.

User equipment has its own consensus algorithm to decide how to respond to paged warnings. As long as the conditions in the previous section are met, user equipment receives two paged warnings from different radio nodes containing either the adversary's MAC address or 5G-AKA key. After the consensus is reached, the user equipment disconnects and restarts its registration procedure. The user equipment portion of the radio resource control and non-access stratum (NAS) procedures are modified to check that they do not reconnect to the adversary.

4 π-Calculus Model of 5G Registration

This section lays the foundation for the defense-in-depth consensus algorithm. The qualitative description of the 5G registration procedure in Sect. 3.1 is formally specified as a π-calculus. The adversary model and contextual integrity property are also formalized.

Note that only the relevant π-calculus results are presented. These include the π-calculus operations used to describe the communications models of processes that represent agents in the 5G registration procedure and defense-in-depth mechanism. Interested readers are referred to Milner et al. [11,12] and Sangiorgi and Walker [15] for details about the π-calculus.

4.1 π-Calculus Semantics

The π-calculus expresses the evolution of concurrent processes in a distributed system as they pass names among themselves. The names represent variables or communications channels. The π-calculus operations used to model the consensus algorithm are:

$$P ::= \bar{x}y.P \mid x(z).P \mid \nu z\ z.P \mid \tau.P \mid P|P' \mid P + P' \mid [x = y].xP \mid \bar{0}$$

where P represents a subsequent process or the underlying state machine of a device.

The operations are described as follows:

- $\bar{\mathbf{x}}\mathbf{y}.\mathbf{P}$: Process P sends name y on channel x.
- $\mathbf{x}(\mathbf{z}).\mathbf{P}$: Process P receives name z on channel x. Any subsequent z actions in process P are replaced by input z.
- $\nu\mathbf{z}\ \mathbf{z}.\mathbf{P}$: A restriction of name z where sending or receiving on channel z can only occur if the involved process P already knows name z exists.
- $\tau.\mathbf{P}$: Silent action τ representing an action internal to process P that results in no name being sent or received.
- $\mathbf{P}|\mathbf{P}'$: Composition of processes P and P' that can occur concurrently.
- $\mathbf{P}+\mathbf{P}'$: Summation of processes P and P' representing the selection of one of the two processes.
- $[\mathbf{x}=\mathbf{y}].\mathbf{xP}$: Matching of names x and y representing an if-then statement in the distributed system.
- $\bar{\mathbf{0}}$: Null process $\bar{0}$ takes no further action.

The processes evolve via communications with other intra-agent processes or with the environment (i.e., processes external to the process being analyzed).

4.2 Formalized 5G Registration Procedure

The 5G registration procedure must be completed for user equipment to communicate with a radio node gNB.

The user equipment process UE during the registration procedure is defined as:

$$UE = RRC_u.AMFS_u.NAS_u.SM_u$$

which performs the explicit operations:

$$\begin{aligned} UE = \bar{p}_{int}.p(c).\bar{c}\,TID.c(s).\bar{c}\,fin.\\ c(er).\bar{c}\,eid.\\ c(K).[K = K_{AC1}]\bar{c}\,K_{AC2}.\\ c(k).\bar{c}\,fin.SESSION_u \end{aligned} \tag{1}$$

The radio node process gNB during the registration procedure is defined as:

$$gNB = RRC_g.AMFS_g.AMF_g.SM_g$$

which performs the explicit operations:

$$\begin{aligned} gNB = p(i).\bar{p}\,dc.dc(t).\bar{d}c\,SRB_1.dc(f).\\ s\bar{e}c\,dc.\\ sec(a).\\ \bar{d}c\,K_{sesh}.dc(f).SESSION_g \end{aligned}$$

The control network process CN during the registration procedure is defined as:

$$CN = AMFS_c.NAS_c.AMF_c \tag{2}$$

which performs the explicit operations:

$$\begin{aligned} CN = sec(ci).\bar{ci}\ eidr.ci(i). \\ \bar{ci}K_{AC1}.ci(K).[K = K_{AC2}]. \\ s\bar{e}c\ ack \end{aligned}$$

The radio node process gNB and control network process CN are composed into a singular process to represent their wired communications during the registration procedure. This is represented as:

$$HS = \nu\ sec\ gNB|CN \tag{3}$$

where sec represents the encryption of gNB and CN communications using a public key infrastructure.

4.3 Active Adversary Model

The adversary model must simulate the outbound messages of the radio node's RRC_g and SM_g processes, along with the control network's $AMFS_c$ and NAS_c processes. The communications models are represented as the process:

$$A_o = RRC_g.AMFS_c.NAS_c.SM_g$$

which performs the explicit operations:

$$\begin{aligned} A_o = p(i).\bar{p}\ dc.dc(t).\bar{dc}\ SRB_M.dc(f). \\ \bar{dc}\ eidr.dc(i).\bar{dc}\ K_M.dc(K).\bar{dc}\ K_{sesh}.dc(f).INTERCEPT \end{aligned}$$

The cryptanalytic capability of the adversary is expressed by making the names $K_m = K_{AC1}$ when parsed by the user equipment process UE.

4.4 Formalized Contextual Integrity Property

The following theorem formally states the property that enables the defense-in-depth process to control the mobility of radio nodes in an administrative area.

Theorem 1: *If the assumptions in Sect. 3.4 hold, then the contextual integrity property is maintained if and only if the adversary never learns the name sec and cannot distinguish between a user equipment registration process and a radio node interrogation process.*

Theorem 1 is proved using the results in Sects. 6.6 and 6.8.

5 Formalized Consensus Algorithm

This section presents the modified 5G processes that create the consensus algorithm before presenting the semantic artifacts that a 5G intrusion detection system can leverage to detect violations of mobility control in an administrative area.

The modifications to the user equipment process UE enables it to form a consensus from the paged warnings broadcast by multiple radio node processes when they have detected an adversary in the administrative area.

The modifications to the radio node process gNB enable it to pretend to be user equipment that attempts to connect with the adversary if it cannot be reached via the public key infrastructure. The radio node process interrogates the adversary via a modified user equipment registration process or it passively intercepts the adversary's credentials that are transmitted publicly.

The modifications to the control network process CN enable it to respond to queries from deadlocked radio nodes and coordinate the immediate warning pages if the adversary is imitating a radio node or refuses to be interrogated.

The formalization of this defense-in-depth mechanism assumes that a process transitions through its state machine by maintaining strictly ordered traces. Also, messages are transmitted on a reliable channel.

The formalization treats the user equipment, adversary and radio node/control network combination as three distinct entities. The communications between radio nodes and the control network are treated as intra-process communications.

5.1 *UE* Process Modifications

The modified form of the UE process is:

$$UE_m = RRC_{um}.AMFS_u.NAS_{um}.SM_u$$

where processes RRC_{um} and NAS_{um} are modified in the event the registration process is re-attempted.

The UE process is modified to receive and act on warning pages from legitimate radio node processes. The consensus process can be inserted between any part of Eq. 1 up to the $\bar{c}\,fin$ operation to maintain the contextual integrity property. This is achieved using the π-calculus abstraction of a context hole [.] placed anywhere in a process before its conclusion. A context hole enables another process to be inserted into a process being studied [15]. The UE consensus process is given by:

$$b_i(s_i, n_i).b_j(s_j, n_j).[s_i = s_j]RRC_{um} \tag{4}$$

where $s_{i,j} = SRB_m \vee K_m$ and n_i is a nonce from the radio node process gNB_i.

The UE process portion of the registration must also be modified to act on any warning that passes the consensus. This requires its RRC_u and NAS_u processes to be modified.

The modified form of the RRC_u process is:

$$RRC_{um} = \bar{p}_{int}.p(c).\bar{c}\,TID.c(s).([s = s_i]RRC_{um} + \bar{c}\,fin)$$

If the UE process ends up reconnecting with the adversary, it begins the process again until it connects with a legitimate radio node.

The modification to the NAS_u process is required if the adversary refuses to participate in the interrogation process. The adversary still has to broadcast its 5G-AKA key as cleartext [6], enabling a radio node process to intercept it. The UE process can, therefore, perform a consensus on the malicious keys. Hence, the reconnection algorithm also needs to begin when the key verification takes place.

The modified form of the NAS_u process is:

$$NAS_{um} = c(K).([K = K_M]RRC_{um} + \bar{c}\, K_{AC2})$$

5.2 *gNB* Process Modifications

This section describes the additional processes incorporated in a radio node process gNB that enable it to interrogate a new radio node that it cannot connect to using the name *sec*. The interrogation processes enable the radio node to control mobility in its administrative area. The modified form of the gNB process is:

$$\begin{aligned} gNB_i = RRC_g.AMFS_g.AMF_g.SM_g| \\ RRC_{ugs}.(([s = \varnothing]R_g + D_g)|I_g|B_g) \end{aligned} \tag{5}$$

The interrogation begins with the radio node process gNB simulating the RRC_u process of user equipment UE. Next, gNB attempts to form a local consensus using the MAC address obtained from the SRB_m message. However, if the adversary refuses the interrogation, gNB enacts the R_g process to intercept the adversary's 5G-AKA key.

The simulated form of process UE is modified with a composite process to pass SRB_m securely to the gNB/CN process from the environment. This is given by:

$$RRC_{ugs} = \bar{p}_{int}.p(c).\bar{c}\, TID.c(s).(\bar{c}\, fin|\bar{bo}\, s)$$

If the adversary refuses to engage in the interrogation, the radio node process gNB listens for the adversary to use the communications channel dc in any of its handshakes with user equipment. This is the channel in the NAS_c process where the control network passes its 5G-AKA key to user equipment. After the key has been intercepted, the gNB process simultaneously pages its warning to all user equipment and messages the control network to coordinate with another radio node to page another warning:

$$R_g = dc(K_s).(s\bar{e}c\, K_s, reject|bo_i\, K_s) + sec(K, r).\bar{bo}_i\, K$$

If the adversary engages with the interrogation process, the radio nodes attempt to resolve the consensus locally. The four possible algorithm terminations of the interrogation are ordered as expressed in Eq. 6 below:

- Both radio nodes recognize the interrogated radio node and leave it to its operations.

- One radio node recognizes the interrogated radio node, but the other radio node does not. In this case, the recognizing radio node waits for the other radio node interrogator to resolve the deadlock by sending a query to the control network. If the control network recognizes it, the deadlocked radio node informs the other radio node to terminate. Otherwise, both radio nodes page warning messages.

Equation 6a expresses the two ways a radio node can recognize an interrogated device whereas Eq. 6b expresses the two ways a radio node can ban an interrogated device:

$$\begin{aligned} D_g = [s = known][(sec(s', c).[s' = s].\bar{sec}\ s', conf|\bar{sec}\ s, conf.sec(s', c))+ \\ (sec(s', a).[s' = s].\bar{sec}\ s', conf|\bar{sec}\ s, conf.sec(s', a)). \quad \text{(6a)} \\ (sec(s, c) + sec(s, a).\bar{bo}_i\ s)] \\ +[(sec(s', a).[s' = s].\bar{sec}\ s', alerta|\bar{sec}\ s, alerta.sec(s', a)).\bar{bo}_i s+ \\ (sec(s', c).[s' = s].\bar{sec}\ s', alerta|\bar{sec}\ s, alerta.sec(s', c)).\bar{sec}\ s, query. \quad \text{(6b)} \\ (sec(s, c).\bar{sec}\ s, conf + sec(s, a).(\bar{sec}\ s, alerta|\bar{bo}_i s))] \end{aligned}$$

- Both radio nodes do not recognize the interrogated radio node and page warnings to user equipment in the area.
- One radio node recognizes the interrogated radio node, but the other radio node does not. In this case, the recognizing radio node waits for the other radio node interrogator to resolve its deadlock by sending a query to the control network. If the control network does not recognize it, the deadlocked radio node pages a warning and instructs the other radio node interrogator to do this as well.

Each of the use cases that a radio node consensus can follow is distinguished by the + summation operation.

The I_g process exists in the event that a radio node realizes that the SRB_m it received during the interrogation is imitating its identity. In this case, the imitated radio node simultaneously pages its warning and informs the control network to immediately coordinate a second paged warning. The I_g process is defined as:

$$I_g = sec(su, i).\bar{bo}_i\ su + [s = me].(\bar{sec}\ s, imitation|\bar{bo}_i\ s)$$

Process B_g expresses a radio node paging all user equipment not to connect to a device with a given SRB_m or K_m:

$$B_g = bo_i(s).\bar{b}_i\ s, n_i$$

As stated in Sect. 3.4, the contextual integrity property is maintained if one of the D_g, I_g or R_g processes is terminated by two radio nodes before the adversary completes its registration process simulation with a targeted user equipment.

5.3 *CN* Process Modifications

The control network participates in the defense-in-depth mechanism as a consensus deadlock breaker and coordinator of paged warnings for urgent violations of mobility control in an administrative area, such as the detection of radio node imitation or an adversary refusing interrogation. The modified form of the CN process defined in Eq. 2 is:

$$CN_m = AMFS_c.NAS_c.AMF_c|I_c|Q_c|R_c$$

The Q_c process resolves the deadlock if one radio node recognizes an interrogated device, but another radio node does not. The radio node that does not recognize the name SRB_m queries the control network. The control network responds to the querying radio node to resolve the deadlock and allow for consensus termination. The Q_c process is specified as:

$$Q_c = sec(s,q).([s = known]s\bar{e}c\ s, confirm + s\bar{e}c\ s, alerta)$$

The processes I_c and R_c ensure that user equipment receives multiple paged warnings:

$$I_c = sec(s,i).s\bar{e}c\ s, imitation$$

$$R_c = sec(k,r).[k = kn\bar{o}wn]s\bar{e}c\ K, reject$$

Specfically, processes I_c and R_c coordinate a second radio node page to user equipment so that the consensus process in Eq. 4 can terminate. In the case of the R_c process, the control network only sends the message if it does not recognize the adversary's key $kn\bar{o}wn$.

5.4 *gNB-CN* Process Coordination Modification

The process that models the intra-process communications between a radio node and control network is modified to include the extra radio node necessary for the consensus algorithm. The DiD process is specified as:

$$DiD = \nu\ sec, bo_i, bo_j\ CN_m|gNB_i|gNB_j \tag{7}$$

5.5 Intrusion Detection Artifacts

Provided that the defense-in-depth consensus can terminate before an adversary can connect to targeted user equipment and the assumptions in Sect. 3.3 hold, the algorithm is able to maintain the contextual integrity property by identifying either the adversary's MAC address in the SRB_1 message or the key K_m it uses in the simulated 5G-AKA protocol. These artifacts can be used by intrusion detection systems to identify unauthorized agents that attempt to broadcast in an administrative area instead of waiting for signs of compromise to be discerned from artifacts due to user equipment reconnecting to the legitimate network after the IMSI catcher has released them.

6 Consensus Algorithm Equivalence Proofs

This section presents several proofs that collectively construct the contextual integrity property. The proofs demonstrate that, as long as the assumptions in Sect. 6.2 hold, the adversary can only intercept traffic or imitate a radio node if it wins a race condition. This occurs regardless of whether or not it has broken the 5G-AKA protocol.

The first section introduces the π-calculus process transitions and behavioral equivalences that are used to prove the contextual integrity property. Also, it presents the assumptions underlying the proofs.

The first proof demonstrates the indistinguishability between the cryptanalytic adversary and 5G registration process. The next proof shows that the adversary process cannot tell the difference between user equipment and a radio node that interrogates user equipment. The subsequent proofs demonstrate the termination of the user equipment consensus process (and registration renegotiation after it has been warned), and that user equipment can distinguish between the defense-in-depth process DiD and adversary process A_o. The final proof shows that, given the assumptions, the adversary will not learn the name sec from the consensus algorithm and, thus, gain access to the gNB-CN intra-process communications.

These proofs collectively demonstrate that the consensus algorithm maintains the contextual integrity property described in Sect. 4.4.

6.1 Weak Transitions and Bisimulation

In the π-calculus, processes are compared using behavioral equivalences that examine process semantics to see if they are indistinguishable to an external observer. The primary equivalence used in the π-calculus is bisimulation, which compares process inputs and its internal transitions [15]. An example of internal process transitions is the messages passed between the radio node process gNB and control network process CN in Eq. 3.

Whether or not two processes are bisimilar can be viewed as a game between two observers [18]. An observer randomly selects one of the two processes being compared and transitions its state via the receipt of an input or an internal state transition. The other observer attempts to make the same transition on the other process with the same input/internal transition. The two processes are bisimilar if the game can run forever or the processes reach a configuration of inputs/internal transitions that has been encountered before. Otherwise, the processes are not bisimilar because the second observer is unable to copy the transition of the first process.

The equivalence proofs in this section use weak bisimulation ($\approx$) that relaxes the rule on comparing internal transitions [15]. In this bisimilarity, the observer cannot discern the number of internal transitions performed by a process. Zero or a positive number of internal transitions could have occurred between a process receiving inputs.

Weak bisimulation is used due to Assumption 2 in the next section. It is not guaranteed that a process can physically observe the transitions of other processes due to geographical separation. Therefore, the security of the system must be derived from the messages that processes send and receive.

6.2 Assumptions

All the proofs in this section are predicated on the assumptions in Sect. 3.3 and the introduction to Sect. 5. The assumptions are:

1. The state machines underlying the processes described in Sect. 5 maintain strictly ordered traces with no errors and their messages are transmitted on reliable channels.
2. A process can only see messages sent and received by other processes, not their internal state transitions.
3. Only the sending or receiving of messages cause state transitions to a process.
4. The adversary process has broken the 5G-AKA protocol $K_M = K_{AC1}$ for user equipment, but does not have access to the radio node/control network communications public key infrastructure.
5. There are more legitimate radio nodes than malicious/imitated adversary radio nodes, i.e., $N_{gNB} > N_A$.

6.3 Adversary and Registration Process Indistinguishability

The need for the defense-in-depth consensus algorithm is predicated on user equipment being unable to distinguish between a cryptanalytic adversary (IMSI catcher) simulating the messages emitted by the gNB/CN process and the gNB/CN process itself. Given the formalized adversary in Sect. 4.3 and the assumptions in Sect. 6.2, this section proves that the UE process cannot distinguish between the two processes given only the messages it receives. The direct proof below is truncated to start after RRC_g because the two processes do not differ until this point. The truncated adversary simulation process presented in Sect. 4.3 is:

$$A_o = \bar{d}c\ eidr.dc(i).\bar{d}c\ K_M.dc(K).\bar{d}c\ K_{sesh}.dc(f)$$

and the truncated form of Eq. 3 is:

$$HS = \nu\ sec\ s\bar{e}c\ dc.sec(a).\bar{d}c\ K_{sesh}.dc(f)| \tag{8a}$$

$$sec(ci).\bar{c}i\ eidr.ci(i).\bar{c}iK_{AC1}.ci(K).[K = K_{AC2}]s\bar{e}c\ ack \tag{8b}$$

where Eqs. 8a and 8b express represent the gNB and CN processes, respectively.

Proposition 1: *The processes HS and A_o are indistinguishable to an observing UE process.*

Proof: *Table 1 shows the comparison of the process transitions of the two processes. In transitions 1 and 6, HS has multiple internal actions that are not observed by the UE process whereas A_o has none. Therefore, the two processes are weakly bisimilar, i.e., $A_o \approx HS$.* □

Table 1. Reduction procedure of the A_o and HS processes.

		$\mathbf{A_o}$	**HS**	
1	$\Rightarrow$	$\bar{d}c\ eidr.dc(i).\bar{d}c\ K_M.$ $dc(K).\bar{d}c\ K_{sesh}.dc(f)$	$s\bar{e}c\ dc.sec(a).\bar{d}c\ K_{sesh}.dc(f)\vert$ $sec(ci).\bar{c}i\ eidr.ci(i).\bar{c}iK_{AC1}.$ $ci(K).[K = K_{AC2}]s\bar{e}c\ ack$	$\Rightarrow$
2	$\xRightarrow{\bar{d}c\ edir}$	$dc(i).\bar{d}c\ K_M.$ $dc(K).\bar{d}c\ K_{sesh}.dc(f)$	$sec(a).\bar{d}c\ K_{sesh}.dc(f)\vert$ $dc(i).\bar{d}cK_{AC1}.$ $dc(K).[K = K_{AC2}]s\bar{e}c\ ack$	$\xRightarrow{\bar{d}c\ edir}$
3	$\xRightarrow{dc\ eid}$	$\bar{d}c\ K_M.$ $dc(K).\bar{d}c\ K_{sesh}.dc(f)$	$sec(a).\bar{d}c\ K_{sesh}.dc(f)\vert$ $\bar{d}cK_{AC1}.dc(K).$ $[K = K_{AC2}]s\bar{e}c\ ack$	$\xRightarrow{dc\ eid}$
4	$\xRightarrow{\bar{d}c\ K_M}$	$dc(K).\bar{d}c\ K_{sesh}.dc(f)$	$sec(a).\bar{d}c\ K_{sesh}.dc(f)\vert$ $dc(K).[K = K_{AC2}]s\bar{e}c\ ack$	$\xRightarrow{\bar{d}c\ K_{AC1}}$
5	$\xRightarrow{dc\ K_{AC2}}$	$\bar{d}c\ K_{sesh}.dc(f)$	$sec(a).\bar{d}c\ K_{sesh}.dc(f)\vert$ $[K = K_{AC2}]s\bar{e}c\ ack$	$\xRightarrow{dc\ K_{AC2}}$
6	$\Rightarrow$	$\bar{d}c\ K_{sesh}.dc(f)$	$\bar{d}c\ K_{sesh}.dc(f)$	$\Rightarrow$
7	$\xRightarrow{\bar{d}c\ K_{sesh}}$	$dc(f)$	$dc(f)$	$\xRightarrow{\bar{d}c\ K_{sesh}}$
8	$\xRightarrow{dc\ fin}$			$\xRightarrow{dc\ fin}$

6.4 *UE* Registration and *gNB* Interrogation Process Indistinguishability

For the defense-in-depth consensus algorithm to control the mobility of radio nodes in an administrative area and deny IMSI catchers, the adversary must not be able to distinguish between a UE process and a gNB interrogation process. The only way the adversary could distinguish the processes is through prior reconnaissance. A weak bisimulation must be proven between the UE's RRG_u process and the gNB's modified RRC_{ugs} process to demonstrate the indistinguishability. This proof uses the same method as the previous section and focuses on the last transition because the two processes are identical until then.

Proposition 2: *The RRG_u and RRC_{ugs} processes are indistinguishable to an observing A_o process.*

Proof: *Table 2 shows the comparison of the last transition of the two processes. The adversary cannot observe the intra-process DiD transition that sends its SRB_m to the other radio nodes in Eq. 5 to validate. Therefore the two processes are weakly bisimilar, i.e., $RRC_u \approx RRC_{ugs}$.* □

6.5 *UE* Consensus Algorithm Termination

The user equipment consensus algorithm begins when user equipment receives the first paged warning from a radio node. To ensure that the user equipment

Table 2. Last transition of the UE and gNB interrogation processes.

	$\mathbf{RCC_u}$	$\mathbf{RCC_{ugs}}$	
$\xRightarrow{\bar{c}\ fin}$	$\bar{c}\ fin$	$\nu\ \bar{bo}(\bar{c}\ fin \| \bar{bo}\ s)$	$\xRightarrow{\bar{c}\ fin}$

can complete its registration procedure and prevent a denial-of-service attack by the adversary, the consensus algorithm that is inserted into a context hole must either return to a null process or restart the registration procedure. The first step is to rearrange Eq. 4 so that it has a summation process if the matching operation fails. This is done using π-calculus structural congruences [15], a set of axioms that state the equivalent process structures. Using the SC-Sum-Inact rule where $P + \bar{0} \equiv P$, Eq. 4 becomes:

$$b_i(s_i, n_i).b_j(s_j, n_j).[s_i = s_j]RRC_{um} \equiv$$
$$b_i(s_i, n_i).b_j(s_j, n_j).([s_i = s_j]RRC_{um} + \bar{0})$$

where $\bar{0}$ models the consensus reducing to an inactive process in all other cases, enabling the continuation of any subsequent process.

Proposition 3: *UE consensus terminates if user equipment only receive mismatched names s.*

Proof: *Proof by contradiction. Assume that $s_i = s_j$ is the only case that leads to an inactive process and all mismatches transition to the RRC_{um} process. When the consensus process reaches the matching prefix operation, only $s_i = s_j$ leads to the initiation of the RCC_{um} process. This is a contradiction.* □

6.6 Adversary and DiD Registration Process Distinguishability

This section demonstrates that user equipment will receive two paged warnings from the modified gNB/CN processes if an adversary is discovered in the administrative area. The two warning messages distinguish the A_o and DiD processes, meaning that they are not weakly bisimilar. The next three proofs demonstrate this lack of equivalence by going through the reduction of the relevant π-calculus expressions.

Proposition 4: *UE can distinguish between the A_o and DiD processes after the gNB process has interrogated A_o.*

Proof: *The modified gNB/CN processes reduced to the relevant interrogation processes is:*

$$\nu\ sec, bo_i, bo_j\ RCC_{us,i}D_{g,i}|B_{g,i}|RCC_{us,j}D_{g,j}|B_{g,j}|Q_c$$

DiD sends multiple warnings to a UE process in two cases – both radio nodes do not recognize SRB_m and one radio node recognizes SRB_m, but the control network does not when it is queried.

Case 1: Both radio nodes do not recognize SRB_1:

$$(sec(s', a).[s' = s].s\bar{e}c\ s', alerta | s\bar{e}c\ s, alerta.sec(s', a)).\bar{b}o_i\ s| \tag{9a}$$

$$(sec(s', a).[s' = s].s\bar{e}c\ s', alerta | s\bar{e}c\ s, alerta.sec(s', a)).\bar{b}o_j\ s| \tag{9b}$$

$$sec(s, q).([s = known]s\bar{e}c\ s, confirm + s\bar{e}c\ s, alerta) \tag{9c}$$

Equations 9a and 9b represent two different radio nodes and Eq. 9c represents the control network. If both gNB processes do not recognize SRB_M from the A_o process, they perform a consensus by messaging each other with name alerta over channel sec. After the consensus is complete, they page the adversary's SRB in a warning to user equipment. A UE process receives two $\bar{b_i}\ s, n_i$ names from the DiD's $B_{g,i}$ process.
Case 2: One radio node recognizes SRB_m, but the control network does not when it is queried:

$$\begin{aligned}
&[(sec(s', c).[s' = s].s\bar{e}c\ s', alerta | s\bar{e}c\ s, alerta.sec(s', c)).s\bar{e}c\ s, query.\\
&\qquad (sec(s, c).s\bar{e}c\ s, conf + sec(s, a).(s\bar{e}c\ s, alerta | \bar{b}o_i s))]|\\
&[s = known][(sec(s', a).[s' = s].s\bar{e}c\ s', conf | s\bar{e}c\ s, conf.sec(s', a)).\\
&\qquad (sec(s, c) + sec(s, a).\bar{b}o_j\ s)]|\\
&sec(s, q).([s = known]s\bar{e}c\ s, confirm + s\bar{e}c\ s, alerta)
\end{aligned}$$

During the radio node consensus, one radio node announces it recognizes SRB and the other does not. The radio node that does not sends a query message to the control network and the other radio node waits for a resolution:

$$\begin{aligned}
&[(sec(s, c).s\bar{e}c\ s, conf + sec(s, a).(s\bar{e}c\ s, alerta | \bar{b}o_i s))]|\\
&\qquad [(sec(s, c) + sec(s, a).\bar{b}o_j\ s)]|\\
&sec(s, q).([s = known]s\bar{e}c\ s, confirm + s\bar{e}c\ s, alerta)
\end{aligned}$$

The control node receives the query and, if it does not recognize SRB_m, it replies to the original querier with name alerta:

$$\begin{aligned}
&[(sec(s, a).(s\bar{e}c\ s, alerta | \bar{b}o_i s))]|\\
&\quad [(sec(s, c) + sec(s, a).\bar{b}o_j\ s)]|
\end{aligned}$$

Upon receiving the control network reply, the querying radio node simultaneously pages its $\bar{b_i}\ s, n_i$ message to user equipment and instructs the other radio node to do this as well.

In both cases, the user equipment receives two $\bar{b_i}\ s, n_i$ messages. As stated in Sect. 5.1, the messages are inserted in the user equipment consensus algorithm in Eq. 4 using a context hole. The adversary cannot do this as a result of Assumption 5. This implies that the A_o and DiD processes are not weakly bisimilar when the adversary is interrogated, i.e., $A_o \not\approx DiD$. □

Proposition 5: *UE can distinguish between the A_o and DiD processes if the adversary attempts to imitate a radio node.*

Proof: *The modified gNB/CN processes reduced to the imitation detection process is:*

$$DiD = \ \nu\ sec, bo_i, bo_j\ RCC_{us,i} I_{g,i} | B_{g,i} | RCC_{us,j} I_{g,j} | B_{g,j} | I_c$$

$$\begin{array}{r} sec(su,i).b\bar{o}_i\ su + [s = me].(s\bar{e}c\ s, imitation | b\bar{o}_i\ s) | \\ sec(su,i).b\bar{o}_j\ su + [s = me].(s\bar{e}c\ s, imitation | b\bar{o}_j\ s) | \\ sec(s,i).s\bar{e}c\ s, imitation \end{array}$$

Each radio node can detect it is being imitated upon seeing its SRB_1 used by another process. The imitated radio node simultaneously sends its $\bar{b}_i\ s, n_i$ message via the $B_{g,i}$ process and informs the control network it is being imitated:

$$\begin{array}{r} | \\ sec(su,i).b\bar{o}_j\ su + [s = me].(s\bar{e}c\ s, imitation | b\bar{o}_j\ s) | \\ sec(s,i).s\bar{e}c\ s, imitation \end{array}$$

The control network accepts the imitation warning and informs another radio node to page a second warning:

$$\begin{array}{r} | \\ sec(su,i).b\bar{o}_j\ su \\ | \end{array}$$

The second radio node then sends its $\bar{b}_j\ s, n_j$ message.

As mentioned in Sect. 5.1, user equipment can receive two $\bar{b}_i\ s, n_i$ messages by inserting the messages in the consensus algorithm in Eq. 4 using a context hole. The adversary cannot do this as a result of Assumption 5. This implies that the A_o and DiD processes are not weakly bisimilar when the adversary is attempting to imitate a radio node, i.e., $A_o \not\approx DiD$. □

Proposition 6: *UE can distinguish between the A_o and DiD processes if the adversary rejects interrogation by a radio node.*

Proof: *The modified gNB/CN processes reduced to the rejection of interrogation processes is:*

$$DiD = \ \nu\ sec, bo_i, bo_j\ RCC_{us,i}[s = \varnothing] R_{g,i} | B_{g,i} | RCC_{us,j}[s = \varnothing] R_{g,j} | B_{g,j} | R_c$$

$$\begin{array}{r} [s = \varnothing] dc(K_s).(s\bar{e}c\ K_s, reject | bo_i\ K_s) + sec(K,r).b\bar{o}_i\ K | \\ [s = \varnothing] dc(K_s).(s\bar{e}c\ K_s, reject | bo_j\ K_s) + sec(K,r).b\bar{o}_j\ K | \\ sec(k,r).[k = kn\bar{o}wn] s\bar{e}c\ K, reject \end{array}$$

The rejected radio node waits for the adversary to send its 5G-AKA key as cleartext when attempting to register with a targeted user equipment. After *intercepting the key, the radio node simultaneously pages its $\bar{b}_i\ s, n_i$ warning via the $B_{g,i}$ process and sends the key to the control network to coordinate a second warning:*

$$\begin{array}{r} | \\ [s = \emptyset]dc(K_s).(s\bar{e}c\ K_s, reject|bo_j\ K_s) + sec(K, r).b\bar{o}_j\ K| \\ sec(k, r).[k = kn\bar{o}wn]s\bar{e}c\ K, reject \end{array}$$

The control network confirms that it is an unknown key represented by the name $kn\bar{o}wn$ and coordinates a second radio node to page a warning:

$$\begin{array}{r} | \\ sec(K, r).b\bar{o}_j\ K \\ | \end{array}$$

The second radio node sends its $\bar{b}_j\ s, n_j$ message.
As mentioned in Sect. 5.1, user equipment can receive two $\bar{b}_i\ s, n_i$ messages by inserting the messages in the consensus algorithm in Eq. 4 using a context hole. The adversary cannot do this as a result of Assumption 5. This implies that the A_o and DiD processes are not weakly bisimilar when the adversary rejects interrogation by a radio node, i.e., $A_o \not\approx DiD$. □

The three propositions demonstrate that a defense-in-depth algorithm built in the 5G standard can automatically inform user equipment in an administrative area that an IMSI catcher is operating. This denies the adversary the ability to violate the privacy promises in the 5G standard as long as the DiD process wins the race condition.

6.7 UE Termination After a Successful DiD Process

After user equipment concludes its consensus algorithm regarding the paged warnings, it must disconnect from the adversary. However, in accordance with the 5G standard, user equipment will attempt to connect with the strongest radio node signal. While it is in the process of connecting with a new radio node, user equipment must ensure that it is not reconnecting to the adversary. This is accomplished by checking the name received in the paged warnings in the modified processes RRC_{um} and NAS_{um} in Sect. 5.1. According to Assumption 5, there must be a legitimate radio node for the user equipment to connect, otherwise the contextual integrity property would not be possible. The following proof demonstrates that if a legitimate radio node exists for connection after a successful consensus has been achieved, user equipment can terminate its registration process.

Proposition 7: *The modified UE registration process terminates after a successful defense-in-depth consensus.*

Proof: *Proof by contradiction. Assume that no name s or K_m exists that passes the matching operations in the RRC_{um} and NAS_{um} processes.*
Case 1: The adversary is successfully interrogated by two radio nodes. Propositions 4 and 5 show that the radio node consensus terminates and that user equipment will receive two paged warnings with the name s. After the user equipment consensus concludes, RRC_{um} proceeds to subsequent processes as long as $s \neq SRB_m$.
Case 2: The adversary avoids being interrogated by both radio nodes, but a radio node intercepts its key. This is shown in Proposition 6. The modified registration process reaches NAS_{um} that continues to termination as long as $K \neq K_m$.
Thus, the condition for the modified user equipment registration process to terminate is:

$$s \neq SRB_m \wedge K = K_m$$

are not in the paged warnings. This contradicts the premise. □

6.8 Impossibility of A_o Learning *sec*

This brief proof demonstrates that, given the contextual integrity property in Sect. 3.3, Asumption 1 and the reduced processes in Sect. 6.6, an adversary will never learn the name *sec* that would give it access to the encrypted communications of the gNB/CN process.

Proposition 8: *DiD's $B_{g,i}$ processes will never broadcast the name sec.*

Proof: *Proof by contradiction. Assume that process B_g is able to page the name sec. This implies that a radio node received the name via a registration process with user equipment or by interrogating an adversary. Given Assumption 1, this is a contradiction.* □

7 Conclusions

The consensus algorithm presented in this chapter enforces control over radio node mobility in a 5G administrative area. The algorithm builds an additional defense-in-depth layer in the 5G standard by creating a contextual integrity property that reduces the likelihood of privacy violations should vulnerabilities be found in 5G-AKA protocol implementations. In particular, the 5G infrastructure automatically warns all the user equipment in an administrative area not to connect to IMSI catchers.

The algorithm depends on multiple radio nodes deceiving and interrogating a suspected IMSI catcher to discern if its credentials are legitimate. After a consensus on the malicious nature of IMSI catcher is reached between the radio nodes and control network, multiple paged warnings are broadcast to all the user equipment in the administrative area. Following this, each user equipment must form a consensus on whether to disconnect. The π-calculus was employed to represent modifications to the 5G communications models needed for the consensus algorithms. The contextual integrity property was verified using π-calculus equivalence proofs by demonstrating that the consensus processes are distinguishable from a cryptanalytic adversary.

The research demonstrates that it is possible to use network semantics to construct context-dependent security properties such as enforcing control over which devices are permitted to broadcast in a wireless network.

One avenue for future research is to incorporate the tolerance of race conditions in the algorithm to enable computations of the efficacy of 5G network implementations should the encrypted channels fail. This would also allow for a stricter equivalence relation to be used in the proofs because processes would have to have a notion of the internal state of the other communicating processes.

The algorithm may also be extended to incorporate malicious signatures gathered by sensor-based and network-based IMSI catcher detection systems. This would turn the detection systems into a prophylaxis instead of a remedy.

In order to incorporate the proposed defense-in-depth layer in the 5G standard, it would be necessary to implement the consensus algorithm to ascertain its feasibility. Current research is focusing on an implementation that measures the latency advantage needed to succeed with regard to the race conditions. Following this, the π-calculus will have to be translated into state machines/communications models that could be formally verified, before being implemented in the 5G standard.

References

1. Basin, D., Dreier, J., Hirschi, L., Radomirovic, S., Sasse, R., Stettler, V.: A formal analysis of 5G authentication. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 1383–1396 (2018)
2. Borgaonkar, R., Hirschi, L., Park, S., Shaik, A.: New privacy threat on 3G, 4G and upcoming 5G-AKA protocols. Proc. Privacy Enhanc. Technol. **2019**(3), 108–127 (2019)
3. Cremers, C., Dehnel-Wild, M.: Component-based formal analysis of 5G-AKA: channel assumptions and session confusion. In: Proceedings of the Twenty-Sixth Network and Distributed Systems Security Symposium (2019)
4. Dabrowski, A., Petzl, G., Weippl, E.R.: The messenger shoots back: network operator based IMSI catcher detection. In: Monrose, F., Dacier, M., Blanc, G., Garcia-Alfaro, J. (eds.) RAID 2016. LNCS, vol. 9854, pp. 279–302. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45719-2_13
5. Dabrowski, A., Pianta, N., Klepp, T., Mulazzani, M., Weippl, E.: IMSI - catch me if you can: IMSI-catcher-catchers. In: Proceedings of the Thirtieth Annual Computer Security Applications Conference, pp. 246–255 (2014)
6. European Telecommunications Standards Institute, 5G. Procedures for the 5G System, ETSI Technical Specification 23.502, version 15.2.0, release 15, Sophia Antipolis (2018)
7. European Telecommunications Standards Institute, 5G. NR; Radio Resource Control (RRC) Protocol Specification, ETSI Technical Specification 38.331, version 15.3.0, release 15, Sophia Antipolis (2018)

8. European Telecommunications Standards Institute. Digital Cellular Telecommunications System (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 3GPP System Architecture Evolution (SAE); Security Architecture, ETSI Technical Specification 33.401, version 15.7.0, release 15, Sophia Antipolis (2019)
9. Jover, R.: The Current State of Affairs in 5G Security and the Main Remaining Security Challenges. arxiv.org/abs/1904.08394v2 (2019)
10. Khan, M., Ginzboorg, P., Järvinen, K., Niemi, V.: Defeating the downgrade attack on identity privacy in 5G. In: Cremers, C., Lehmann, A. (eds.) SSR 2018. LNCS, vol. 11322, pp. 95–119. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04762-7_6
11. Milner, R.: Communicating and Mobile Systems: The π-Calculus. Cambridge University Press, Cambridge (1999)
12. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, II. Inf. Comput. **100**(1), 41–77 (1992)
13. Morrissey, C.: Windows quality updates primer, Microsoft IT Pro Blog, 21 July 2021
14. Park, S., Shaik, A., Borgaonkar, R., Seifert, J.: Anatomy of commercial IMSI catchers and detectors. In: Proceedings of the Eighteenth ACM Workshop on Privacy in the Electronic Society, pp. 74–86 (2019)
15. Sangiorgi, D., Walker, D.: The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press, Cambridge (2003)
16. Shaik, A., Borgaonkar, R., Park, S., Seifert, J.: New vulnerabilities in 4G and 5G cellular access network protocols: exposing device capabilities. In: Proceedings of the Twelfth Conference on Security and Privacy in Wireless and Mobile Networks, pp. 221–231 (2019)
17. Steig, S., Aarnes, A., Do, T., Nguyen, H.: A network based IMSI catcher detection. In: Proceedings of the Sixth International Conference on IT Convergence and Security (2016)
18. Stirling, C.: Modal and temporal logics for processes. In: Moller, F., Birtwistle, G. (eds.) Logics for Concurrency. LNCS, vol. 1043, pp. 149–237. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-60915-6_5
19. Yocam, E., Gawanmeh, A., Alomari, A., Mansoor, W.: 5G mobile networks: reviewing security control correctness for mischievous activity. SN Appl. Sci. **4**(11), 304 (2022)
20. Zhang, M.: Provably-secure enhancement of 3GPP authentication and key agreement protocol. Cryptology ePrint Archive, vol. 2003, p. 92 (2003)

Infrastructure Security

Modeling and Assessing the Impacts of Cyber Threats on Interdependent Critical Infrastructures

Valeria Bonagura[1], Chiara Foglietta[1], Stefano Panzieri[1](✉), Massimiliano Rossi[2], Riccardo Santini[2], Monica Scannapieco[2], and Luisa Franchina[3]

[1] University Roma Tre, Rome, Italy
stefano.panzieri@uniroma3.it
[2] National Cybersecurity Agency, Rome, Italy
[3] Hermes Bay, Rome, Italy

Abstract. Critical infrastructures are complex networks with physical, geographical, logical and cyber interdependencies whose disruption can cause serious impacts to citizenry and society. Meanwhile, the use of information and communications technology to manage physical processes in critical infrastructure assets has significantly increased their cyber attack surfaces. The increased threats have led to the creation of national and international cyber security agencies to promote awareness of cyber threats and coordinate responses to cyber attacks.

In 2019, Italy set up the National Security Perimeter for Cyber, a regulatory construct that stipulates measures for guaranteeing the safety and security of public and private entities that provide essential functions and services. The law associated with the regulatory construct requires the covered entities to accurately describe their networks, information and communications technology systems and related services. The 2021 Italian legislation that established the National Cybersecurity Agency requires all National Security Perimeter for Cyber entities to inform the national agency about their assets. The National Cybersecurity Agency also collects detailed infrastructure information as well as reports about cyber attacks from the entities.

This chapter describes an ongoing research effort that supports Italian legislative requirements. In particular, it demonstrates how the consequences of cyber threats can be assessed in complex scenarios using an agent-based simulator that evaluates the National Cybersecurity Agency model under ransomware and distributed-denial-of-service attacks on interconnected Italian infrastructures.

Keywords: Critical Infrastructure Modeling · Simulation · Cyber Attacks · Cyber Impacts · Italian National Security Perimeter for Cyber

1 Introduction

Modern control systems integrate physical processes with communications and computational resources that improve system efficiency and operational performance. In recent years, attention has focused on a particular class of control

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 143–163, 2024.
https://doi.org/10.1007/978-3-031-49585-4_7

systems called cyber-physical systems. Several definitions have been proposed for cyber-physical systems and their functionalities [20]. However, their essential behavior is that they act independently, cooperatively or as “systems of systems.”

From a practical control systems perspective, cyber-physical system behavior is characterized by nonlinear interactions between discrete phenomena (digital systems) and continuous phenomena (physical systems). Several techniques are required to capture and analyze behavior at the low level such as discrete control logic, communications and distributed computing effects as well as at the global level. While the integration improves system efficiency and operational performance, the threats posed by system intrusions by adversaries are elevated. Additionally, the increased amount of sensor data complicates the task of detecting malicious attacks.

Examples of cyber-physical systems include supervisory control and data acquisition (SCADA) systems, transportation networks, electric power generation and distribution networks, water and gas distribution networks, advanced communications systems and, more generally, critical infrastructures. The systems straddle the information technology (IT) and operational technology (OT) domains with cyber-physical security becoming a focus of attention due to the convergence of previously-disjointed security functions.

Operational technology security has historically lagged information technology security. This is largely because operational technology has prioritized safety and uptime without much regard for cyber security [19]. However, this situation must change on account of digital integration. Indeed, digital integration has increased the attack surfaces of critical infrastructure assets, causing them to be targeted by cyber attacks by malicious actors that leverage the ubiquitous connectivity provided by information technology to access and breach systems that were once thought to be impenetrable [9,18].

This situation has highlighted the fragility of cities, states and nations [1]. A well-cited example is the 2021 attack on Colonial Pipeline in the United States [7,15,16]. Unknown, well-resourced actors successfully targeted the gasoline transportation infrastructure. Although the company and the U.S. government cooperated to restore full capacity, the critical infrastructure was shut down for several days. The impacts were serious – 71% of gas stations in the Charlotte, North Carolina metropolitan area were short of or ran out of fuel.

The Colonial Pipeline attack is just one of many incidents reported around the world. Nation states have become cognizant of the serious cascading impacts of cyber attacks on critical infrastructure assets and, ultimately, on society. Analyzing the interdependencies between critical infrastructure assets at the regional, national and international levels are essential to understanding the consequences of adverse events. It is the responsibility of nation states to define appropriate cyber security strategies and institute regulatory constructs that will render critical infrastructure assets safe, secure and resilient to adverse events.

Recent reforms related to the Italian cyber ecosystem have led to the enactment of an Italian national law – National Security Perimeter for Cyber – that identifies key private and public entities in Italy, including critical infrastructure assets that perform essential functions or provide essential services, and endeavors to protect them from cyber attacks [25]. According to the law, every perimeter subject is required to inform the National Cybersecurity Agency of its information and communications technology (ICT) assets, networks, information systems and related services, and share data about cyber attacks and the effects observed on their infrastructure assets.

This chapter describes an ongoing research effort that supports the Italian legislative requirements. In particular, it demonstrates how the consequences of cyber threats can be assessed in complex scenarios using an agent-based simulator that evaluates the National Cybersecurity Agency model under chains of synthetic ransomware and distributed denial-of-service attacks on interconnected Italian infrastructures. The research leverages the mixed holistic reductionist approach, a hierarchical method that decomposes infrastructures into simple elements at multiple levels of abstraction [6]. The approach employs data drawn from the national security perimeter to generate an impact model of interconnected infrastructures for analyzing hypothetical scenarios. The agent-based CISIApro 2.0 simulator [4,13] implementing the mixed holistic reductionist approach is employed to convey the impacts of cyber attacks on interconnected infrastructures in terms of the confidentiality, integrity and availability (CIA) security triad.

2 Related Work

Critical infrastructure assets have achieved high degrees of interoperability due to the pervasive integration of information and communications technology to the point where interdependencies couple infrastructure assets regardless of their nature, type or geographic locations [22]. Due to the high degree of interoperability, it is vital to model critical infrastructure interdependencies to assess the consequences of adverse events such as natural disasters, failures and cyber attacks in terms of the CIA security triad. At this time, no single modeling tool fits every need. However, depending on the application and available information, some tools are more suitable than others.

EPANET2 is an open-source tool that is widely used to model water distribution systems [23]. The tool, which leverages network analysis and hydraulic simulation to model water system behavior over time, has been used to simulate the effects of cyber attacks on water distribution systems and identify potential vulnerabilities.

Ficco et al. [12] developed a hybrid, distributed simulation platform for conducting cyber security evaluations of large-scale critical infrastructure systems. The platform supports the integration of multiple simulated environments and the use of penetration testing and monitoring tools to evaluate complex, distributed experimental scenarios in the cloud.

The DOMINO simulation tool enables critical infrastructure asset managers to create and update questionnaires pertaining to the autonomy of their facilities in the absence of primary and alternative resources [14]. Asset managers are assisted in ensuring business continuity via an early warning system that provides alerts about potential problems. The DOMINO tool provides insights into potential cascading temporal and spatial impacts in training scenarios.

The Critical Infrastructure Program for Modeling and Analysis (CIPMA) is an Australian public-private sector approach that identifies and assesses critical infrastructure risks, recommends prioritization of investments and evaluates mitigation strategies and business continuity plans [5]. The communications, energy, water and transportation sectors have leveraged CIPMA to develop improved emergency management responses. CIPMA has also been used to study large-scale scenarios, including a cyclone in Queensland, gas supply disruption on the North West Shelf and submarine cable shelf/cable outages [2].

This research employs the CISIApro 2.0 agent-based simulator [4,13] to analyze the consequences of adverse events on interconnected infrastructures. In the CISIApro 2.0 simulator, each infrastructure is decomposed into agents that describe complex behaviors. Details about the CISIApro 2.0 simulator are provided in Sect. 5.2.

3 National Security Perimeter for Cyber

A nation state is responsible for defining strategies focused on planning, coordinating and implementing measures that ensure the country's cyberspace is secure, safe and resilient while ensuring its citizenry can leverage the competitive advantages of cyberspace with complete protection of their fundamental rights and freedoms.

Since 2013, the Italian Government has invested considerable effort to keep pace with technological advances in the cyber domain. Over time, it has instituted a number of measures designed to acquire, develop and strengthen national cyber capabilities, and to guarantee institutional uniqueness of direction and action with respect to cyber security as an area of intervention that is national in scale and engages all stakeholders.

At the European Union (EU) level, the EU Network and Information Security (NIS) Directive 2016/1148 [11] specifies measures intended to achieve a "high level of security of network and information systems in the national sphere, contributing to increase the common level of security in the European Union." The directive was adopted into Italian law by Legislative Decree of May 18, 2018 (L.D. no. 65/2018) [24], which dictates the legislative framework of measures for securing networks and information systems and identifies the entities responsible for implementing the obligations under the EU NIS Directive.

This section highlights the Italian National Security Perimeter for Cyber Law [25] as a regulatory construct that covers more entities than the EU NIS directive and incorporates more compulsory rules. Following this, the section introduces recent Italian cyber ecosystem reforms.

On September 21, 2019, Law Decree no. 105/2019 – Urgent Measures Concerning the National Security Perimeter for Cyber (and Special Powers of the Government in the Strategic Sectors) [25] – was enacted by the Italian Government. The decree established the "National Security Perimeter for Cyber" that introduces measures to guarantee safety standards for networks and information systems as well as information technology services for public administrations, private and public entities and critical infrastructure assets that perform essential state functions or provide essential services in the civil, social and economic domains and whose malfunction may pose risks to national security.

The legislation has established provisions that are implemented via four Prime Ministerial Decrees and a Presidential Decree in order to:

- Identify the public and private entities falling within the National Security Perimeter for Cyber and the criteria for creating the lists of networks, information systems and relevant services (DPCM no. 131/2020) [26].
- Define the procedures for the notification of cyber incidents to the Computer Security Incident Response Team of Italy that impact networks, information systems and information services (DPCM no. 81/2021) [28].
- Define the evaluation procedures for information and communications technology assets used in the National Security Perimeter for Cyber and notify the National Assessment and Certification Center in charge of conducting security assessments with the goal of verifying the absence of known vulnerabilities in information and communications technology assets, systems and services (DPR no. 54/2021) [27].
- Identify the categories of information and communications technology assets, systems and services used by the entities included in the National Security Perimeter for Cyber and the procurement of communications technology assets evaluated by the National Assessment and Certification Center (DPCM no. 198/2021) [30].
- Define the accreditation procedures for Accredited Evaluation Laboratories and coordination procedures for the National Assessment and Certification Center, Accredited Evaluation Laboratories and Evaluation Centers belonging to the Italian Ministry of Defense and Italian Ministry of the Interior (DPCM no. 92/2022) [31].

These goals are being pursued through recent reforms of the national cyber ecosystem enacted by the Legal Decree of June 14, 2021 (L.D. no. 82/2021) [29]. The decree established the National Cybersecurity Agency of Italy with the mission of rationalizing and consolidating the fragmented expertise existing at the national level in compliance with the competencies attributed to other administrations by legislation in force, and further enhancing the cyber security and resilience for the purposes of protecting national security in cyberspace. As the national authority, the National Cybersecurity Agency of Italy develops the National Cybersecurity Strategy [21].

Furthermore, pursuant to L.D. no. 82/2021 [29], the National Cybersecurity Agency of Italy is designated as the exclusive competent national authority and single point of contact for the purposes referred to in the legislation on the security of networks and information systems (NIS) [11], National Cybersecurity Certification Authority, National Coordination Center with reference to the European Cybersecurity Competence Centre and Network [8] and central element of the National Security Perimeter for Cyber. It should be noted that these competencies were previously attributed to a plurality of institutional actors and that the Computer Security Incident Response Team of Italy and National Assessment and Certification Center are established within the National Cybersecurity Agency.

4 Ontology-Based Approach

Decree of the President and the Council of Council of Ministers of July 30, 2020 (DPCM no. 131/2020) [26] assigns to every public and private entity in the National Security Perimeter for Cyber the mandatory duty to inform the National Cybersecurity Agency of its information and communications technology networks, information systems and related services by compiling a comprehensive list. To support these entities, the National Cybersecurity Agency has designed a formal model for accurately describing all the relevant assets (e.g., information systems, routers and services) and their relationships (e.g., structures and dependencies). The model captures the characteristics of the two key domains in which the entities perform essential state functions and/or provide essential information and communications technology and operational technology services.

The National Cybersecurity Agency model can be viewed as an ontology because it formalizes domain knowledge in a structured manner using two types of components. The first component type is entities, which are defined as classes of objects of interest with homogeneous characteristics along with their related properties. The second component type is the relationships between entities.

A domain is described by accurately defining the entity instances along with their characteristic properties and relationships. Additionally, the model enables the expression of the applicable constraints.

The National Cybersecurity Agency model, which is called the perimeter ontology, has four logical sections:

- Entity description, information and communications technology functions and/or services, and the relationships between them.
- Information and communications technology networks, systems and services, hardware and software components and nodes. Nodes are components collected in physical or logical spaces such as data centers and electrical substations.
- Outgoing dependencies such as external services on which entities depend.
- Geographical locations of all the components listed above.

Details about the perimeter ontology are not provided in this chapter for national security reasons. The complete lists of networks, information systems and services pertaining to the entities are also protected by confidentiality clauses.

However, the authors of this chapter believe that it is important to present the approach for collecting perimeter data in a structured manner using an ontology. The approach has three principal advantages. One is ambiguity reduction at the data sources because the semantics of the collected data is formally specified at the data collection stage. The second is the reduction of the complexity of the steps following data collection, especially related to the storage and analysis of the collected data. The third is the quality (completeness) of the collected data due to the use of well-defined and somewhat rigid collection tools.

5 Modeling Approach

This section describes the mixed holistic reductionist approach for modeling interdependent critical infrastructures. Also, it describes the CISIApro 2.0 simulator that is designed to assess the impacts of adverse events in complex modeled critical infrastructure scenarios.

5.1 Mixed Holistic Reductionist Approach

The mixed holistic reductionist approach leverages the benefits of holistic and reductionist thinking [6]. The approach provides a roadmap for meticulously modeling critical infrastructures and their interdependencies.

The mixed holistic reductionist approach describes interconnected infrastructures as a set of networks. Each infrastructure is described at different abstraction levels to capture phenomena that emerge at different granularities. The idea is to integrate the advantages of the holistic and reductionist approaches.

Infrastructures are viewed as distinct entities with clearly-defined borders and functional attributes in holistic modeling to provide a comprehensive and global picture. When considering an infrastructure as a whole, it is possible to identify and describe the infrastructures as well as their regional reaches. At this level, the amount of data required for modeling operations is small and may be available in open databases.

On the other hand, the reductionist paradigm emphasizes the need to carefully study the roles and behaviors of individual components to fully understand the entire system. Specifically, the reductionist approach breaks down each component into its inputs and outputs. Relations between machinery and individual parts are easily specified at this level of abstraction.

Service efficiency (referred to as "service") functions as the link between the holistic and reductionist levels. This layer describes the functional connections between infrastructures and components at varying granularities. Consumers and other connected infrastructures reside in the middle layer between the holistic and reductionist levels.

Different systems require different levels of analysis, and their limitations are lost in complex case studies. The mixed holistic reductionist approach allows for top-down or bottom-up analyses of network interactions at various levels. It also enables critical infrastructures to be modeled at different degrees of abstraction based on the available data.

Figure 1 shows a mixed holistic reductionist model representation starting from the perimeter ontology. The central nodes are in the holistic layer, the dark

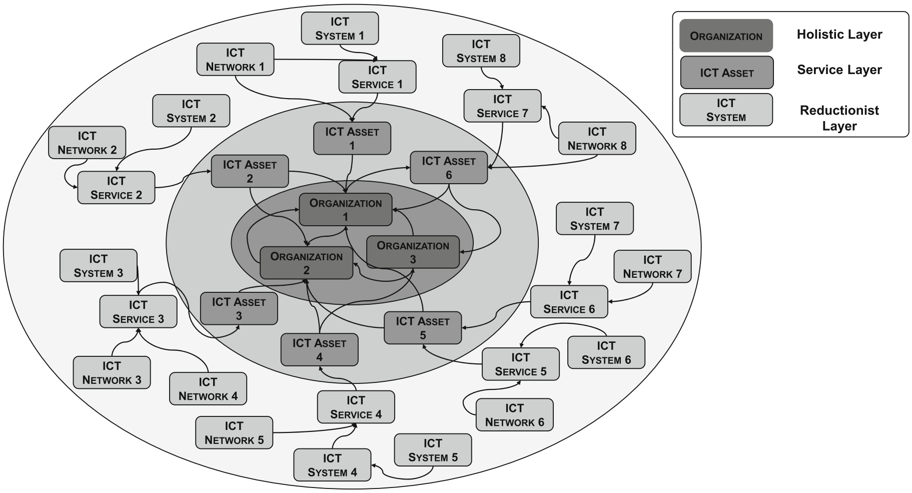


Fig. 1. Mixed holistic reductionist model representation.

grey nodes are in an intermediate layer and the external nodes are in the reductionist layer. The transition from the ontology to the mixed holistic reductionist model is not obvious, and human intervention is required to resolve conflicts between the two views. The model shown in Fig. 1 is the original proposal for modeling interdependencies and lacks direct correspondence with the ontology in terms of entities and relationships.

The holistic layer contains all the agents representing all the entities that are part of the perimeter or are directly linked to an entity in the perimeter. Due to the difficulty of determining the particular devices on which connections occur, the corresponding agents are connected among themselves primarily to exchange cyber risk. Cyber risks are related to cyber attack impacts, which are primarily confidentiality and integrity (a data breach at an entity has no direct impact on information availability, but it can impact the entity's trust and reputation at the holistic level). When data is not available, the model may contain blocks related to the entity without additional details.

An entity provides essential services to its customers and other entities. Each service is produced by an information and communications technology asset. Therefore, each service layer contains agents called information and communications technology assets that represent parts of the information and communications technology network that are necessary to deliver essential services. Entity blocks are linked to information and communications technology assets in two ways. The first corresponds to the information and communications technology assets of an infrastructure that provide resources, faults and cyber attacks to the entity blocks associated with the same infrastructure. The second corresponds to the information and communications technology assets that produce specific resources (services) used by other infrastructures.

As shown in Fig. 1, an information and communications technology asset comprises systems, networks and services. The three categories represent devices (hardware and software) that are fundamental to delivering services. These elements are part of the reductionist layer of the model. Hence, the blocks represent the information and communications technology portions of the operational technology environment. Note that all the blocks are not depicted in the figure. The reductionist layer contains some blocks that are cyber-physical components such as data centers, buildings and electrical substations. A cyber-physical system incorporates several components needed to produce a service, but also contains some information and communications technology components.

The case study described in this chapter also considers the possibility of infrastructures that are interconnected at all the layers in the model. For instance, an airline company, which is considered to be a reductionist component, depends on electricity supplied by a utility whose information and communications technology assets need bank services to collect payments from customers. However, impacts such as confidentiality, integrity and availability are exchanged at the entity level, namely, at the holistic layer.

5.2 CISIApro 2.0 Simulator

The Critical Infrastructure Simulator with Interdependent Agents 2.0 (CISIApro 2.0) [4,13] is employed to evaluate the consequences of adverse events on interconnected critical infrastructures. The simulator engages agent-based modeling using three main components, agents, simple interaction rules and the environment in which the agents are placed. Multiple agents acting simultaneously according to the interaction rules model complex systems. In agent-based modeling, central control does not drive agent behavior. Instead, following the local rules leads to an outcome or aggregate behavior that adapts to the environment or reacts to adverse situations. Thus, an agent-based model is a simply a set of agents that follow simple rules to collectively generate an emergent property or behavior. The main drawback of agent-based modeling is that it requires high levels of detail to provide adequate predictions. As a result, the accuracy of agent-based modeling depends on the specificity of the underlying assumptions.

Figure 2 shows the CISIApro 2.0 agent representation. Each infrastructure is decomposed into agents with the same overall input and output structures. Each agent receives resources, faults and cyber attacks from upstream agents and sends resources, faults and cyber attacks to downstream agents. Resources are supplies of materials, quantities and other assets that are required by entities to function effectively. Faults include malfunctions and natural events that must be exploited to assess different outcomes, depending on the details of the initial adverse events. Cyber attacks are malicious activities that attempt to collect, disrupt, deny, degrade or destroy information and communications technology resources. In a CISIApro 2.0 simulation, resources, faults and cyber attacks are exchanged among agents.

Agent state is identified by its operational level. The operational level indicates an agent's ability to function properly and execute its tasks. Every agent

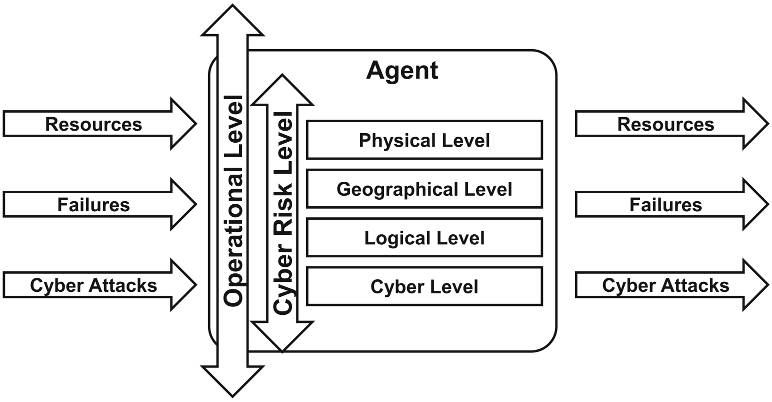


Fig. 2. CISIApro 2.0 agent representation.

has an internal state variable that represents its internal behavior based on the evaluation of resources, faults and cyber attacks. Based on its operational level, each agent sends resources, faults and cyber attacks to downstream agents.

To better handle cyber attacks and evaluate their consequences, each agent has an additional state variable called the cyber risk level that identifies how the agent is affected by internal and incoming cyber attacks. Cyber risk is based on the CIA triad. The CIA triad may be difficult to apply in industrial automation and control environments, but the three security goals are useful to deal with information in classical information technology environments and to spread information about cyber attacks in industrial automation and control environments. In fact, the CIA triad is invaluable when it comes to determining the impacts of cyber attacks on the telecommunications network portions of industrial automation and control systems.

It is instructive to clarify the meanings of the CIA terms and their relationships in industrial automation and control environments. Real-time processes at Purdue levels 0 to 2 [3,32] are often exempt from the confidentiality requirement because operational and real-time parameters are not viewed as secrets. Secret manufacturing formulas are to be protected and this must be done in the information technology and industrial automation and control zones [17]. Since real-time operating data has not been tampered with, it can be trusted. However, the industrial automation and control zone is viewed as being insecure by design. Therefore, by employing trustworthy design, perimeter security and supplemental cyber security, the integrity of the industrial automation and control zone can be guaranteed.

Dependability, productivity and business continuity standards for the industrial automation and control zone also address availability. Similar to integrity, availability must be guaranteed through trustworthy architectures, dependable goods and trustworthy software.

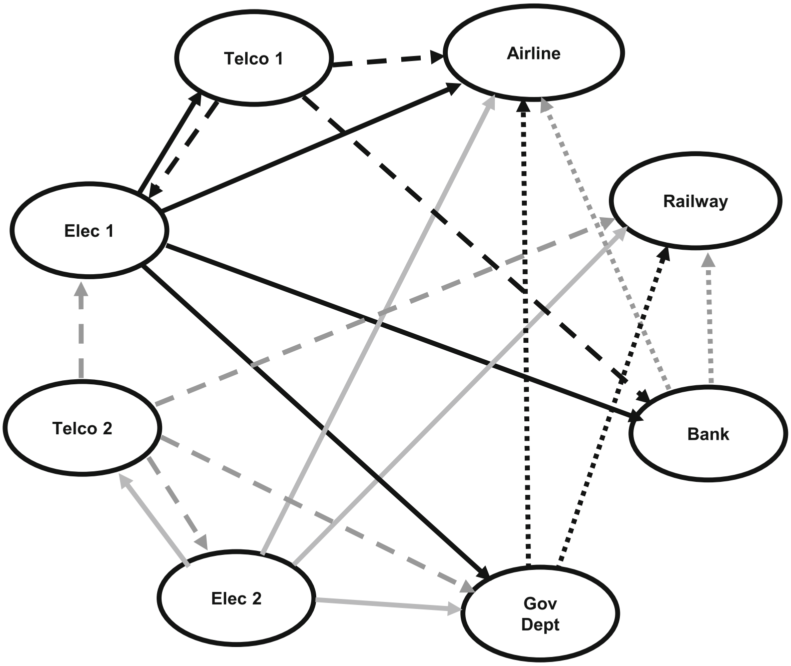


Fig. 3. Interdependencies between the interconnected infrastructures.

In a CISIApro 2.0 simulation, the availability of information transmitted by a telecommunications network is captured by its operational level. In contrast, confidentiality and integrity are expressed as cyber risk levels. As shown in Fig. 2, the operational and cyber risk level metrics may be interconnected and partially overlap.

6 Case Study

A case study involving eight interconnected infrastructures is used to demonstrate the efficacy of the mixed holistic reductionist approach and CISIApro 2.0 simulation. The interconnected infrastructures include two telecommunications companies, two electrical power distribution companies, a railway company, an airline company, a bank and a government department.

Figure 3 shows the interdependencies between the eight interconnected infrastructures. The two telecommunications companies provide services such as Internet access and mobile and backbone telecommunications. The two electrical power distribution companies provide electricity for equipment as well as to buildings, railway stations and airports. The bank processes customer payments to the railway and airline companies. The government department issues licenses for rail transport of people and goods and regulates airline company operations.

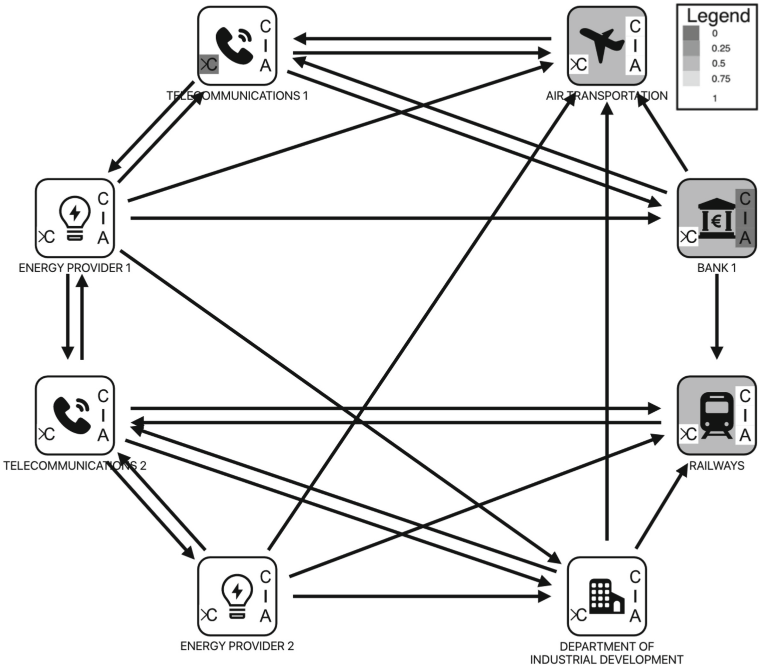


Fig. 4. Bank infrastructure view after the ransomware attack.

6.1 Ransomware Attack

The first scenario involves a ransomware attack on customer payment services provided by the bank. A ransomware attack enables an adversary to seize control of the targeted assets and demand a ransom in exchange for availability of the assets [9,10]. In 2022, ransomware was one of the top cyber threats, affecting all sectors indiscriminately and with numerous high-profile cases [10].

Figure 4 shows the bank infrastructure view after the ransomware attack. The ransomware disrupts the bank services that process customer payments to the railway and airline companies. All the entities have the operational levels expressed by the gray scale in the icon backgrounds, the CIA triads on the right-hand sides of the icons and the cyber risk due to the interconnected infrastructures indicated by $> C$ in the bottom-left corners of the icons.

As expected, the ransomware attack causes drastic reductions in the three components of the CIA triad at the bank. However, no impacts are observed on the primary transportation functions of the railway and airline companies; as a result, the operative levels of the two companies are 0.5. Additionally, the possibility exists that the attack impacts the telecommunications company providing services to the bank when the adversary conducts lateral movements and exploits vulnerabilities to enter and control remote systems in the interconnected telecommunications network.

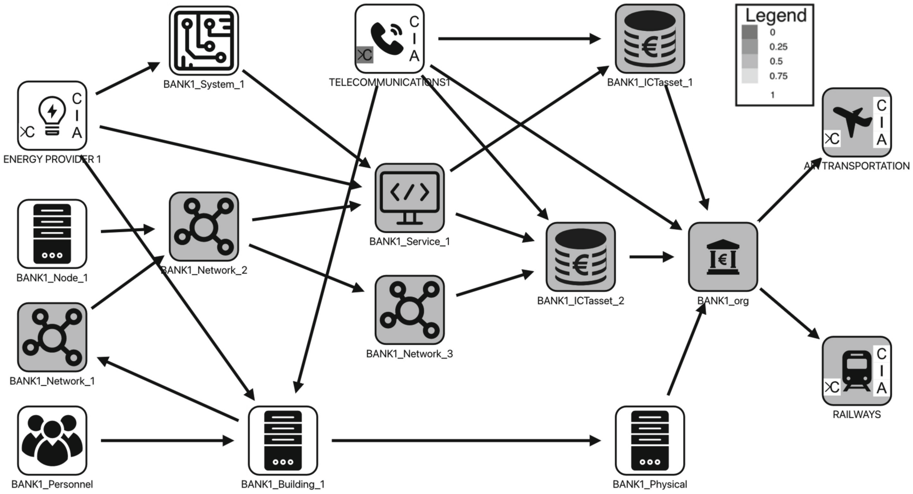


Fig. 5. High-level bank representation.

Figure 5 shows the high-level representation of the bank. Since the ransomware attack targets an information and communications technology service common to the two information and communications technology assets at the bank, the operational level of the bank drops to zero.

The railway company (Fig. 6) and airline company (Fig. 7) are also affected partially by the ransomware attack. The impacted services at the two companies primarily relate to ticket sales. Specifically, the two companies rely on telecommunications and bank services for ticket sales and the observed impact is mainly on the transactions. The combination of the services supplied by the two information and communications technology assets is evaluated using the average operation. The operational levels of the railway and airline companies are both equal to 0.5, where one corresponds to fully operational.

As mentioned above, the information and communications technology systems and networks of the bank and telecommunications company are linked. Thus, due to lateral movements and the exploitation of vulnerabilities by the adversary, the telecommunications company may be affected in a different manner by the ransomware attack.

Figure 8 shows the impact on the telecommunications company due to lateral movements from the bank network and vulnerability exploitation. The telecommunications network does not have a direct impact on the functional level; instead, the impact is on company trust and reputation. The operational level of the telecommunications company is one because there is no impact on telecommunications services.

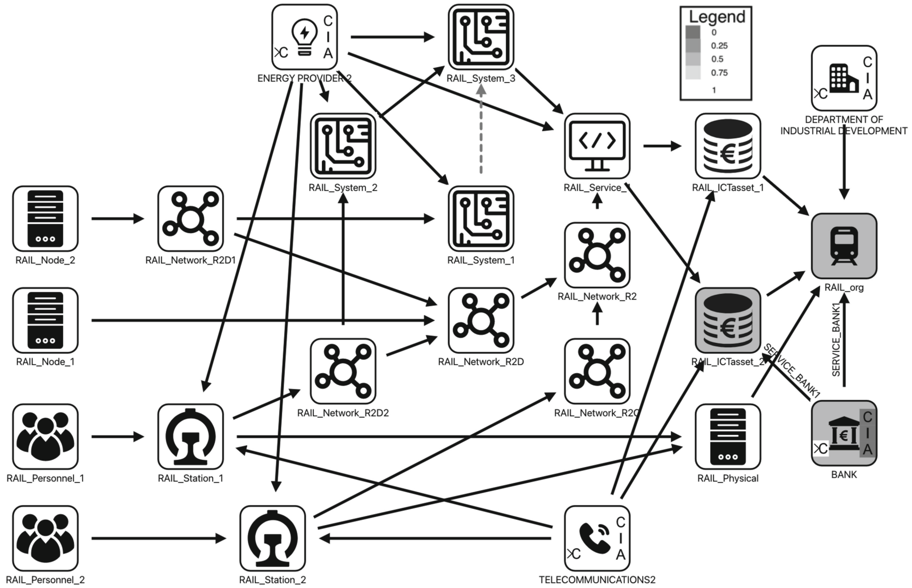


Fig. 6. Railway company representation.

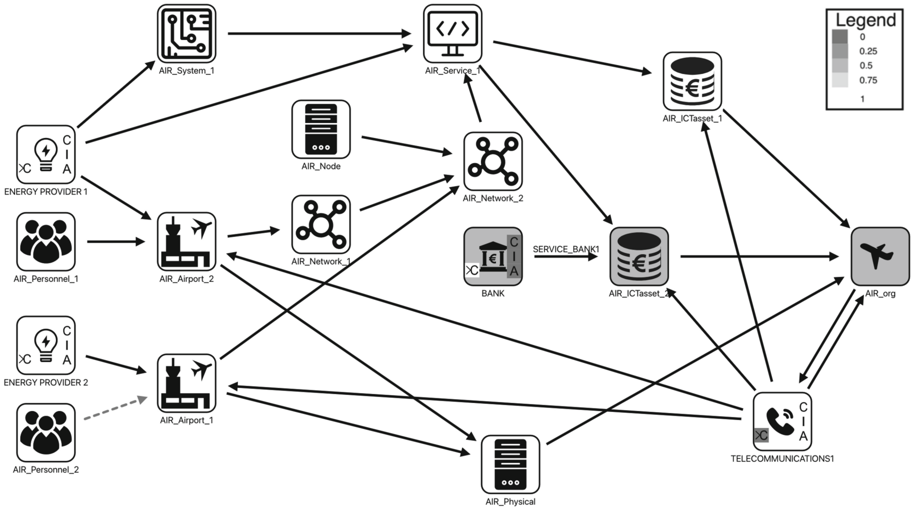


Fig. 7. Airline company representation.

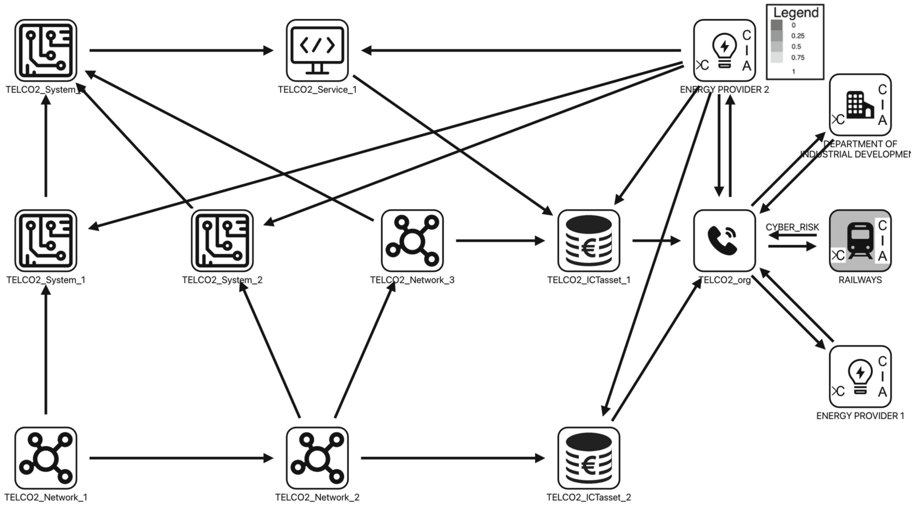


Fig. 8. Telecommunications company representation.

6.2 Distributed Denial-of-Service Attack

A denial-of-service attack directly impacts the availability of computer and network resources, causing temporary problems for customers who rely on services. A typical denial-of-service (DoS) attack floods a target with traffic or sends information that triggers a crash. A distributed denial-of-service (DDoS) attack occurs when multiple systems orchestrate a synchronized denial-of-service attack on a single target. The main difference is that, instead of being attacked from one location, the target is attacked simultaneously from multiple locations. Distributed denial-of-service was ranked the most serious cyber threat in 2022 whereas ransomware was ranked the most serious cyber threat in 2021 [9].

Figure 9 shows a synoptic view of the infrastructures after a distributed denial-of-service attack on an information and communications technology service agent at the second telecommunications company. As expected, the telecommunications service disruption has a profound impact on all the other interconnected infrastructures.

Figure 10 shows the impact of the distributed denial-of-service attack on the second telecommunications company. All the services provided by the company are affected. The situation is more serious than the one shown in Fig. 8. This is because the second telecommunications company provides services to the two electric power distribution companies, railway company and government department.

Figure 11 shows the impact of the telecommunications distributed denial-of-service attack on the railway company. Railway operations are highly impacted by the telecommunications service disruption because the company has only one telecommunications provider whose services are used to coordinate information

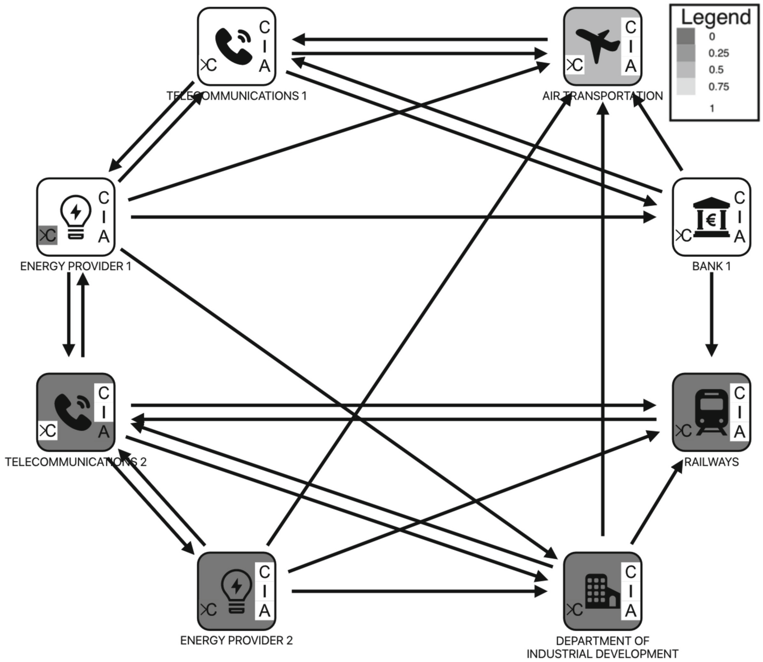


Fig. 9. Overall impact of DDoS attack on telecommunications company.

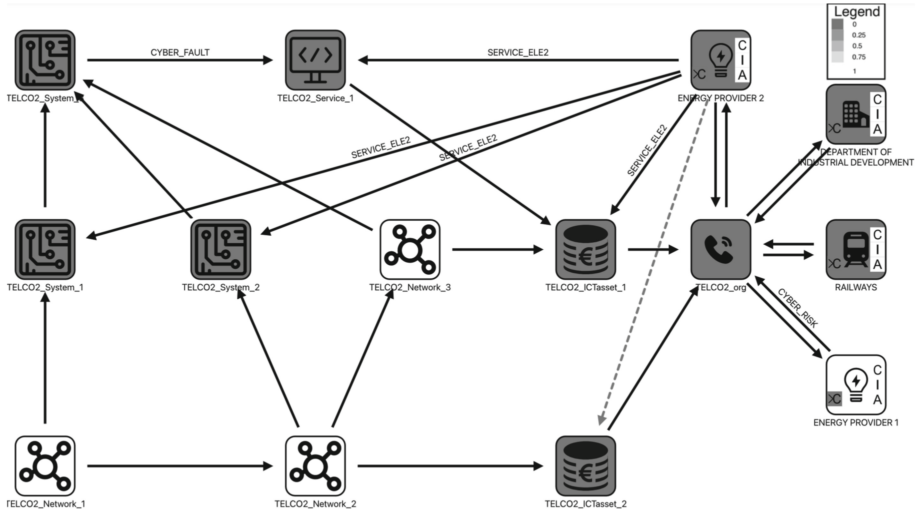


Fig. 10. Impact of DoS attack on telecommunications company.

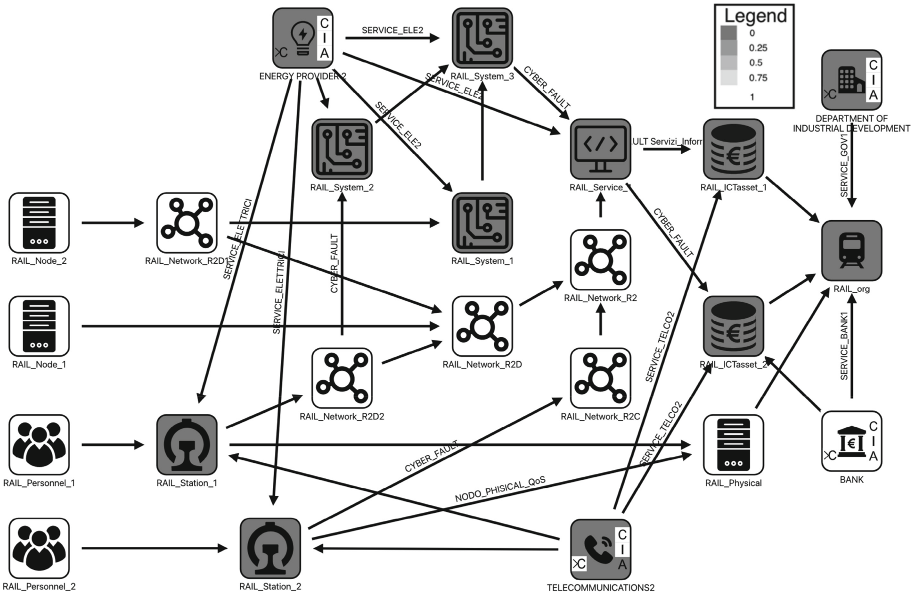


Fig. 11. Impact of telecommunications DDoS attack on railway company.

and communications systems and services at two railway stations. The railway company is also potentially impacted by the disruption of electricity from its electric power distribution company that receives services from the targeted telecommunications company.

The government department relies on telecommunications services to perform its functions. Figure 12 shows that the telecommunications distributed denial-of-service attack impacts the government department building as well as its two information and communications technology assets.

Figure 13 shows the impact of the telecommunications distributed denial-of-service attack on electric power distribution. The two electric power distribution companies have different supply chains. The company shown in Fig. 13 has a single telecommunications service provider whereas the other company has two telecommunications service providers. As a result, the impacts are completely different. The impact on the first company is significant whereas the second company is not affected.

The impact of the telecommunications distributed denial-of-service attack on the electric power distribution company in Fig. 13 leads to negative impacts on other infrastructures. The railway system needs electricity for its information and communications technology systems (Fig. 11) and the airline company needs electricity for one of the two airports (Fig. 14).

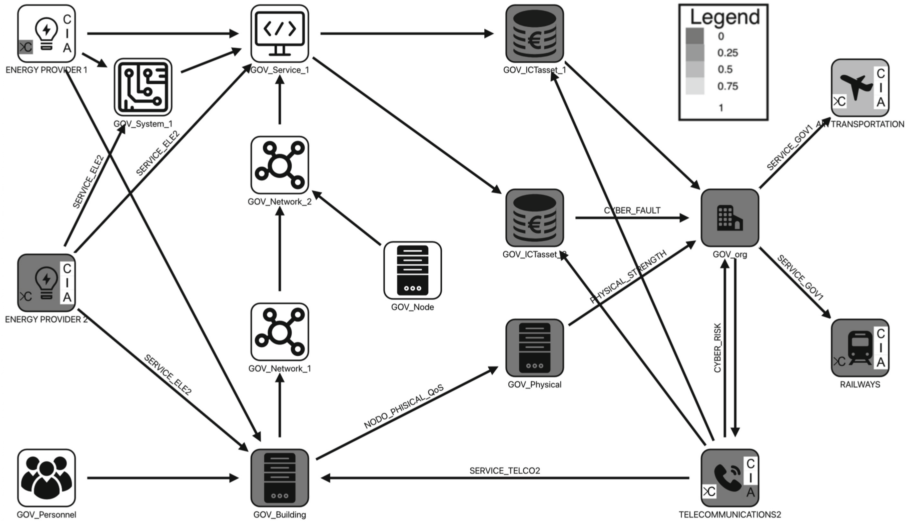


Fig. 12. Impact of telecommunications DDoS attack on government department.

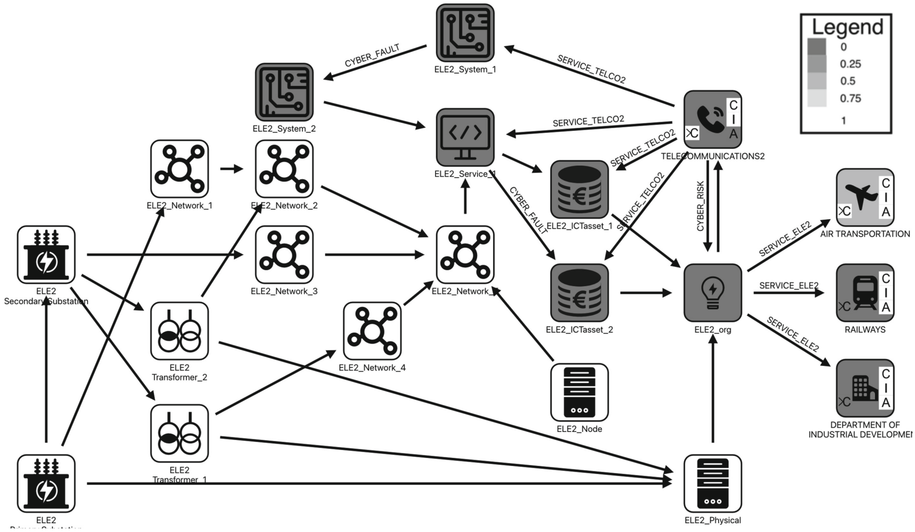


Fig. 13. Impact of telecommunications DDoS attack on electric power distribution.

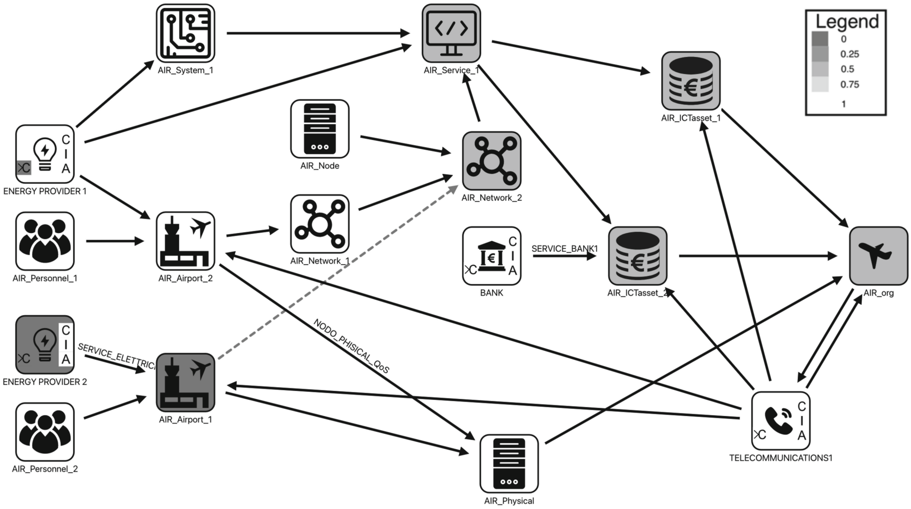


Fig. 14. Impact of telecommunications DDoS attack on airline company.

7 Conclusions

Nation states have become cognizant of the serious cascading impacts of cyber attacks on critical infrastructure assets and, by extension, on society. The increased threats have led to the creation of national and international cyber security agencies to promote awareness of cyber threats and coordinate responses to cyber attacks. By law, all the public and private entities in the Italian National Security Perimeter for Cyber must inform the National Cybersecurity Agency about all their information and communications technology assets, networks, information systems and services. This information is submitted using an ontology provided by the National Cybersecurity Agency. However, modeling interdependent infrastructures and assessing the impacts of cyber attacks are complex problems.

This chapter has demonstrated how the mixed holistic reductionist approach can be employed to decompose each infrastructure into different abstraction layers, model their interdependencies and evaluate the effects of adverse events. By employing the mixed holistic reductionist approach with the ontology proposed by the Italian National Security Agency, the CISIApro 2.0 agent-based simulator can be used to model complex cyber scenarios involving the Italian National Security Perimeter for Cyber. The case study shows that the proposed approach can effectively assess the consequences of ransomware and distributed denial-of-service attacks on the connected infrastructures in terms of confidentiality, integrity and availability.

Future research will model additional interconnected infrastructures. The model will also be enhanced by considering propagation delays involving the data exchanged between infrastructures and the interactions between physical processes and information and communications technology services.

References

1. Alladi, T., Chamola, V., Zeadally, S.: Industrial control systems: cyberattack trends and countermeasures. Comput. Commun. **155**, 1–8 (2020)
2. Amélie, G., Aurélia, B., Emmanuel, L., Mohamed, E., Gilles, D.: The challenge of critical infrastructure dependency modelling and simulation for emergency management and decision making by the civil security authorities. In: Rome, E., Theocharidou, M., Wolthusen, S. (eds.) CRITIS 2015. LNCS, vol. 9578, pp. 255–258. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33331-1_23
3. Assante, M., Lee, R.: The Industrial Control System Cyber Kill Chain, White Paper, SANS Institute, Bethesda, Maryland (2015)
4. Bernardini, E., Foglietta, C., Panzieri, S.: Modeling telecommunications infrastructures using the CISIApro 2.0 simulator. In: ICCIP 2020. IAICT, vol. 596, pp. 325–348. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62840-6_16
5. Cyber and Infrastructure Security Centre, CIPMA: Critical Infrastructure, Program for Modeling and Analysis, Australian Department of Home Affairs, Canberra, Australia (2015)
6. Digioia, G., Foglietta, C., Panzieri, S., Falleni, A.: Mixed holistic reductionistic approach for impact assessment of cyber attacks. In: Proceedings of the European Intelligence and Security Informatics Conference, pp. 123–130 (2012)
7. Dudley, R., Golden, D.: The Colonial Pipeline ransomware hackers had a secret weapon: self-promoting cybersecurity firms, ProPublica (24 May 2021)
8. European Cybersecurity Competence Centre and Network, Bucharest, Romania (2023). (cybersecurity-centre.europa.eu/index_en)
9. European Network and Information Security Agency, ENISA Threat Landscape 2022, Heraklion, Greece (2022) (www.enisa.europa.eu/publications/enisa-threat-landscape-2022)
10. European Network and Information Security Agency, ENISA Threat Landscape for Ransomware Attacks, Heraklion, Greece (2022). (www.enisa.europa.eu/publications/enisa-threat-landscape-for-ransomware-attacks)
11. European Parliament and the Council of the European Union, Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 Concerning Measures for a High Common Level of Security of Network and Information Systems Across the Union, Document 32016L1148, Brussels, Belgium (2016)
12. Ficco, M., Choras, M., Kozik, R.: Simulation platform for cyber-security and vulnerability analysis of critical infrastructures. J. Comput. Sci. **22**, 179–186 (2017)
13. Foglietta, C., Panzieri, S.: Resilience in critical infrastructures: the role of modeling and simulation. In: Rosato, V., Di Pietro, A. (eds.) Issues on Risk Analysis for Critical Infrastructure Protection, IntechOpen, London, United Kingdom, pp. 3–18 (2020)
14. Franchina, L., Socal, A.: Innovative predictive model for smart city security risk assessment. In: Proceedings of the Forty-Third International Convention on Information, Communications and Electronic Technology, pp. 1831–1836 (2020)
15. Goodell, J., Corbet, S.: Commodity market exposure to energy-firm distress: evidence from the colonial pipeline ransomware attack. Finance Res. Lett. **51**, 103329 (2023)

16. Hobbs, A.: The Colonial Pipeline hack: Exposing vulnerabilities in U.S. cybersecurity, SAGE Business Cases (6 July 2021)
17. International Electrotechnical Commission, IEC 62443 Series - Industrial Communication Networks - Network and System Security, Geneva, Switzerland, 2009–2023
18. Katagiri, N.: Hackers of critical infrastructure: expectations and limits of the principle of target distinction. Inter. Rev. Law Comput. Technol. article no. 2164462 (2023)
19. Knowles, W., Prince, D., Hutchison, D., Pagna Disso, J., Jones, K.: A survey of cyber security management in industrial control systems. Inter. J. Critical Infrastructure Protect. **9**, 52–80 (2015)
20. Miclea, L., Sanislav, T.: About dependability in cyber-physical systems. In: Proceedings of the Ninth East-West Design and Test Symposium, pp. 17–21 (2011)
21. National Cybersecurity Agency, National Cybersecurity Strategy 2022 – 2026, Rome, Italy. (2022) (www.acn.gov.it/ACN_EN_Strategia.pdf)
22. Oliva, G., Panzieri, S., Setola, R.: Modeling and simulation of critical infrastructures. WIT Trans. State-of-the-Art Sci. Eng. **54**, 39–56 (2012)
23. Pathirana, A.: EPANET2 desktop application for pressure-driven demand modeling. In: Proceedings of the Twelfth Annual Conference on Water Distribution System Analysis, pp. 65–74 (2010)
24. Republic of Italy, Legislative Decree of May 18, 2018, no. 65 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, L.D. no. 65/2018, Rome, Italy (2018). (www.gazzettaufficiale.it/eli/id/2018/06/09/18G00092/sg)
25. Republic of Italy, Law Decree of September 21, 2019, no. 105 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, L.D. no. 105/2019, Rome, Italy (2019). (www.gazzettaufficiale.it/eli/id/2019/09/21/19G00111/sg)
26. Republic of Italy, Decree of the President and the Council of Ministers of July 30, 2020, no. 131 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, DPCM no. 131/2020, Rome, Italy (2020). (www.gazzettaufficiale.it/eli/id/2020/10/21/20G00150/sg)
27. Republic of Italy, Decree of the President of the Republic of February 5, 2021, no. 54 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, DPR no. 54/2021, Rome, Italy (2021). (www.gazzettaufficiale.it/eli/id/2021/04/23/21G00060/sg)
28. Republic of Italy, Decree of the President and the Council of Ministers of April 14, 2021, no. 81 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, DPCM no. 81/2021, Rome, Italy (2021). (www.gazzettaufficiale.it/eli/id/2021/06/11/21G00089/sg)
29. Republic of Italy, Legal Decree of June 14, 2021, no. 82 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, L.D. no. 82/2021, Rome, Italy (2021). (www.gazzettaufficiale.it/eli/id/2021/06/14/21G00098/sg)
30. Republic of Italy, Decree of the President and the Council of Ministers of June 15, 2021, no. 198 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, DPCM no. 198/2021, Rome, Italy (2021). (www.gazzettaufficiale.it/eli/id/2021/08/19/21A05087/sg)
31. Republic of Italy, Decree of the President and the Council of Ministers of May 18, 2022, no. 92 (in Italian), *Gazzeta Ufficiale della Repubblica Italiana*, DPCM no. 92/2022, Rome, Italy (2022). (www.gazzettaufficiale.it/eli/id/2022/07/15/22G00099/sg)
32. Williams, T.: The Purdue enterprise reference architecture. Comput. Ind. **24**(2–3), 141–158 (1994)

Security-Enhanced Orchestration Platform for Building Management Systems

Raymond Chan[1](✉), Wye Kaye Yan[1], Jung Man Ma[1], Kai Mun Loh[2],
Tan Yu[1], Malcolm Low[1], Habib Rehman[3], and Thong Chee Phua[3]

[1] Singapore Institute of Technology, Singapore, Singapore
Raymond.Chan@singaporetech.edu.sg
[2] University of Glasgow, Glasgow, UK
[3] Firefense, Singapore, Singapore

Abstract. A building management system is an infrastructure asset that operates critical building components such as water supply management, electric power monitoring and heating, ventilation and air conditioning systems. Internet of Things devices are increasingly employed in building management systems for efficient operations. The Message Queuing Telemetry Transport protocol is commonly used for communications when integrating these devices. However, each device is typically isolated and has its own platform and management dashboard. The isolation and heterogeneity hinder device visibility and render it challenging to monitor and respond to abnormal conditions, including those induced by cyber attacks.

This chapter describes a security-enhanced orchestration platform for building management systems. The orchestration platform receives a variety of data from building systems and Internet of Things devices to provide situation awareness and support efficient operation. The integration of novel device auto-recovery and auto-isolation functionality in the orchestration platform enables the monitoring and mitigation of cyber attacks.

Keywords: Building Management Systems · Internet of Things Devices · Security-Enhanced Orchestration Platform

1 Introduction

A building management system incorporates several industrial control systems that manage critical electric power control, water and gas supply, elevator operation, access control and fire alarming and suppression systems. However, the various systems, which are tied to specific products and services, are often deployed separately and are isolated from each other because they are installed and managed by different providers. In fact, most providers do not recommend connecting their systems to other systems for latency and performance reasons [10].

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 164–176, 2024.
https://doi.org/10.1007/978-3-031-49585-4_8

The use of Internet of Things (IoT) devices and sensors in building management systems has increased in recent years, making buildings smarter and more efficient [13]. However, the devices and sensors increase the cyber attack surfaces and render buildings more vulnerable to cyber attacks. An adversary who compromises a building access control system can gain entrance and steal items or destroy property without breaking physical locks. Closed-circuit television (CCTV) systems can also be compromised surreptitiously to perform malicious acts.

To address these and other security issues, orchestration platforms are required for building management systems to perform monitoring, control and threat and anomaly detection and response. An orchestration platform integrates diverse operational technology (OT) and information technology (IT) systems along with Internet of Things devices and sensors to facilitate efficient and secure building operation.

This chapter describes a security-enhanced orchestration platform for building management systems. The platform receives data from diverse building management system components and Internet of Things devices to provide situation awareness and support efficient and stable building operation. The integration of novel device auto-recovery and auto-isolation functionality enables the orchestration platform to monitor cyber attacks and mitigate their negative effects.

2 Related Work

In recent years, the building management system industry has been exploring the possibility of integrating legacy systems and devices in a single platform that collects building sensor data and efficiently controls the various building systems.

Agarwal et al. [1] have developed BuildingDepot, an extensible and distributed architecture for building data storage, access and sharing. The architecture leverages the representational state transfer application programming interface (REST API) to access sensor networks in a building. Their subsequent BuildingDepot 2.0 platform [14] provides advanced data analysis and supervisory control features. It enables reusable applications to be employed in different building environments and provides a template that describes sensors and building systems in a common language.

Alsuhli and Khattab [2] have proposed an Internet of Things architecture for building management that controls lighting and heating, ventilation and air conditioning systems. They proceeded to implement a prototype system and evaluate its accuracy and efficiency.

Due to their vital building monitoring and control functionality, it is important to secure building management systems from cyber attacks. Fisk [6] notes that legacy systems elevate risk because they have limited computing power and many known vulnerabilities that are still unpatched. Chan et al. [4] have identified vulnerabilities in smart lighting systems and energy metering systems. The increased use of Internet of Things devices with legacy systems expands the attack surface of building management systems. Brooks [3] has investigated current and emerging security vulnerabilities in automated building systems. The results reveal that using wireless devices in open architectures with extended system communications significantly elevates the cyber risk to building management systems.

Much of the research literature has focused on capturing and sharing sensor data from diverse building systems. However, due to the increased risks posed by legacy systems and the integration of Internet of Things devices in a building management system, a security platform must be implemented to monitor real-time data and ensure system integrity. Kalaska and Czarnul [9] conducted a security evaluation of available Internet of Things platforms covering device authorization, data filtering, access control and protecting against service threats. According to the evaluation, few platforms support device authorization and limited protection is provided against network attacks, especially denial-of-service (DoS) attacks. Unfortunately, existing building management system platforms do not detect cyber attacks by analyzing the data they receive. To provide full visibility of building systems and protect them from malicious actors, it is necessary to integrate devices and systems with diverse building management system protocols in a single platform.

To address the security gap, this research has developed a proof-of-concept security-enhanced orchestration platform for building management systems. The platform supports multiple common communications protocols, including Modbus TCP [7], BACnet [5], REST API, MQTT [8], Zigbee [11], Wi-Fi and Bluetooth. Protocol traffic in the building systems and Internet of Things devices is analyzed to detect and mitigate cyber attacks.

3 Orchestration Platform

The security-enhanced orchestration platform for building management systems is designed to manage building system applications as well as security and other critical systems. The overall architecture comprises a system architecture and a security architecture. The system architecture incorporates a central control unit (orchestration platform) that integrates multiple systems in an existing building management system with Internet of Things devices to provide seamless user experience while ensuring stability and reliability. The central control unit implements a security architecture with mechanisms for monitoring and controlling access, detecting intrusions and isolating compromised systems and devices. The orchestration platform provides a robust solution for monitoring and controlling building systems while maintaining security.

3.1 System Architecture

Figure 1 shows the system architecture of the security-enhanced orchestration platform for building management systems. The system architecture comprises two components, selected systems (Systems 1 to 4) and devices in an existing

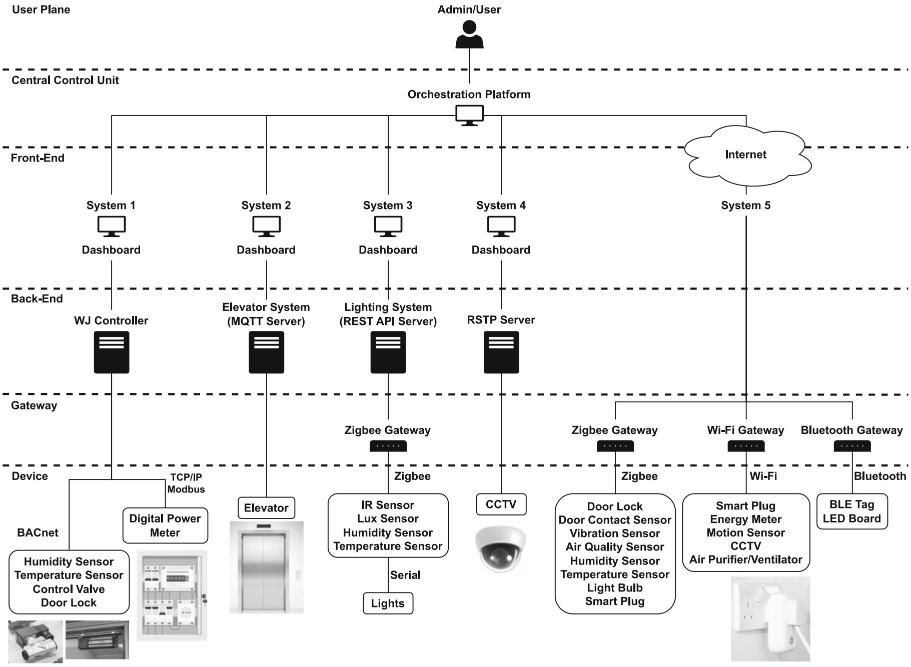


Fig. 1. System architecture.

building management system, and a central control unit with Internet of Things devices (System 5). The selected systems and devices in the building management system monitor and control various building applications. The devices are categorized as operational technology and include equipment such as lighting and elevator control systems.

The central control unit with Internet of Things devices is incorporated in the system architecture to integrate existing systems and devices while providing remote management capabilities. Internet of Things devices are smart systems and sensors that support and enhance building management system functionality. The devices include humidity sensors, temperature sensors, vibration sensors and air quality sensors, among others. Table 1 provides information about the devices incorporated in the system architecture.

3.2 Security Architecture

Building management system architectures typically isolate subsystems that support specific applications, often requiring different back-ends and/or front-ends for subsystem operation. However, this architecture introduces latency and performance problems when the subsystems are interconnected. The isolation significantly complicates building management governance by operators due to the reduced visibility of system state. The orchestration platform is intended to

Table 1. Device information.

System	Device	Protocol	Class
System 1 (WJ Controller)	Humidity/temperature sensors	BACnet	OT
	Control valve	BACnet	OT
	Door lock	BACnet	OT
	Digital power meter	Modbus TCP	OT
System 2 (MQTT Server)	Elevator	MQTT	OT
System 3 (REST API Server)	Infrared sensor	ZigBee	IoT
	Lux sensor	ZigBee	IoT
	Humidity/temperature sensors	ZigBee	IoT
	Light bulb	Serial	OT
System 4 (RSTP Server)	Closed-circuit TV	MQTT	OT
System 5 (IoT Devices)	Door lock	ZigBee	IoT
	Door contact sensor	ZigBee	IoT
	Vibration sensor	ZigBee	IoT
	Air quality sensor	ZigBee	IoT
	Humidity sensor	ZigBee	IoT
	Temperature sensor	ZigBee	IoT
	Light bulb	ZigBee	IoT
	Smart plug	ZigBee/Wi-Fi	IoT
	Energy meter	Wi-Fi	IoT
	Motion sensor	Wi-Fi	IoT
	Closed-circuit TV	Wi-Fi	IoT
	Air purifier	Wi-Fi	IoT

address these deficiencies by serving as the central governor of a building management system, providing key capabilities such as monitoring, control, data collection, storage and analysis, as well as physical security and cyber security.

A key advantage of the orchestration platform is its ability to overcome the limitations of traditional isolated systems. It is difficult to detect attacks and abnormalities in typical building management systems due the limited visibility they provide. However, incorporating Internet of Things devices in building management systems along with an orchestration platform with intrusion detection capabilities provides improved visibility of system performance as well as efficient governance and secure building operation.

Figure 2 shows the security-enhanced orchestration framework. It incorporates a front-end alert and notification system that displays attack alerts and abnormal reading notifications in real time. This enables building management operators to quickly identify and respond to potential security incidents. The back-end of the framework provides several key functions that ensure overall stability and security. The system health check function monitors CPU and mem-

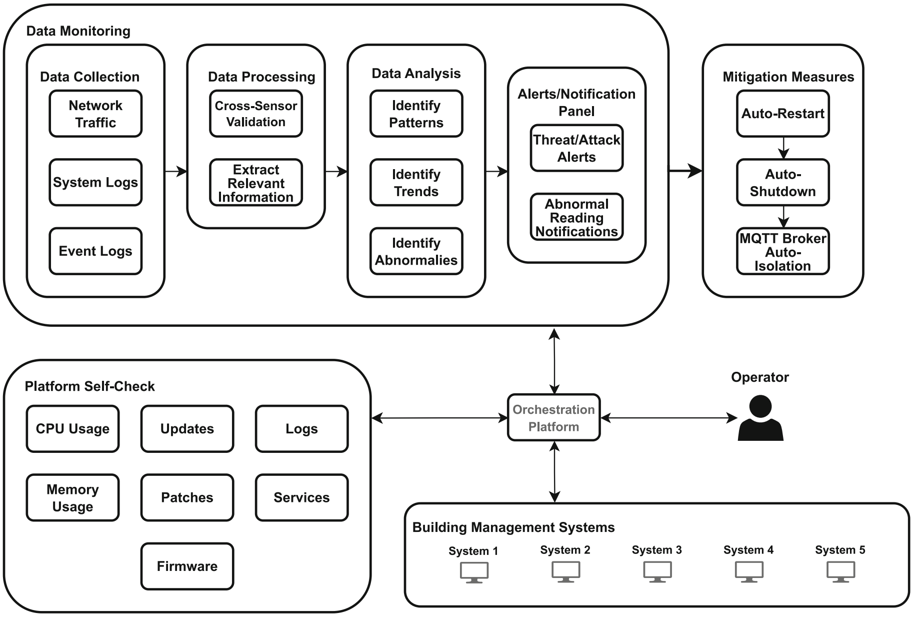


Fig. 2. Security-enhanced orchestration framework.

ory usage, updates, patches, firmware, network traffic and services to ensure that the system is operating at optimal conditions. Additionally, the framework provides monitoring and identification functionality that leverage data collection, processing and analysis to detect potential security threats and attacks.

The framework also provides response measures to mitigate the negative effects of incidents. This includes device auto-restart, device auto-shutdown and device auto-isolation. Compromised devices and networks are rapidly isolated and shut down to prevent security incidents from spreading and minimize their impacts on building management. The framework provides a comprehensive and robust solution that enables operators to rapidly identify and respond to potential security incidents and maintain overall building stability and security.

4 Device Auto-Recovery and Auto-Isolation

Incorporating Internet of Things devices in building management systems introduces new cyber security risks because it expands the attack surfaces for malicious actors. Nevertheless, the devices are vital because it is difficult to provide security services without visibility into building systems and operations. The orchestration platform provides full visibility into all aspects of the building management system, including all the connected devices.

The orchestration platform provides device auto-recovery and device auto-isolation functionality. Figure 3 shows the workflow geared for restartable and

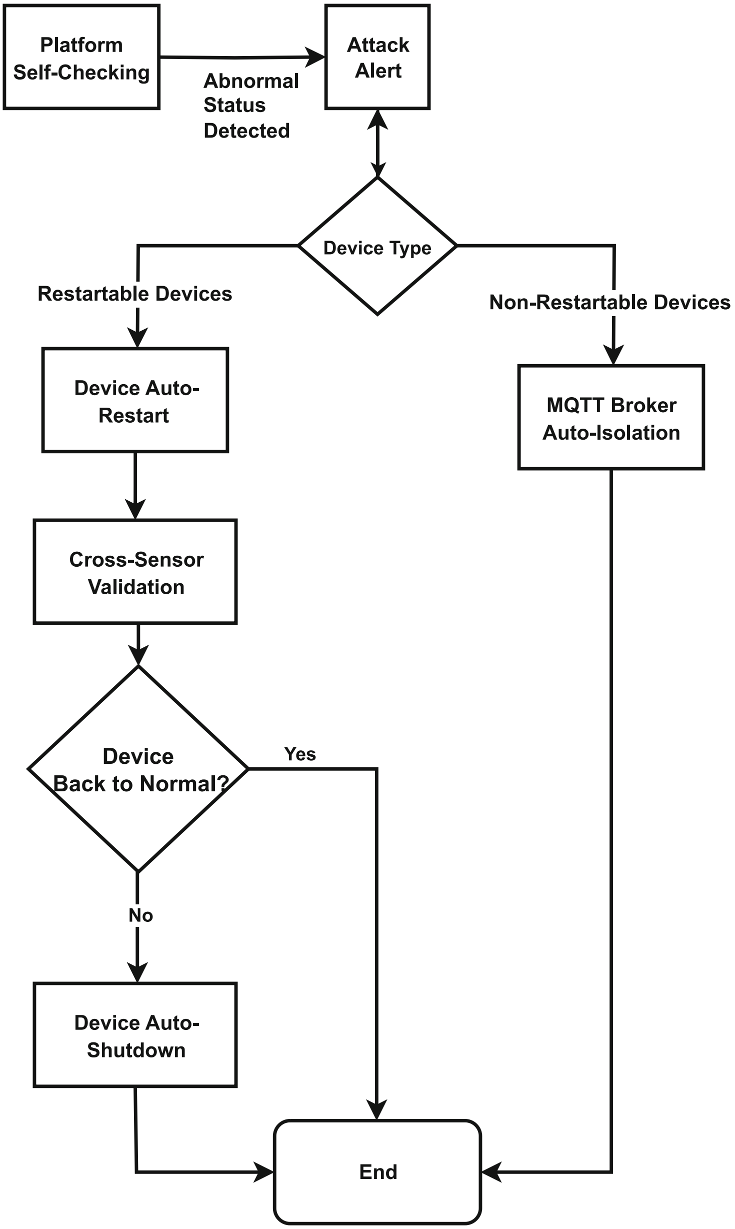


Fig. 3. Device auto-recovery and auto-isolation workflow.

non-restartable devices. Restartable devices are not critical to building management system operation and can be safely restarted by the orchestration platform after attacks or incidents. Non-restartable devices such as those installed in important elevator and door lock systems require manual intervention because auto-restarts could negatively impact building operation. The purpose of the workflow is to bring the devices and systems back to normal operation.

Self-checking and Abnormal Device Identification. The monitoring function of the orchestration platform periodically checks system parameters such as CPU and memory usage as well as network traffic. This helps track the stability and health of the orchestration platform and the connected devices.

Additionally, the orchestration platform monitors the message-sending frequencies of connected devices. Deviations from the average values are considered to be anomalous and appropriate actions are instituted to investigate and address the anomalies.

Cross-Sensor Validation. One of the key features of the orchestration platform is its ability to perform cross-sensor validation. The process is designed to ensure that the readings received from connected sensors are accurate and not compromised by a malicious actor.

The first step in cross-sensor validation is to identify the sensors that are sending abnormal readings. This is done by comparing the sensor readings against historical data. Following this, data from relevant sensors is used to validate whether or not other sensors measuring the same or similar data have similar patterns as the sensors with abnormal readings. If most of the other sensors have different readings, it is concluded that the sensors with abnormal readings have issues that are more serious than simply anomalous. In such a situation, the orchestration platform proceeds to the device auto-restart phase.

Device Auto-Restart. In many situations, restarting a device is an effective way of resolving problems and bringing the device back to the normal state. This is especially true for Internet of Things devices due to their diverse hardware, software and protocols. Unfortunately, manual device restarts can be problematic and often require time-consuming human operator intervention. The orchestration platform is specifically designed to automate device restart when abnormal status is detected.

An affected device is automatically restarted after abnormal status is detected. After the restart, the orchestration platform reconnects to the device and performs cross-sensor validation to ensure that the device is operating normally. The cross-sensor validation process compares the device readings against those of other relevant sensors to ensure that the device is functioning correctly.

By automating the device restart process, the orchestration platform can quickly resolve problems and minimize disruptions to building operation. This important feature provides operators with real-time monitoring and rapid device recovery capabilities.

Device Auto-Shutdown. In the event an Internet of Things device is unable to resume normal operation after a restart, the orchestration platform takes additional measures to maintain the overall stability and security of the building management system. Specifically, the platform disconnects and shuts down the affected device to prevent it from affecting other devices and systems. This action is taken as a precautionary measure to protect the integrity of the building management system. The orchestration platform also notifies operators that the device should be repaired or replaced.

Device Auto-Isolation. Unlike Internet of Things devices, certain building management devices and systems may be deemed critical and cannot be restarted or shut down by the orchestration platform without disrupting building operations. In such cases, the orchestration platform takes a different approach to maintain the overall stability and security of the building management system.

When abnormal status is detected in a critical device or system, the orchestration platform automatically disconnects and isolates the component from the network infrastructure. This prevents the abnormality from affecting other normal devices and systems and the cyber attack from propagating. Additionally, the orchestration platform also automatically blacklists the IP address of the device or system to block further communications with the building management system and prevent malicious traffic from entering.

The orchestration platform notifies operators that the abnormality should be investigated. Steps are then taken to repair or replace the device or system and return it to normal operation.

Disconnecting, isolating and blacklisting IP addresses also apply to devices that cannot be restarted or shut down. This ensures that the building management system is always operating at optimal conditions and protects against cyber attacks.

5 Experiments

Experiments were conducted to demonstrate the security features of the orchestration platform for building management. Figure 4 shows the experimental setup. A Mosquitto MQTT broker [8] was deployed on the orchestration platform and connected to several devices and systems. Many devices using the MQTT communications protocol do not provide adequate security measures and often rely on insecure default configurations [12]. Therefore, a username-password authentication mechanism was employed to set up the MQTT broker configuration in the experiments. As seen in the figure, the MQTT broker was connected to multiple Internet of Things devices and a Raspberry Pi computer that served as the malicious attack device. A smart plug connected to the Raspberry Pi was employed to shut down and restart the device during the experiments.

Three experimental scenarios were investigated. The first scenario focused on the device auto-restart security feature. The second scenario focused on the

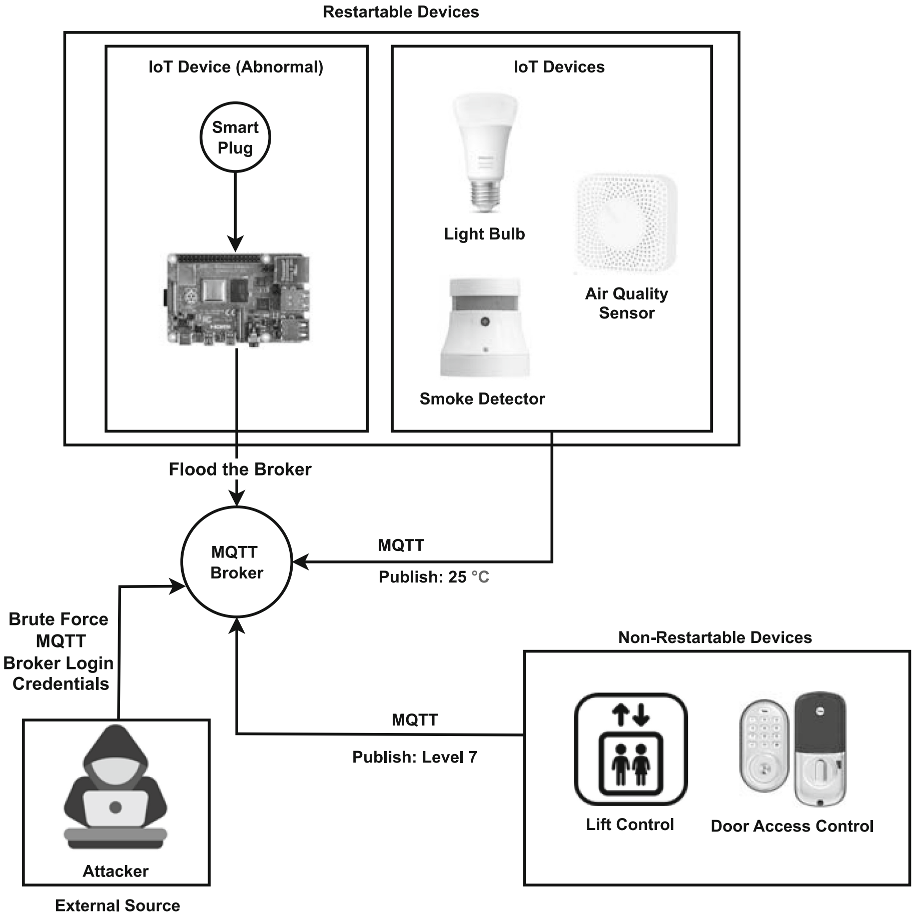


Fig. 4. Experimental setup.

device auto-restart and device auto-shutdown security features. The third scenario focused on the device auto-isolation security feature when a device cannot be restarted or shut down due to its criticality when the building management system is under attack.

5.1 Scenario 1

Scenario 1 involved a situation where a legitimate device connected to the MQTT broker malfunctions or where an attacker gains control of a legitimate device and floods the MQTT broker with messages. Upon receiving an alert, the orchestration platform is designed to engage its device auto-restart security feature. After restarting the device, the orchestration platform performs cross-sensor validation to ensure that the device is functioning normally and does not continue to flood the broker.

The Raspberry Pi was set up as the abnormal device connected to the MQTT broker. A Python script on the Raspberry Pi established a connection with the

MQTT broker and flooded it with 10,000 messages per second for 15 min. Attack alerts were triggered and the orchestration platform leveraged the smart plug to restart the connected Raspberry Pi. After restarting the Raspberry Pi, the orchestration platform performed cross-sensor validation to ensure that the device was working normally. The security feature successfully prevented the abnormal device from flooding the MQTT broker and overloading the server CPU.

5.2 Scenario 2

Scenario 2 involved a situation where a device upon restarting continues to flood the MQTT broker with messages. The orchestration platform is designed to detect and mitigate the attack by automatically shutting down the device after cross-sensor validation.

The Raspberry Pi was set up to run the Python script that simulated an abnormal device flooding the MQTT broker with messages after it was auto-restarted. The orchestration platform detected the flooding attack by monitoring the MQTT broker and identified the device based on the attack alert. After the Raspberry Pi was restarted, cross-sensor validation was performed, but the device was found to be abnormal and was, therefore, shut down. This scenario demonstrates the effectiveness of the orchestration platform at mitigating message flooding attacks and ensuring the stability of the MQTT broker.

5.3 Scenario 3

Scenario 3 was designed to evaluate the effectiveness of the MQTT broker auto-isolation security feature. A brute force attack was launched to gain unauthorized access to the MQTT broker by repeatedly trying different login credentials in a short time frame. Specifically, ten connection attempts were made in 20 s from an external source.

Upon receiving the attack alert, the MQTT broker auto-isolation feature captured the username and removed it from the broker's access control list, effectively isolating the attack by preventing a connection to the MQTT broker. The security feature ensures that, even if access is gained to the MQTT broker via a brute force attack, the compromised username and password cannot be used to connect to the MQTT broker or access any MQTT topics.

It should be noted that the device auto-restart and auto-shutdown security features do not apply to critical building devices and systems such as the lift control and door control systems. Instead, the MQTT broker auto-isolation security feature is leveraged to protect these critical systems from external attacks.

5.4 Results and Limitations

Table 2 shows the average recovery times for the security features triggered during the three scenarios. The recovery times were measured by determining the time elapsed from when an attack alert was received to when the device was returned to its normal state or when isolation was successfully activated by removing the username from the MQTT broker access control list. The average

Table 2. Experimental results.

Scenarios	Average Recovery Time	Triggered Security Feature
1	15.26 s	Device auto-restart
2	55.57 s	Device auto-restart and shutdown
3	24.10 s	MQTT broker auto-isolation

recovery times were computed as the averages over ten executions in each of the three scenarios.

The experimental results involving the MQTT broker demonstrate the effectiveness of the security features implemented in the orchestration platform.

In Scenario 1, the device auto-restart feature successfully mitigated an MQTT message flooding attack by restarting the abnormal device and performing cross-sensor validation to ensure normal operation.

The results in Scenario 2 further show the importance of the device auto-shutdown security feature in preventing repeated MQTT message flooding. The orchestration platform successfully detected the attack, restarted the abnormal device and subsequently shut it down when it determined that the device was not functioning properly.

The results in Scenario 3 involving a brute force (credential stealing) attack demonstrate the effectiveness of the MQTT broker auto-isolation feature in preventing unauthorized broker access and minimizing potential damage.

However, it is important to consider the potential negative side effects of the auto-restart and auto-shutdown features. If triggered too frequently, the two features could disrupt normal device operation, causing user inconvenience and frustration.

6 Conclusions

Increasing numbers of Internet of Things devices are being deployed in infrastructure assets such as buildings. However, it is challenging for building management operators to monitor these heterogeneous devices and troubleshoot them when they malfunction or are targeted by cyber attacks.

The security-enhanced orchestration platform described in this chapter is designed for building management systems that incorporate operational technology and diverse Internet of Things devices. The orchestration platform receives a variety of data from building management components and Internet of Things devices to provide situation awareness and support efficient and stable operation. The integration of novel device auto-recovery and auto-isolation functionality in the orchestration platform enables the monitoring and mitigation of abnormal conditions, including those initiated by cyber attacks.

Future research will attempt to apply machine learning techniques to enhance the detection of abnormal Internet of Things device operations. Additionally, it will augment the orchestration platform to monitor and manage additional building components such as gas and water supply systems.

References

1. Agarwal, Y., Gupta, R., Komaki, D., Weng, T.: BuildingDepot: an extensible and distributed architecture for building data storage, access and sharing. In: Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pp. 64–71 (2012)
2. Alsuhli, G., Khattab, A.: A fog-based IoT platform for smart buildings. In: Proceedings of the International Conference on Innovative Trends in Computer Engineering, pp. 174–179 (2019)
3. Brooks, D.: Intelligent buildings: an investigation into current and emerging security vulnerabilities in automated building systems using an applied defeat methodology. In: Proceedings of the Fourth Australian Security and Intelligence Conference, pp. 16–26 (2011)
4. Chan, R., Tan, F., Teo, U., Kow, B.: Vulnerability assessments of building management systems. In: ICCIP 2020. IAICT, vol. 596, pp. 209–220. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62840-6_10
5. Fernbach, A., Granzer, W., Kastner, W.: Interoperability at the management level of building automation systems: a case study for BACnet and OPC UA. In: Proceedings of the International Conference on Emerging Technologies and Factory Automation (2011)
6. Fisk, D.: Cyber security, building automation and the intelligent building. Intell. Buildings Inter. **4**(3), 169–181 (2012)
7. Fovino, I.N., Carcano, A., Masera, M., Trombetta, A.: Design and implementation of a secure modbus protocol. In: Palmer, C., Shenoi, S. (eds.) ICCIP 2009. IAICT, vol. 311, pp. 83–96. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04798-5_6
8. Hillar, G.: MQTT Essentials - A Lightweight IoT Protocol. Packet Publishing, Birmingham, United Kingdom (2017)
9. Kalaska, R., Czarnul, P.: Some security features of selected IoT platforms. TASK Q. **24**(1), 29–61 (2020)
10. Makonin, S.: App programming and its use in smart buildings. In: Pacheco-Torgal, F., Rasmussen, E., Granqvist, C., Ivanov, V., Kaklauskas, A., Makonin, S. (eds.) Start-Up Creation: The Smart Eco-Efficient Built Environment, Woodhead Publishing, Sawston, United Kingdom, pp. 451–463 (2016)
11. Muthu Ramya, C., Shanmugaraj, M., Prabakaran, R.: Study on ZigBee technology. In: Proceedings of the Third International Conference on Electronics Computer Technology, pp. 297–301 (2011)
12. Palmieri, A., Prem, P., Ranise, S., Morelli, U., Ahmad, T.: MQTTSA: a tool for automatically assisting the secure deployments of MQTT brokers. In: Proceedings of the IEEE World Congress on Services, pp. 47–53 (2019)
13. Rathore, M., Ahmad, A., Paul, A., Rho, S.: Urban planning and building smart cities based on the Internet of Things using big data analytics. Comput. Netw. **101**, 63–80 (2016)
14. Weng, T., Nwokafor, A., Agarwal, Y.: BuildingDepot 2.0: an integrated management system for building analysis and control. In: Proceedings of the Fifth ACM Workshop on Embedded Systems for Energy-Efficient Buildings (2013)

Practical Deep Neural Network Protection for Unmodified Applications in Intel Software Guard Extension Environments

Dee Meng Kang[1](✉), Haadee Faahym[1,2], Souhail Meftah[1,3], Sye Loong Keoh[2], and Mi Mi Aung Khin[1]

[1] Agency for Science, Technology and Research, Singapore, Singapore
Kang_Dee_Meng@i2r.a-star.edu.sg
[2] University of Glasgow Singapore, Singapore, Singapore
[3] National University of Singapore, Singapore, Singapore

Abstract. Trusted computing, often referred to as confidential computing, is an attempt to enhance the trust of modern computer systems through a combination of software and hardware mechanisms. The area increased in popularity after the release of the Intel Software Guard Extensions software development kit, enabling industry actors to create applications compatible with the interfaces required to leverage secure enclaves. However, the prime choices of users are still libraries and solutions that facilitate code portability to Software Guard Extension environments without any modifications to native applications. While these have proved effective at eliminating additional development costs, they inherit all the security concerns for which Software Guard Extensions has been criticized.

This chapter proposes a split computing method to enhance the privacy of deep neural network models outsourced to trusted execution environments. The key metric that guides the approach is split computing performance that does not involve architectural modifications to deep neural network models. The model partitioning method enables stricter security guarantees while producing negligible levels of overhead. This chapter also discusses the challenges involved in developing a pragmatic solution against established Intel Software Guard Extensions attacks. The results demonstrate that the method introduces negligible performance overhead and reliably secures the outsourcing of deep neural network models.

Keywords: Trusted Computing · Intel Software Guard Extensions · Machine Learning

1 Introduction

Machine Learning as a Service (MLaaS) platforms are increasingly deployed by cloud infrastructure providers such as Amazon Web Services and Microsoft Azure

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 177–192, 2024.
https://doi.org/10.1007/978-3-031-49585-4_9

to support remote computations for sensitive decision making and security-critical environments. The use of cloud infrastructure assets expands the attack surfaces of machine learning applications that support critical operations. These include attacks from malicious programs and adversaries that compromise operating systems and hypervisors, posing serious threats to the integrity and privacy of machine learning models.

1.1 Trusted Execution Environment

Trusted execution environments utilize hardware and software protection mechanisms to isolate sensitive code from the remaining portions of applications. They offer practical solutions for enterprises and cloud service providers that support the secure handling of confidential information. Trusted execution environments such as ARM TrustZone and Intel Software Guard Extensions (SGX) are widely used by many processors to provide integrity and privacy guarantees. In the context of outsourced machine language computations, trusted execution environments outperform pure cryptographic implementations by several orders of magnitude [24]. However, the isolation guarantees of trusted execution environments come with the steep price of poor scalability compared with other untrusted alternatives executing in native environments.

1.2 Intel Software Guard Extensions

Intel SGX is a set of hardware enforcement mechanisms designed to provide integrity and confidentiality guarantees to the operating system, kernel, hypervisors and privileged software. It enables user programs to allocate private memory regions called enclaves that isolate application code and data through hardware-based memory encryption. Intel SGX also enables cross-enclave communications via software attestation to verify that an application is running on real hardware in an up-to-date trusted execution environment with the expected initial state.

Nevertheless, Intel SGX has been criticized by the research community for its vulnerabilities to attacks that target page units, segmentation units, CPU caches, dynamic RAM, page tables, branch predictions, enclave interfaces and hardware. Some notable attacks include SGXPectre [1], CacheZoom [13], DRAMA [15] and rowhammer [23]. Intel SGX has also been criticized because its software development kit introduces high development and integration costs, and does not enable native applications to execute out of the box. As a result, efforts have been undertaken to develop libraries that port applications into Intel SGX environments.

2 Background

Intel SGX is computationally expensive due to its design limitations and limited memory. The implementation requires application code to be divided into trusted and untrusted components. Trusted component code accesses the confidential data within the Intel SGX enclave whereas the untrusted component accesses the remaining application data outside the protection of the enclave. This distinction requires major code refactoring to successfully execute natively-developed applications on Intel SGX.

In order for trusted and untrusted components to interact with each other, enclave and outside calls (ecalls and ocalls) must be invoked to interface with the hardware, which causes overhead. Zhao et al. [26] have demonstrated that ecall and ocall cycles per operation are higher than system and function calls. Furthermore, the page swapping mechanism triggered when the available enclave memory is exceeded increases the overhead for each page swap by several hundred thousand CPU cycles. Nevertheless, the security mechanisms offered by Intel SGX enable developers to seek trade-offs between security enhancements and computational costs. Additionally, Intel SGX utilization must consider issues such as discovered vulnerabilities and the development overhead incurred to adjust code to the hardware and the software development kit. Fortunately, porting frameworks such as Gramine-SGX [7] and Mystikos-SGX [4] provide out-of-the-box code integration to Intel SGX, drastically reducing the engineering effort required to deploy applications in trusted execution environments.

2.1 Evaluation Setup

The evaluation setup employed in the research comprised a Microsoft Azure Standard DC4s v2 machine with four virtual Intel Xeon E-2288G 3.70 GHz CPUs, 200 GiB storage and 16 GiB of memory. The machine executed Ubuntu 20.04 LTS (Linux Version 5.13.0-1017-Azure). All the Intel SGX frameworks were allocated 8 GB of trusted memory for the implementation to utilize and execute machine learning model inference.

2.2 Gramine-SGX

Gramine-SGX is a lightweight guest operating system designed to execute applications in isolated environments with benefits that include ease of porting and process migration with minimal host requirements. It comprises the library operating system and a shared library named `shim` in the source code. Additionally, it includes the platform adaption layer and GNU C Library, a set of shared libraries, that initializes upon loading the Intel SGX enclave.

Each application requires a manifest file, a metadata file containing information about the resources and required environment for executing a Gramine-SGX application [7]. Gramine-SGX includes a framework for developing privacy-preserving machine learning applications. The framework enables machine learning model training and inference workloads to execute in third-party environments while providing integrity and confidentiality guarantees to the models and inputs.

This research employed the PyTorch machine learning framework. The Intel SGX enclave in an untrusted machine isolates the PyTorch runtime environment from attacks that target confidentiality and integrity. It also provides cryptographic attestation to the correct initialization and execution of different enclaves, enabling distributed computations. The workflow of the PyTorch workload in a Gramine-SGX environment is detailed in [6].

This research has benchmarked the machine learning inference performance against several PyTorch deep neural network model variants – Squeezenet [19], MobileNet V3 Small and MobilNet V3 Large [18], ResNet50 and ResNet101 [17], AlexNet [16] and VGG16 and VGG19 [20].

2.3 Mystikos-SGX

Mystikos-SGX is a set of runtime tools for running Linux applications in trusted execution environments. It streamlines the processing of lift-and-shift applications in a containerized Intel SGX trusted execution environment using Docker. Developers have control over the trusted computing base, which enables effective monitoring of all the components involved in program execution [4].

However, proper key management and attestation are out of scope for the particular Mystikos-SGX implementation. In addition, Mystikos-SGX is only compatible with applications developed with the `musl` library. In contrast, Gramine-SGX uses `glibc` as its default C library and also allows `musl` to be mounted.

3 Threat Model

Figure 1 shows the Intel SGX threat model. The Intel Enhanced Privacy ID (EPID) cloud server used to attest EPID keys from the server is outside the scope of this research as are attacks originating from remote clients. Attacking applications running on Intel SGX enclaves by breaking their isolation and confidentiality are considered to be more important by the research community [3].

Fei et al. [5] specify a taxonomy of Intel SGX security vulnerabilities derived by capitalizing on risky channels that can be compromised to initiate attacks against Intel SGX security. These include address translation, CPU cache, dynamic RAM, branch prediction, and enclave software and hardware vulnerabilities. Mainstream attacks on Intel SGX are geared towards successfully executing cache side-channel attacks that generally exploit CPU cache, dynamic RAM and branch prediction vulnerabilities.

Intel [8] has determined that providing defensive measures against side-channel attacks are beyond its scope. Therefore, it is up to developers to devise security mechanisms against the attacks. In a standard CPU, each physical core has exclusive access to the L1 and L2 caches while time-sharing other levels of cache with the remaining CPU cores. Under the assumption that all software running in an Intel SGX stack shares access to the same memory cache, an adversary can exploit side-channels such as the time difference between cache accesses.

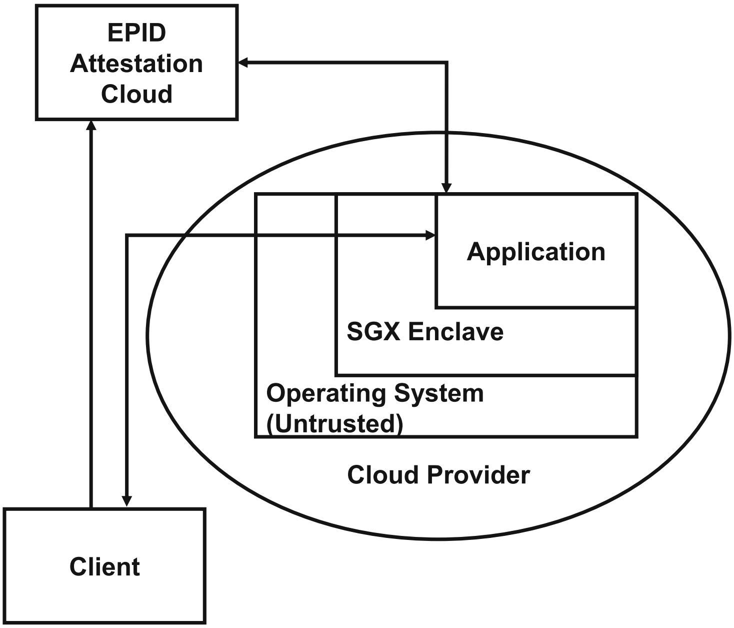


Fig. 1. Intel SGX threat model.

Prominent timing-channel attacks on the memory cache include three main variants, Evict+Reload [22], Prime+Probe and Flush+Reload [25]. These variants are fundamental to more advanced side-channel attacks like the SGXPectre attack [1]. The speculative execution threads of Intel SGX can be exploited by the SGXPectre attack that subverts the confidentiality of SGX enclaves. The control flow of an SGX enclave as well as its branch prediction can be compromised to enable cache state changes to be measured and confidential information about the machine learning model and inputs to be extracted. Furthermore, SGXPectre can steal encryption keys and attestation keys from enclaves that could jeopardize entire projects. The effectiveness of the attack has been demonstrated on the SGX software development kit.

The Large-Scale Data and Systems Group at Imperial College London [12] has demonstrated a conceptual branch prediction Intel SGX attack that was inspired by the Meltdown attack on Intel SGX [21]. The enclave application reads an input from outside the enclave by invoking a function. However, before the application can invoke the function, the attack flushes the cache line using the `clflush` instruction to force the application to load the input that resides in the cache [21]. The conceptual attack is only feasible on the SGX software development kit framework. It cannot be implemented on the Gramine-SGX framework although it shares the same library vulnerability.

The Intel SGX attacks mentioned above have minimal feasibility, but mitigation methods to prevent them from successfully using confidential applications are crucial. In this research, the mitigations would have to combat attempts at extracting a machine learning model residing in an Intel SGX enclave. These would guarantee the confidentiality of the machine learning model and ensure that is not used by untrusted parties.

4 Split Computing Model for Security

Split computing without architectural modifications to deep neural network models has been studied for image classification tasks [9], speech recognition [11], object detection [2,10] and sentiment analysis. Narra et al. [14] have employed Origami split computing to ensure privacy-preserving inference while also improving performance. The approach splits a machine learning model into multiple partitions and encrypts the first partition inside an Intel SGX enclave. It then sends the encrypted output to an untrusted environment for computation using a GPU. The de-blinding factors are kept private by the enclave and only decrypted after the untrusted computations have been completed. However, an adversary could still access layers that are not computed in the Intel SGX enclave, thereby compromising its confidentiality.

As the name suggests, split computing is a model partitioning method that enables the independent execution of certain layers of a deep neural network model in a pipelined manner to produce the same inference results without any increase in model complexity. The technique has been proven to be especially useful in collaborative edge computing, where mobile devices with limited computing power can execute portions of a machine learning model collaboratively with a server. However, at this time, there is no mention in the research literature of this technique being leveraged for security objectives.

All the deep neural network models considered in this work were faithfully implemented from their descriptions in the research literature without any notable modifications.

The first step in the approach is to split a deep neural network model in a manner that maximizes the number of partitions. Figure 2 illustrates how the AlexNet architecture for image classification is split using a few images from the ImageNet dataset for inference. The deep neural network variants employed in this research are compatible with this splitting approach in which a flatten layer is always inserted after a two-dimensional adaptive pooling layer. The flatten layer is needed to support sub-model inferences without having to completely reshape the existing model layers. The number of submodels that could be split depends on the number of iterable layers. In the case of an AlexNet PyTorch model, the maximum number of submodels that could be extracted via splitting is 22.

Model splitting is guided by the maximum number of possible combinations that an adversary could encounter when using a brute-force attack. Table 1 shows the increase in complexity due to model splitting. Specifically, the number of combinations yielded by model splitting is the factorial of the number of models/submodels.

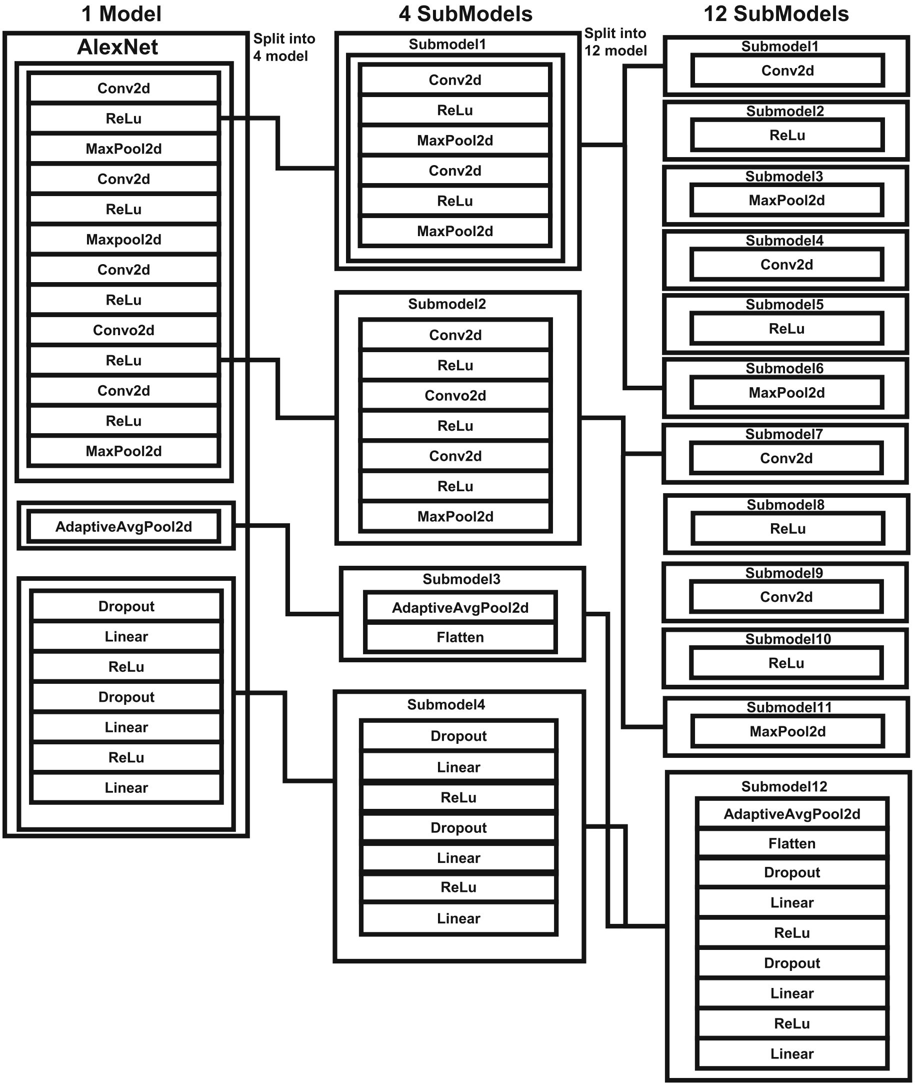


Fig. 2. Breakdown of the split computing method for AlexNet.

Table 2 shows the total inference times required by various deep neural network models without model splitting and with model splitting to 12 submodels. The inference times provide insights into the optimal number of submodels to achieve the desired complexity.

Specifically, in the case of the AlexNet model, the time required for a single inference with one model in an Intel SGX enclave is 2.028153 s (Table 2). Splitting the model into 12 submodels does not affect the runtime, but it increases the total possible model reconstruction combinations to 479,001,600 (Table 1).

Table 1. Possible combinations based on the number of submodel splits.

Models/ Submodels	Combinations Combinations
1	1! = 1
2	2! = 2
4	4! = 24
8	8! = 40,320
10	10! = 3,628,800
12	12! = 479,001,600

Table 2. Inference time increase due to submodel reassembly.

Model	One Model Inference Time	Twelve Submodels Inference Time
Squeezenet	0.226625 s	3.442 yrs
Mobilenet V3 Small	0.163645 s	2.485 yrs
Mobilenet V3 Large	0.331020 s	5.027 yrs
ResNet50	1.230008 s	18.682 yrs
ResNet101	2.044985 s	31.061 yrs
AlexNet	2.028153 s	30.805 yrs
VGG16	4.991928 s	75.822 yrs
VGG19	5.113581 s	77.670 yrs

An adversary running an inference on every possible combination to deduce the correct model would require 30.805 years assuming comparable computing resources (Table 2). Indeed, due to the exponential growth of the possible combinations caused by model splitting, it is advantageous to split a deep neural network model to the maximum number of submodels possible.

The next step is to encrypt each submodel with a unique AES secret key to prevent the adversary from inspecting the raw data. The AES encryption employed a 32-byte key with the cipher-block chaining (CBC) mode. The CBC mode enhances machine learning model security by having different ciphers for identical blocks. This is ideal for deep neural network models that comprise identical nodes in their hidden layers. An AlexNet model has 7 × ReLu activation layers, 5 × Conv2d layers, 3 × MaxPool2d layers, 3 × linear layers and 2 × dropout layers. These interchangeable layers have to be encrypted with different ciphers to further protect the models from being successfully recovered. Fortunately, the overhead incurred when encrypting the submodels with individual AES secret keys is minimal.

Figure 3 shows the memory growth due to encryption for various model splits into submodels. Encrypting the model with splitting incurs memory growth

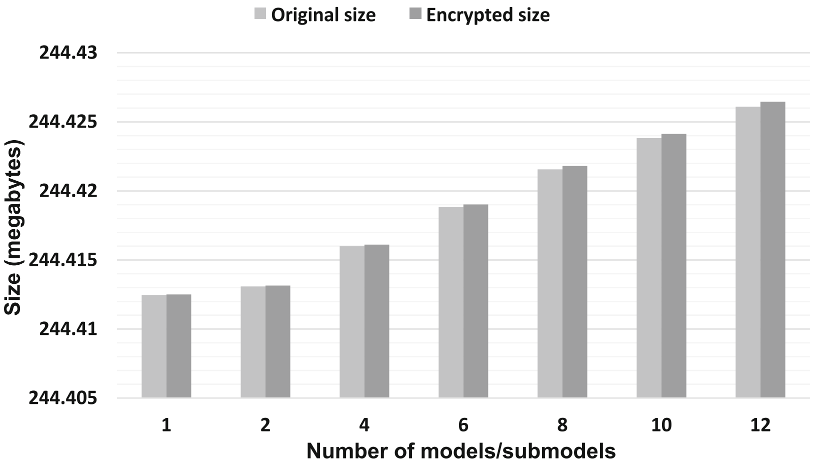


Fig. 3. Memory growth due to encryption for model splits.

from 244.412 MB to 244.426 MB, which is 0.014 MB. Splitting the model into 12 submodels incurs memory growth from 244.412 MB to 244.426 MB, which is 0.014 MB. The memory overhead is negligible and does not cause significant additional loads to the SGX enclave application and its execution.

Next, all the AES secret keys are encoded with a wrapper key generated by Gramine-SGX. The encoded secret keys can only be decoded by a provisioned secret from the Intel SGX quote generator. The encrypted submodels and encoded secret keys are then uploaded to the Intel SGX enclave. In order to decode the encoded secret keys, a user would have to complete an attestation process to ensure that the executing machine is trusted.

5 Remote Attestation via EPID Keys

The remote attestation workflow using EPID keys is provided by the provisioning enclave that requests an EPID key from the Intel provisioning service. The EPID-based remote attestation starts with the enclaved application opening a file to start an SGX report write up. Gramine-SGX employs a hardware instruction that creates a SGX report, which opens up another SGX quote file for reading. Gramine-SGX then uses the quoting enclave to receive the SGX quote. Thereafter, the quoting enclave uses the EPID key provided by the provisioning enclave. The provisioning enclave then requests the EPID key linked to the Intel SGX machine from the Intel provisioning service. The quoting enclave creates the SGX quote from the SGX report and directs it to the enclaved application. The enclaved application then stores the SGX quote in its enclave memory.

To validate the SGX enclave, the enclaved application requests remote attestation and forwards the SGX quote to the trusted Intel SGX machine. A user employs the Intel attestation service by sending the SGX quote to receive an

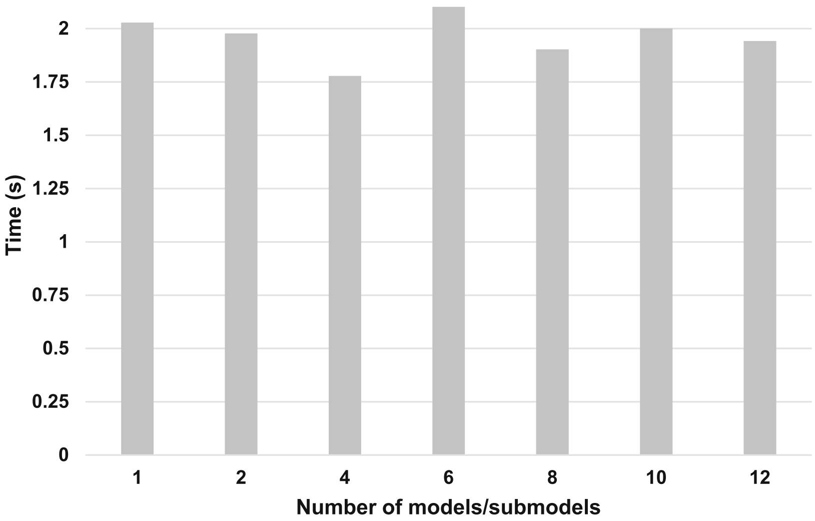


Fig. 4. Average AlexNet inference time in Gramine-SGX.

acknowledgment of the trustworthiness of the Intel SGX machine. Based on the verification procedure, the user can trust the Intel SGX machine and receive the wrapper to decrypt the encoded secret keys [7].

6 Experimental Results and Discussion

Experiments were conducted to evaluate the split computing method as a means to enhance the security of deep neural network models in a trusted execution environment. The experiments employed the Gramine-SGX trusted execution environment, which involved no code modification and provided reduced memory consumption.

The first set of experiments employed the AlexNet deep neural network model to assess the impacts of various submodel splits on inference time, CPU utilization, memory footprint and power consumption in a Gramine-SGX execution environment.

Figure 4 shows that splitting a single AlexNet model all the way up to 12 submodels does not increase or decrease the average inference time significantly. In fact, the average inference time is quite consistent despite the increase in the number of splits.

Figure 5 compares the CPU utilization during AlexNet inference in the Gramine-SGX environment for the single (non-secure) model against the 12-split (secure) model in the Gramine-SGX environment. The two CPU utilization curves track each other with negligible differences.

Figure 6 compares the memory footprints during AlexNet inference in the Gramine-SGX environment for the single (non-secure) model against the 12-split

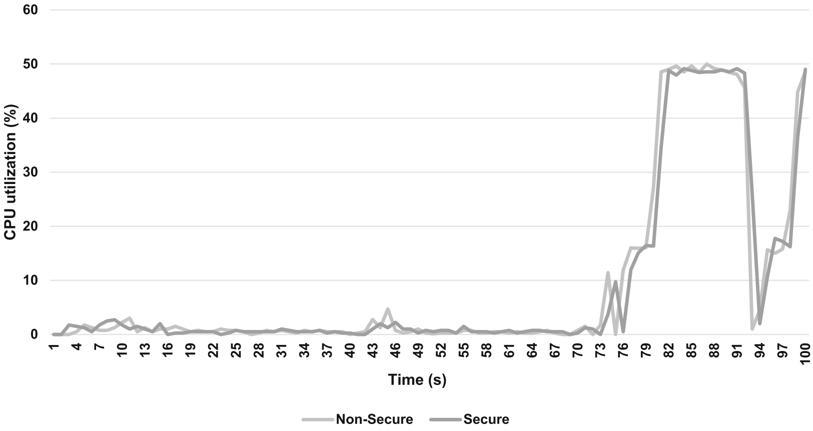


Fig. 5. CPU utilization during AlexNet inference in Gramine-SGX.

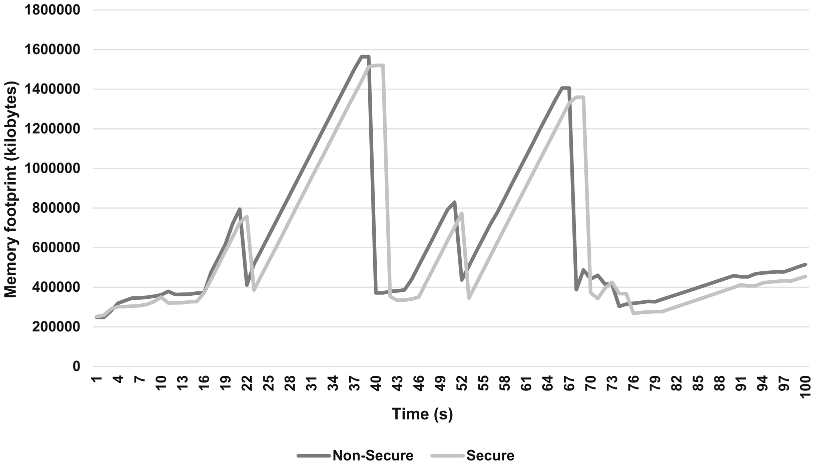


Fig. 6. Memory footprint during AlexNet inference in Gramine-SGX.

(secure) model in the Gramine-SGX environment. The two memory footprint curves are very similar and relatively close to each other.

Figure 7 shows the power consumption during AlexNet inference in the Gramine-SGX environment for a single (non-secure) model and a 12-split (secure) model. The two power consumption curves more or less track each other without significant differences. Overall, the experimental results show that model splitting, while enhancing security, does not introduce significant overhead in terms of time and performance.

Figure 8 compares the average memory footprints in the Gramine-SGX, native and Mystikos-SGX environments. As expected, the native environment has the lowest average memory footprint. However, the Gramine-SGX environ-

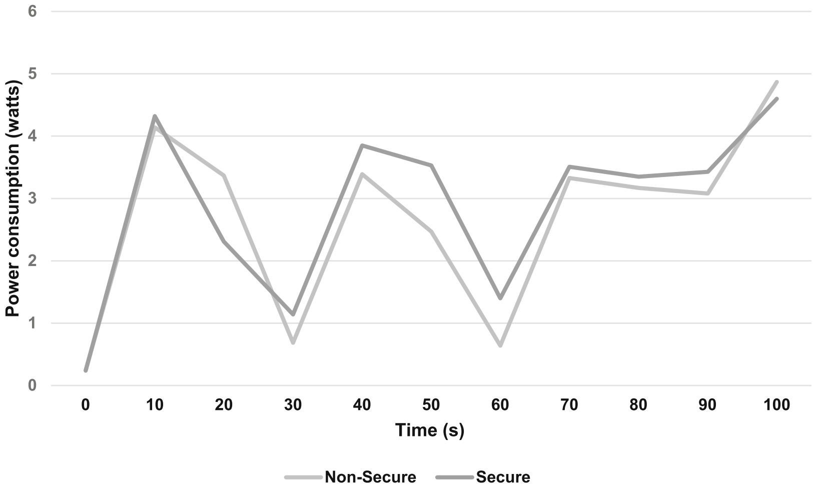


Fig. 7. Power consumption during AlexNet inference in Gramine-SGX.

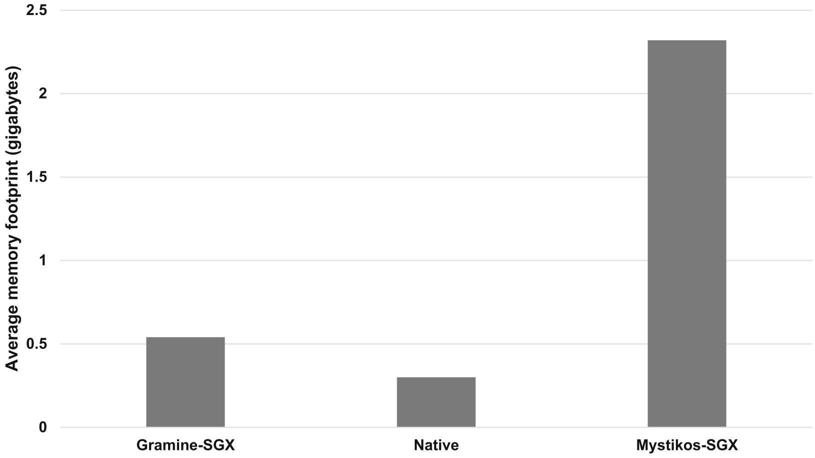


Fig. 8. Average memory footprints in Gramine-SGX, native and Mystikos-SGX.

ment has a footprint that is much closer to the native footprint and significantly lower than the footprint in the Mystikos-SGX environment.

The next set of experiments sought to benchmark the performance times of eight selected deep neural network models during image classification inferencing in the Gramine-SGX environment versus the native environment. The performance time was broken down into inference time, compilation time and total execution time. The inference time was computed as the total execution time minus the compilation time because inference by a deployed deep neural network model does not require any recompilation.

Table 3. Inference performance in native and Gramine-SGX environments.

Model	Native Inference Time	Native Compilation Time	Native Total Execution Time	Gramine-SGX Inference Time	Gramine-SGX Compilation Time	Gramine-SGX Total Execution Time
Squeezenet	0.022 s	0.021 s	0.043 s	0.227 s	75.311 s	75.538 s
Mobilenet V3 Small	0.024 s	0.021 s	0.045 s	0.164 s	76.509 s	76.672 s
Mobilenet V3 Large	0.038 s	0.020 s	0.058 s	0.331 s	77.324 s	77.655 s
ResNet50	0.079 s	0.025 s	0.104 s	1.230 s	77.435 s	78.665 s
ResNet101	0.140 s	0.024 s	0.164 s	2.045 s	78.960 s	81.005 s
AlexNet	0.058 s	0.035 s	0.093 s	2.028 s	82.125 s	84.1535 s
VGG16	0.185 s	0.057 s	0.242 s	4.992 s	84.031 s	89.023 s
VGG19	0.206 s	0.025 s	0.231 s	5.114 s	83.961 s	89.074 s

Table 4. Inference performance in Mystiko-SGX.

Model	Mystiko-SGX Inference Time	Mystiko-SGX Compilation Time	Mystiko-SGX Total Execution Time
Squeezenet	0.517 s	234.154 s	295.096 s
MN V3 Small	0.424 s	237.655 s	293.790 s
MN V3 Large	0.814 s	233.927 s	313.108 s
ResNet50	1.934 s	245.885 s	308.637 s
ResNet101	2.856 s	258.251 s	322.315 s
AlexNet	3.330 s	264.795 s	332.852 s
VGG16	6.816 s	301.268 s	373.483 s
VGG19	7.287 s	291.093 s	368.719 s

Table 3 shows that the model compilation times in the Gramine-SGX environment are significantly greater than the compilation times in the native environment. The inference times are also greater in the Gramine-SGX environment than in the native environment. The results are not unexpected because security always comes with a price.

Another set of experiments were conducted to obtain the inference times, compilation times and total execution times of the eight deep neural network models during image classification inferencing in a Mystiko-SGX environment. The results in Table 4 show that the inference and compilation times for all eight models are significantly higher in the Mystiko-SGX environment than the Gramine-SGX environment. For example, AlexNet model inference in Mystikos-SGX takes 1.3 s longer than in Gramine-SGX. Also, as seen in Fig. 8, its runtime memory footprint is 2.32 GB compared with 0.54 GB for Gramine-SGX. In general, Gramine-SGX is a better trusted execution environment than Mystiko-SGX in that it is less memory intensive and provides more utility and compatibility for applications intended to be ported to Intel SGX.

An additional safeguard would be to implement cache clearance at execution time. This would combat Prime+Probe attack variants that attempt to identify the sets being used by leveraging temporal cache access traces. However, Intel CPUs do not as yet provide an operation for flushing the cache at the user level before exiting an enclave.

7 Conclusions

This research has demonstrated that split computing can be leveraged as a deterrence measure to enhance the confidentiality of deep neural network models ported to Intel SGX environments. The evaluation demonstrates that the approach introduces negligible overhead while securing deep neural network models in transit and at rest in the hardware enclave. The research also provides useful benchmarking of the available libraries for out-of-the-box porting to Intel SGX trusted execution environments such as Gramine-SGX and Mystikos-SGX.

Acknowledgment. This research was supported by Institute for Infocomm Research, an A*STAR research entity, under the RIE2020 Advanced Manufacturing and Engineering (AME) Program (Award no. A19E3b0099).

References

1. Chen, G., Chen, S., Xiao, Y., Zhang, Y., Lin, Z., Lai, T.: SGXPectre: stealing Intel secrets from SGX enclaves via speculative execution. In: Proceedings of the IEEE European Symposium on Security and Privacy, pp. 142–157 (2019)
2. Choi, H., Bajic, I.: Deep feature compression for collaborative object detection. In: Proceedings of the Twenty-Fifth IEEE International Conference on Image Processing, pp. 3743–3747 (2018)
3. Costan, V., Lebedev, L., Devadas, S.: Secure processors. Part II: Intel SGX security analysis and MIT Sanctum architecture, Foundations and Trends in Electronic Design Automation **11**(3), 249–361 (2017)
4. Deis Labs, Mystikos, GitHub (https://github.com/deislabs/mystikos) (2023)
5. Fei, S., Yan, Z., Ding, W., Xie, H.: Security vulnerabilities of SGX and countermeasures: a survey. ACM Comput. Surv. **54**(6), 126 (2021)
6. Gramine Project Contributors, PyTorch PPML Framework, GitHub (https://github.com/gramineproject/graphene/blob/master/Documentation/tutorials/pytorch/index.rst) 2022
7. Gramine Project Contributors, Gramine Documentation (https://gramine.readthedocs.io/en/latest) (2023)
8. Intel, Understanding Intel Software Guard Extensions (Intel SGX), Santa Clara, California (https://intel.com/content/www/us/en/developer/articles/technical/intel-sgx-and-side-channels.html) (2023)
9. Itahara, S., Nishio, T., Yamamoto, K.: Packet-loss-tolerant split inference for delay-sensitive deep learning in lossy wireless networks. In: Proceedings of the IEEE Global Communications Conference (2021)
10. Jahier Pagliari, D., Chiaro, R., Macii, E., Poncino, M.: CRIME: input-dependent collaborative inference for recurrent neural networks. IEEE Trans. Comput. **70**(10), 1626–1639 (2021)
11. Kang, Y., et al.: Neurosurgeon: collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Comput. Architect. News **45**(1), 615–629 (2017)
12. Large-Scale Data and Systems Group, Spectre attack against SGX enclave, GitHub (https://github.com/lsds/spectre-attack-sgx) (2018)
13. Moghimi, A., Irazoqui, G., Eisenbarth, T.: CacheZoom: how SGX amplifies the power of cache attacks. In: Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems, pp. 69–90 (2017)
14. Narra, K., Lin, Z., Wang, Y., Balasubramaniam, K., Annavaram, M.: Privacy-preserving inference in machine learning services using trusted execution environments. arXiv: 1912.03485 (2019)
15. Pessl, P., Gruss, D., Maurice, C., Schwarz, M., Mangard, S.: DRAMA: exploiting DRAM addressing for cross-CPU attacks. In: Proceedings of the Twenty-Fifth USENIX Security Symposium, pp. 565–581 (2016)

16. PyTorch Team, AlexNet (https://pytorch.org/hub/pytorch_vision_alexnet) (2023)
17. PyTorch Team, ResNet (https://pytorch.org/hub/pytorch_vision_resnet) (2023)
18. PyTorch Team, Source code for torchvision.models.mobilenetv3 (https://pytorch.org/vision/stable/_modules/torchvision/models/mobilenetv3.html) (2023)
19. PyTorch Team, Squeezeent (https://pytorch.org/hub/pytorch_vision_squeezenet) (2023)
20. PyTorch Team, VGG-Nets (https://pytorch.org/hub/pytorch_vision_vgg) (2023)
21. Sanders, J.: Spectre and Meltdown explained: a comprehensive guide for professionals, TechRepublic, May 15 (2019)
22. Schwarz, M., Weiser, M., Gruss, D., Maurice, C., Mangard, S.: Malware guard extension: abusing intel SGX to conceal cache attacks, Cybersecurity, vol. 3, article no. 2 (2020)
23. Seaborn, M., Dullien, T.: Exploiting the DRAM rowhammer bug to gain kernel privileges, presented at Black Hat USA (2016)
24. Tramer, F., Boneh, D.: Slalom: fast, verifiable and private execution of neural networks in trusted hardware. arXiv:1806.03287v2 (2019)
25. Yarom, Y., Falkner, K.: Flush+Reload: a high resolution, low noise, L3 cache side-channel attack. In: Proceedings of the Twenty-Third USENIX Security Symposium, pp. 719–732 (2014)
26. Zhao, C., Saifuding, D., Tian, H., Zhang, Y., Xing, C.: On the performance of Intel SGX. In: Proceedings of the Thirteenth Web Information Systems and Applications Conference, pp. 184–187 (2016)

Automobile Security

A Cyber Security Analysis Methodology for Evaluating Automobile Risk Exposures

Kameron Tillman, Jason Staggs, and Sujeet Shenoi(✉)

University of Tulsa, Tulsa, Oklahoma, USA
sujeet@utulsa.edu

Abstract. Modern automobiles incorporate numerous sensors, actuators and electronic control units that work in concert to provide safe, efficient and comfortable driving experiences. Automobile convenience features introduce network connectivity via short-range wireless communications protocols and the Internet, potentially exposing the automobile electronics to remote attacks in addition to physical attacks. New attacks on modern automobiles are constantly being developed; their potential impacts range from inconvenience to severe injury and death.

This chapter describes a security analysis methodology for rapidly evaluating the risk exposures of modern automobiles. The methodology considers the automobile attack surfaces comprising the attack vectors that provide access to automobile targets and the potential impacts resulting from successful attacks on the accessed targets. Key features of the security analysis methodology are that it is holistic and rapid, and can be applied by individuals with limited expertise in automobile technologies and cyber security.

Keywords: Automobiles · Security Assessment Methodology · Attack Vectors · Targets · Attacks · Impacts · Risk Exposure

1 Introduction

Every year, new automobile models are introduced with the latest technologies, advanced safety, convenience and comfort features and ubiquitous connectivity to the Internet, Wi-Fi networks and mobile communications networks [19]. The introduction of highly-networked computing systems capable of controlling critical automobile functionality such as steering and braking in environments that formerly comprised hardwired electromechanical components raises significant security concerns. Automobile attack surfaces have also grown as convenience features provide external connectivity. In addition to physical attacks, it is possible to attack automobile systems remotely with adequate expertise, equipment and access.

Security researchers have developed and continue to develop novel automobile exploits. Passive keyless entry system attacks using inexpensive software-defined radios have been used to steal high-end automobiles [23]. Remote attacks that

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 195–220, 2024.
https://doi.org/10.1007/978-3-031-49585-4_10

disable electronic stability control systems have been demonstrated on Volkswagen automobiles [2]. Door unlocking/locking, exterior lighting and internal audio controls in Tesla automobiles have been remotely exploited [24]. The 2015 Jeep Grand Cherokee attacks remotely operated windshield wipers, engine and braking controls leading to a safety recall of 1.4 million automobiles [16,17].

While automobile exploits garner considerable attention, it is difficult for individuals and organizations to directly engage this knowledge to comprehend and evaluate the risk exposures of automobiles available for purchase, lease or rent. What is needed is a security analysis methodology that accommodates the complex, diverse and ever-changing cyber anatomies, configurations and features of automobiles, and that can be applied by individuals with limited expertise in automobile technologies and cyber security.

Several security analysis methodologies have been proposed to quantify the risk exposures of modern automobiles (see, e.g., [18,40]). However, the methodologies primarily provide composite numerical risk estimates. The principal problem with such estimates is that they do not express the true risk, which is best conveyed semantically in terms of the attack surfaces of automobiles, the various targets that can be accessed and the impacts of successful attacks on the targets.

This chapter describes a security analysis methodology for evaluating the risk exposures of modern automobiles. The methodology considers the automobile attack surfaces and the potential impacts resulting from successful attacks on the accessed targets. The methodology, which requires little if any expertise in automobile technologies and cyber security for its application, enables rapid risk assessments of automobiles available for purchase, lease or rent.

2 Related Work

Koscher et al. [21] investigated the physical attack surfaces of modern automobiles. Their research involved invasive automobile disassembly and extracting automobile components to identify vulnerabilities, attack strategies and potential attack impacts.

Checkoway et al. [9] focused on the cyber attack surfaces of modern automobiles. They developed and leveraged automobile component software exploits to extract critical information such as automobile location data.

Valasek and Miller [37] also investigated the cyber attack surfaces of modern automobiles. Their research involved invasive automobile disassembly to determine automobile attack surfaces.

In contrast, the proposed automobile security analysis methodology engages generic, albeit configurable, attack surfaces with additional attack vectors and configurable targets that may be attacked to cause negative impacts. The security analysis methodology relies on detailed descriptions of the cyber anatomies of modern automobiles that specify their network architectures, underlying systems and networks, network connectivity and the many physical and cyber attack vectors that constitute their attack surfaces.

The security analysis methodology does not require invasive automobile disassembly, vulnerability discovery or attack execution. Indeed, it can be applied by individuals with limited technical expertise in automobile technologies and cyber security. Importantly, the methodology can be completed within hours instead of days or weeks, enabling individuals and enterprises to quickly evaluate risk exposures and make informed choices when selecting automobiles for purchase, lease or rent.

3 Automobile Cyber Anatomy

Automobile networks are diverse and can be complex due to their proprietary sub-networks and protocols. This section describes a generic automobile network specification that models the diverse networks in modern automobiles in order to conduct realistic security analyses without tedious modeling efforts.

A modern automobile network has three principal sub-networks, High-Speed Controller Area Network (High-Speed CAN), Low-Speed CAN and Media Oriented Streaming Transport (MOST) network, which are typically interconnected via a gateway. The automobile network also provides an on-board diagnostics (OBD) interface.

3.1 High-Speed CAN

The High-Speed CAN is used for critical applications such as automobile acceleration and braking. Advanced driver assistance systems connect to the High-Speed CAN bus and interface with critical automobile controls to assist drivers with automobile operation. Examples of advanced driver assistance systems in a High-Speed CAN include the forward collision system, adaptive cruise control system, lane keep assist/departure warning systems, automatic parking system, engine start-stop system, electronic parking brake system and blind spot detection system:

- **Forward Collision System:** The forward collision system employs detection and ranging sensors that monitor driving conditions and alert drivers to potential frontal collisions [26]. Audible and visual warnings are triggered upon potential collision detection, providing drivers with additional time to react to adverse situations. If a collision is imminent, the forward collision system may engage the brakes automatically to mitigate the danger.
- **Adaptive Cruise Control System:** Figure 1 (left-hand side) illustrates the operation of the adaptive cruise control system. The system employs detection and ranging sensors that monitor vehicles traveling in front of the automobile in order to maintain a safe following distance. When a vehicle is detected, adaptive cruise control accelerates or decelerates the automobile as necessary to maintain a safe following distance. When no other vehicles are detected in proximity, the automobile travels at the set cruising speed. Some adaptive cruise control systems can bring automobiles to a complete stop in emergency situations [26].

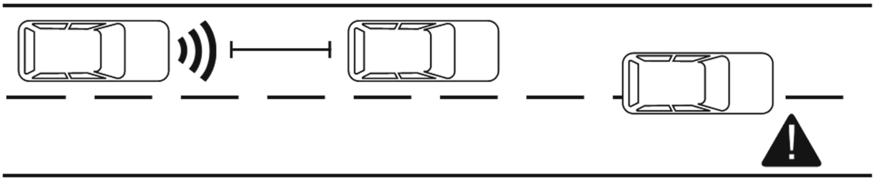


Fig. 1. Adaptive cruise control (left) and lane keep assist (right).

- **Lane Keep Assist/Departure Warning Systems:** Figure 1 (right-hand side) illustrates the operation of the lane keep assist system. The system employs vision sensors to monitor lane markings on the roadway [26]. If a driver moves from a lane without using a turn signal, the lane keep assist system steers the automobile to the center of the lane. Lane departure warning, an advanced driver assistance system similar to lane keep assist, alerts the driver when the automobile moves out of its lane [26].
- **Automatic Parking System:** The automatic parking system employs detection and ranging sensors to determine whether or not a parking spot can accommodate an automobile [26]. After a parking spot has been deemed suitable, the automatic parking electronic control unit computes the steering maneuvers and actuates the steering wheel to complete the parking operation. No steering input by the driver is required to park the automobile.
- **Engine Start-Stop System:** The engine start-stop system employs ignition controls to automatically shut down and restart an engine to reduce engine idling time and emissions. The system shuts down the engine during long stops, such as when waiting at a traffic light or in heavy traffic. When pressure on the brake pedal is released, the engine is restarted automatically and driving may resume [10].
- **Electronic Parking Brake System:** The electronic parking brake system employs actuators to engage the brakes and replaces the manual handbrake with an electronic control. The electronic parking brakes are integrated with other advanced driver assistance systems to enable rapid automatic braking. The brakes may engage automatically when an automobile is parked and the engine is shut down [33].
- **Blind Spot Detection System:** The blind spot detection system employs detection and ranging sensors to monitor automobile blind spots, areas where driver visibility is hampered or obfuscated. The system issues a visible alert when a blind spot is detected [26]. It also issues visible and audible alerts when a driver attempts to merge towards an automobile in the proximity of a blind spot.

3.2 Low-Speed CAN

The Low-Speed CAN supports non-critical applications such as the climate control, power mirror control, windshield wiper control and lighting control systems.

A Local Interconnect Network (LIN) is used to connect Low-Speed CAN bus devices to auxiliary components. For example, the LIN connects the power mirror control system to individual mirrors and connects the lighting control system to the headlights and taillights [30].

3.3 MOST Network

The MOST network connects devices that provide external network connectivity and automobile interfaces. Devices commonly connected in a MOST ring include the telematics system, infotainment system, Wi-Fi module, Bluetooth module and microphone array:

- **Telematics System:** The telematics system enables communications with external systems and networks. Modern automobile telematics systems are equipped with cellular modems for cellular network connectivity. In 2020, approximately 91% of the automobiles sold in the United States were equipped with cellular modems [20]. Modern automobiles with telematics systems support convenience features such as remote unlocking/locking and keyless ignition using a smartphone [26].
- **Infotainment System:** The infotainment system provides a human-machine interface for automobile applications software. The system presents live information (e.g., weather and traffic data) and entertainment options (e.g., media playback and AM/FM radio tuner). An infotainment system is often located at the center of the dashboard and is controlled via a touch screen, physical buttons or voice commands [30].
- **Wi-Fi Module:** The Wi-Fi module enables Wi-Fi hotspot network tethering functionality. A client device connects to the Wi-Fi hotspot in an automobile to obtain cellular network connectivity via the telematics system [38]. The Wi-Fi module also enables Wi-Fi connectivity, enabling an automobile to connect to a wireless network to stream media and download updates.
- **Bluetooth Module:** The Bluetooth module enables mobile device connectivity, media streaming and hands-free voice calling.
- **Microphone Array:** The microphone array enables a driver to interface with the infotainment system via voice commands. The microphone array can also be used to make hands-free voice calls.

3.4 On-Board Diagnostics Interface

The OBD interface is the primary entry point to the automobile network. OBD-II, the latest implementation, enables real-time reporting of automobile system data. It leverages the High-Speed and Low-Speed CANs to interact with practically every system in an automobile network [30].

4 Attack Vectors

Several physical and cyber attack vectors can be leveraged to access systems in automobile networks. Since multiple critical and non-critical automobile systems

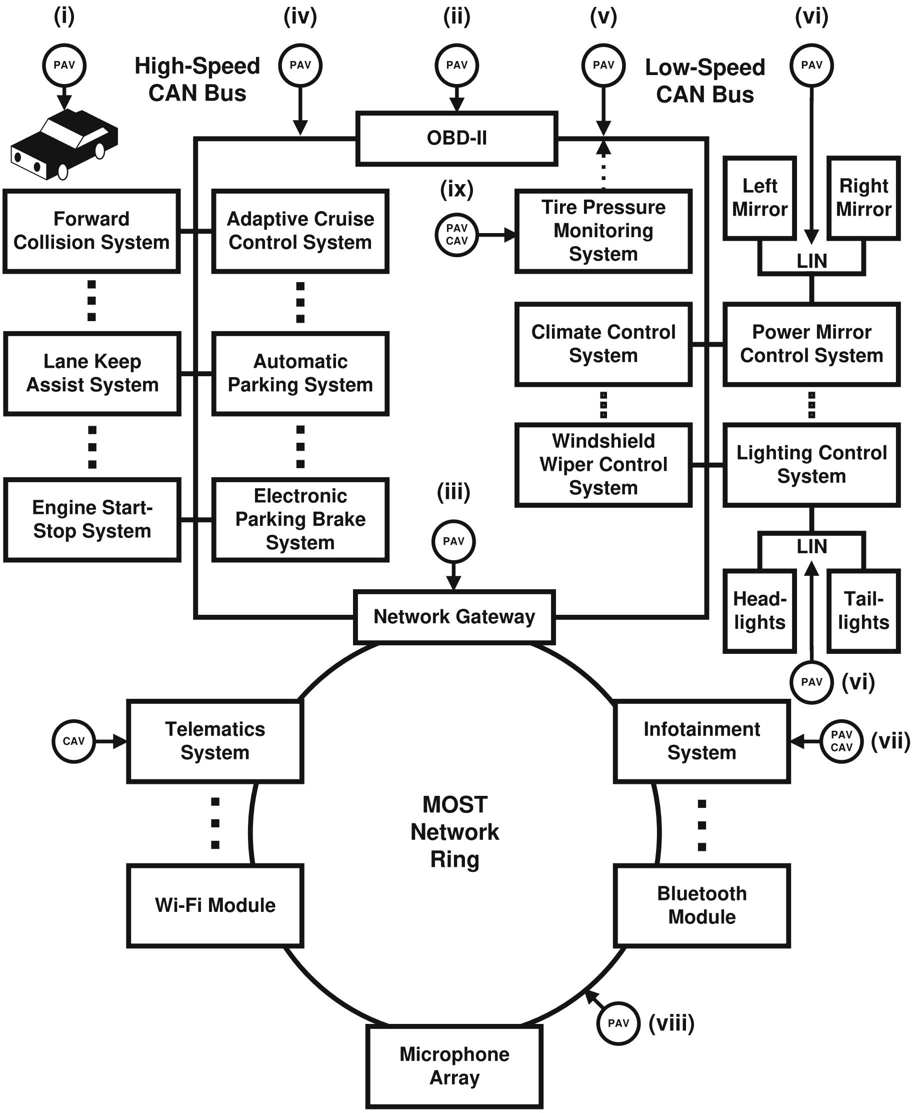


Fig. 2. Physical and cyber attack vectors providing access to automobile targets.

are interconnected, access to one system can be leveraged to access and target other systems and cause negative impacts to an automobile and its occupants.

Figure 2 shows the physical attack vectors (PAVs) and cyber attack vectors (CAVs). Because the focus is on normal (unmodified) automobiles, supply chain compromises of automobile systems and parasitic device implants in automobile systems are not considered.

4.1 Physical Attack Vectors

Koscher et al. [21] specified the physical attack surfaces of modern automobiles as of 2010. Additionally, they identified several vulnerabilities and demonstrated the negative impacts of successful attacks. In contrast, this work describes a generic, albeit configurable, attack surface for modern automobiles through 2022 models with additional physical attack vectors that cover new automobile technologies and designs, along with configurable targets that may be attacked to cause negative impacts.

Table 1 shows the physical attack vectors that provide access to automobile targets, which include the automobile interior, automobile networks and interconnected automobile systems. A physical attack vector provides hands-on access to a targeted system, following which an attack that exploits a vulnerability in the system may be executed to cause negative impacts. Access to the targeted system may also be leveraged to access and subsequently attack other connected automobile systems.

The physical attack vectors that provide access to automobile targets include physical access to the automobile interior, OBD interface, network gateway, High-Speed CAN, Low-Speed CAN, LIN, infotainment system, MOST network and tire pressure monitoring system (TPMS):

- **Physical Access to Automobile Interior:** Physical access to an automobile interior enables all the other attack vectors to be leveraged. An automobile interior may be accessed via graceful entry or forced entry. Graceful entry may employ a paired key fob, physical key or personal identification number keypad, which disengage the anti-theft system. A key fob relay attack [23] provides graceful entry to an automobile interior because it circumvents the anti-theft system. Forced entry, which may employ locksmith or invasive tools to gain access, does not disable the anti-theft system.
- **Physical Access to OBD Interface:** Physical access to an OBD-II interface enables interactions with practically every system in the High-Speed and Low-Speed CANs. OBD-II access may also enable communications with MOST network systems via the network gateway.
- **Physical Access to Network Gateway:** Physical access to a network gateway enables interactions with systems in the High-Speed and Low-Speed CANs as well as MOST network systems.
- **Physical Access to High-Speed CAN:** Physical access to a High-Speed CAN enables interactions with the interconnected critical automobile systems. High-Speed CAN systems may be distributed across multiple, segmented High-Speed CAN buses [37]. Therefore, access to one segmented High-Speed CAN bus may not enable interactions with systems in other segmented High-Speed CAN buses.

  Since a High-Speed CAN does not have authentication, addressing and message encryption schemes [6], CAN wiretapping can be leveraged to circumvent the OBD-II interface to gain direct High-Speed CAN bus connectivity [30]. In fact, malicious interactions with High-Speed CAN systems are easily accomplished.

Table 1. Network and target access provided by physical attack vectors.

Attack Vectors	Network Access	Target Access
Physical Access to Automobile Interior	*Direct Access* HS-CAN, LS-CAN LIN, MOST	ADAS, Other HS-CAN Systems, LS-CAN Systems, TPMS
Physical Access to On-Board Diagnostics Interface	*Direct Access* HS-CAN, LS-CAN, LIN, MOST	ADAS, Other HS-CAN Systems, LS-CAN Systems, TPMS, Microphone Array
Physical Access to Network Gateway	*Direct Access* HS-CAN, LS-CAN, LIN, MOST	ADAS, Other HS-CAN Systems, LS-CAN Systems, TPMS, Microphone Array
Physical Access to High-Speed CAN (HS-CAN)	*Direct Access* HS-CAN	ADAS, Other HS-CAN Systems
	Indirect Access LS-CAN, LIN, MOST	LS-CAN Systems, TPMS, Microphone Array
Physical Access to (LS-CAN)	*Direct Access* LS-CAN, LIN	LS-CAN Systems, TPMS
	Indirect Access HS-CAN, MOST	ADAS, Other HS-CAN Systems, Microphone Array
Physical Access to LIN	*Direct Access* LIN	LS-CAN Systems, TPMS
	Indirect Access LS-CAN	LS-CAN Systems, TPMS
Physical Access to Infotainment System	*Direct Access* HS-CAN, LS-CAN, MOST	ADAS, Other HS-CAN Systems, LS-CAN Systems, TPMS, Microphone Array
	Indirect Access LIN	LS-CAN Systems, TPMS
Physical Access to MOST Network	*Direct Access* MOST	Microphone Array
	Indirect Access HS-CAN, LS-CAN, LIN	ADAS, Other HS-CAN Systems, LS-CAN Systems, TPMS
Physical Access to TPMS (Connected)	*Direct Access* LS-CAN	LS-CAN Systems, TPMS
Physical Access to TPMS (Isolated)	*Direct Access* No Systems and Networks	TPMS

ADAS: Advanced driver assistance HS-CAN systems, HS-CAN: High-Speed CAN, LS-CAN: Low-Speed CAN

- **Physical Access to Low-Speed CAN:** Physical access to a Low-Speed CAN enables interactions with the interconnected non-critical automobile systems. Low-Speed CAN systems may be distributed across multiple, segmented Low-Speed CAN buses [37]. Therefore, access to one segmented Low-Speed CAN bus may not enable interactions with systems in other segmented Low-Speed CAN buses.
- **Physical Access to LIN:** Physical access to a LIN enables interactions with interconnected auxiliary systems and devices. A LIN may contain a master node that serves as a bridge to a Low-Speed CAN bus. In this case, physical access to the LIN would also enable interactions with systems in the connected Low-Speed CAN bus [30].
- **Physical Access to Infotainment System:** Physical access to an infotainment system enables interactions with automobile applications software. The infotainment system provides optical storage media slots, Secure Digital (SD) card slots and USB ports for media and update purposes [22]. The system may have a web browser for accessing local and remote websites. Physical access to the infotainment system in the MOST network may be leveraged to interact with systems in the High-Speed and Low-Speed CANs [30].

  Valasek and Miller [38] exploited a vulnerability in a 2015 Jeep Grand Cherokee infotainment system update mechanism using USB removable media. The attack enabled a custom software installation that provided privileged access to the infotainment system as well as all the other systems in the MOST network and High-Speed and Low-Speed CANs.

  Dimov [11] launched a denial-of-service attack on a Tesla Model 3 automobile using the automobile's web browser to connect to a malicious website. The infotainment system froze upon accessing the malicious website and disabled critical data reporting, including speedometer readings and battery status.
- **Physical Access to MOST Network:** Physical access to a MOST network enables interactions with the telematics system, infotainment system and microphone array. Physical tapping of a MOST network is difficult because the network employs a range of communications media. Smith [30] has suggested that MOST network systems should be targeted directly instead of via communications media. In an automobile with a network gateway, MOST network access enables subsequent access to the systems and devices in the High-Speed and Low-Speed CANs and connected LINs. Otherwise, access is limited to MOST network systems.
- **Physical Access to TPMS:** Physical access to a TPMS enables interactions with the TPMS and, possibly, Low-Speed CAN systems. Figure 2 illustrates the two possibilities for TPMS connectivity in automobile networks. Specifically, a TPMS may be isolated from all the automobile networks or it may be connected to the Low-Speed CAN [37].

  A TPMS utilizes sensors that measure and report tire pressure. The system warns the driver when one or more tires have low pressure. All passenger automobiles in the United States manufactured after 2007 require the installation of TPMSs [28].

Table 2. Network and target access provided by cyber attack vectors.

Attack Vectors	Network Access	Target Access
Cyber Access to	*Direct Access* HS-CAN, LS-CAN, MOST	Advanced Driver Assistance HS-CAN Systems, Other HS-CAN Systems, LS-CAN Systems, TPMS, Microphone Array
	Indirect Access LIN	LS-CAN Systems, TPMS
Cyber Access to Infotainment System	*Direct Access* HS-CAN, LS-CAN, MOST	Advanced Driver Assistance HS-CAN Systems, Other HS-CAN Systems, LS-CAN Systems, TPMS, Microphone Array
	Indirect Access LIN	LS-CAN Systems, TPMS
Cyber Access to TPMS (Connected)	*Direct Access* LS-CAN	LS-CAN Systems, TPMS
Cyber Access to TPMS (Isolated)	*Direct Access* No Systems and Networks	TPMS

HS-CAN: High-Speed CAN, LS-CAN: Low-Speed CAN

4.2 Cyber Attack Vectors

Checkoway et al. [9] and Valasek and Miller [37] have described the cyber attack surfaces of modern automobiles as of 2011 and 2014, respectively. Checkoway et al. [9] also specified several attacks and demonstrated their negative impacts. In contrast, this work describes a generic, albeit configurable, attack surface of modern automobiles through 2022 models with additional cyber attack vectors that cover new automobile technologies and designs, along with configurable targets that may be attacked to cause negative impacts.

Table 2 shows the cyber attack vectors that provide access to automobile targets, automobile networks and interconnected automobile systems. A cyber attack vector provides remote access to a targeted system, following which an attack that exploits a vulnerability in the targeted system can be executed to cause negative impacts. Access to a targeted system may also be leveraged to access and subsequently attack other connected automobile systems. The cyber attack vectors in an automobile network include cyber access to the telematics system, infotainment system and connected (as opposed to isolated) TPMS:

- **Cyber Access to Telematics System:** Cyber access to a telematics system enables remote interactions with MOST network, High-Speed CAN, Low-Speed CAN and LIN systems. Valasek and Miller [37] demonstrated that the telematics system of a 2015 Jeep Grand Cherokee telematics system can be accessed from anywhere with cellular network coverage. Connected vehicle services such as GM OnStar provide remote functionality, including door locking/unlocking, engine ignition and vehicle disabling in the event of theft [41]. Since connected vehicle services are made possible by the telematics system, access to the system enables the exploitation of all the connected vehicle services functionality.
- **Cyber Access to Infotainment System:** Cyber access to an infotainment system enables remote interactions with automobile applications software. Cyber access to the infotainment system in a MOST network may be leveraged to interact with systems in the High-Speed and Low-Speed CANs [30]. Cyber access to the infotainment system also enables connectivity to the Wi-Fi and Bluetooth modules.

  The Wi-Fi module provides Wi-Fi hotspot and Wi-Fi connectivity functionality. The Wi-Fi hotspot enables client devices to obtain cellular network connectivity via the telematics system. Wi-Fi connectivity enables an automobile to connect to a wireless network to stream media and download updates. Wi-Fi communications have ranges of hundreds of feet [25]. Services running on exposed ports may be accessible via the Wi-Fi hotspot and Wi-Fi connectivity, and are susceptible to exploitation. Vanhoef and Piessens [39] demonstrated vulnerabilities in Wi-Fi Protected Access (version 2), an outdated, but widely implemented, Wi-Fi security protocol that enables unauthorized network access and data interception.

  The Bluetooth module enables mobile device connectivity, media streaming and hands-free voice calling. Bluetooth communications have ranges of about 33 ft. The Bluetooth software stack has historically had vulnerabilities that can be exploited by denial-of-service and arbitrary code execution attacks [14].
- **Cyber Access to TPMS:** Cyber access to a connected TPMS enables interactions with the TPMS and, possibly, Low-Speed CAN systems. Figure 2 illustrates the two possibilities for TPMS connectivity in automobile networks. Specifically, a TPMS may be isolated from all the automobile networks or it may be connected to the Low-Speed CAN [37].

  A TPMS incorporates sensors that measure tire pressure and reports the pressure values via radio frequency communications. It is possible to reverse engineer TPMS messages and transmit false tire pressure data [37]. Automobile tracking capabilities via TPMS have been researched, but they appear to be impractical [3].

Table 3. Automobile network target functionality.

Target	Functionality
High-Speed CAN Systems	
Forward Collision System	Braking, Warning
Adaptive Cruise Control System	Acceleration, Braking
Lane Keep Assist System	Steering
Lane Departure Warning System	Warning
Automatic Parking System	Steering
Engine Start-Stop System	Ignition
Electronic Parking Brake System	Braking
Blind Spot Detection System	Warning
Telematics System	Remote Communications
Infotainment System	Information Reporting
Low-Speed CAN Systems	
Tire Pressure Monitoring System	Warning
Climate Control System	Comfort
Power Mirror Control System	Auxiliary Components
Windshield Wiper Control System	Auxiliary Components
Lighting Control System	Auxiliary Components
Telematics System	Remote Communications
Infotainment System	Information Reporting
MOST Network Systems	
Telematics System	Remote Communications
Infotainment System	Information Reporting
Microphone Array	Cabin Audio

5 Targets and Impacts

This section identifies the principal targets in a modern automobile network and describes the impacts of successful impacts on the targets.

5.1 Targets

Table 3 shows the targets in the High-Speed CAN, Low-Speed CAN and MOST network along with their functionality.

The High-Speed CAN connects several critical automobile systems. Advanced driver assistance systems that connect to the High-Speed CAN are the most attractive targets for attackers who wish to compromise automobile safety.

The Low-Speed CAN connects several non-critical automobile systems that can be targeted. The targets also include auxiliary devices implementing climate control and anti-theft functionality that are operated by Low-Speed CAN systems [37].

Table 4. Automobile network potential impacts.

Network	Targets	Potential Impacts
High-Speed CAN (HS-CAN)	*Direct Access* Advanced Driver Assistance HS-CAN Systems, Other HS-CAN Systems	Hazardous Operation, Non-Hazardous Operation
	Indirect Access LS-CAN Systems, TPMS, Microphone Array	Non-Hazardous Operation, Unauthorized Entry, Unauthorized Surveillance
Low-Speed CAN (LS-CAN)	*Direct Access* LS-CAN Systems, TPMS	Non-Hazardous Operation, Unauthorized Entry
	Indirect Access Advanced Driver Assistance HS-CAN Systems, Other HS-CAN Systems, Microphone Array	Hazardous Operation, Non-Hazardous Operation, Unauthorized Surveillance
LIN	*Direct Access* LS-CAN Systems, TPMS	Non-Hazardous Operation, Unauthorized Entry
MOST Network	*Direct Access* Microphone Array	Unauthorized Surveillance
	Indirect Access Advanced Driver Assistance HS-CAN Systems, Other HS-CAN Systems, LS-CAN Systems, TPMS	Hazardous Operation, Non-Hazardous Operation, Unauthorized Entry

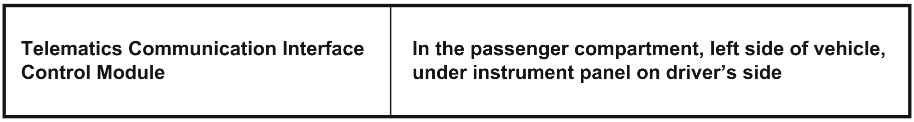
The MOST network connects systems that enable external network connectivity and the microphone array. The microphone array may be used via the infotainment system and connected vehicle services to transmit live automobile cabin audio. Connected vehicle services have been used by law enforcement to acquire evidence in criminal investigations [7].

5.2 Impacts

The negative impacts of compromising automobile network targets are hazardous and non-hazardous automobile control, unauthorized surveillance and unauthorized entry. Table 4 shows the potential impacts of attacks on targets in the High-Speed CAN, Low-Speed CAN, LIN and MOST network.

6 Methodology and Implementation

This section describes the security analysis methodology and its implementation.



Telematics Communication Interface Control Module	**In the passenger compartment, left side of vehicle, under instrument panel on driver's side**

Fig. 3. Service manual entry specifying a target location.

6.1 Methodology

The security analysis methodology involves four steps, automobile target specification, automobile status specification, attack vector chaining and realization, and attack specification and impact analysis:

– **Automobile Target Specification:** This step involves the specification of the targets in an automobile of interest. Modern automobiles come with varying options and configurations, as a result, some targets may not be present in the automobile of interest and a few targets may be located in different networks. This step eliminates the targets that are not present in the automobile of interest and configures the existing targets in the various automobile networks.

  The targets in a modern automobile can be specified after reviewing engineering documentation such as its user manual, service manual and wiring diagram:

  - *User Manual:* An automobile user manual lists the automobile systems and their functionality, and provides guidance on their use during automobile operation. A physical copy is typically located in the automobile glove compartment. Alternatively, a digital copy may be retrieved from the automobile manufacturer website at no cost.
  - *Service Manual:* A service manual provides detailed guidance about automobile service and maintenance procedures for the main automobile systems, which provide valuable information about potential targets. Service manuals are available for purchase online and at automobile parts stores. A service manual is useful for determining whether or not potential targets are installed in an automobile. Figure 3 shows a service manual entry specifying the location of the telematics system in an automobile.
  - *Wiring Diagram:* A wiring diagram, which specifies the connections of electrical components, also provides details about the electronic components (potential targets) installed in an automobile. Figure 4 shows a wiring diagram that specifies the connectivity of the telematics system in an automobile.

    Wiring diagrams may be available online. Alternatively, the diagrams may be purchased directly from automobile manufacturers or from third-party vendors.

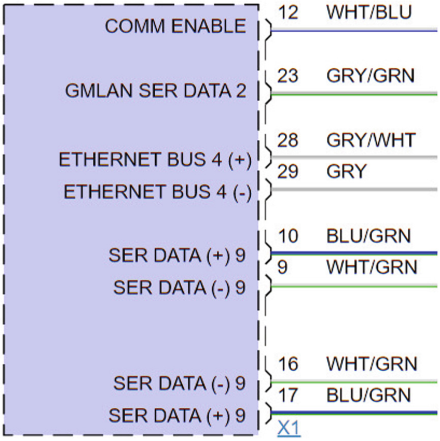


Fig. 4. Wiring diagram showing telematics system connectivity.

Examining automobile fuse blocks also assists in automobile target specification. A fuse block distributes electricity to all automobile systems and each fuse in the fuse block provides overcurrent protection for an automobile system (target). An automobile typically has at least two fuse blocks. One fuse block may be located in the engine bay near the 12 V battery whereas the other fuse block may be located in the cabin near the dashboard [12].

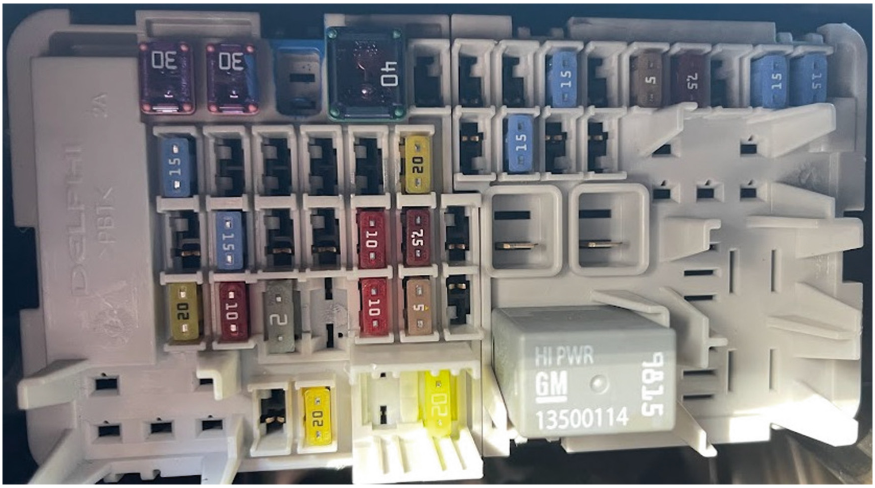
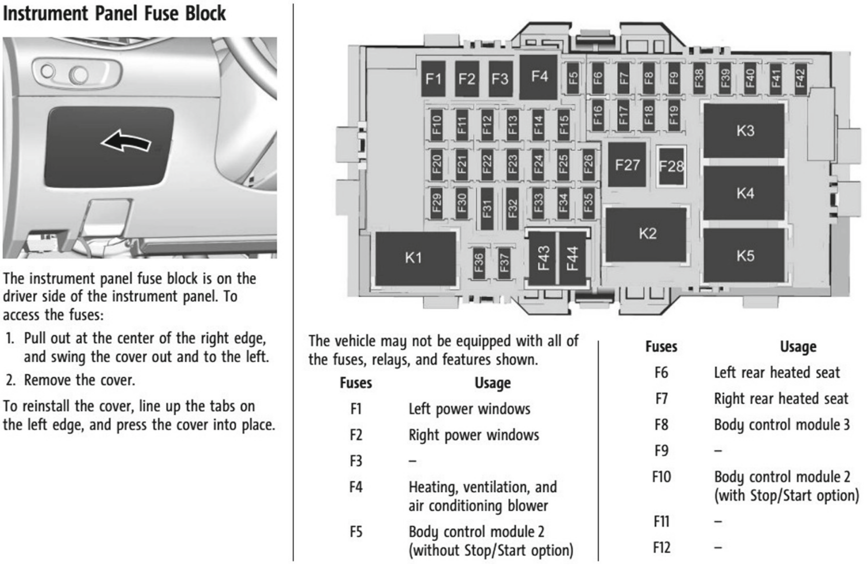


Fig. 5. Automobile fuse block.



Instrument Panel Fuse Block

F1
F2
F3
F4
F5
F6
F7
F8
F9
F38
F39
F40
F41
F42
F10
F11
F12
F13
F14
F15
F16
F17
F18
F19
K3
F20
F21
F22
F23
F24
F25
F26
F27
F28
K4
F29
F30
F31
F32
F33
F34
F35
K2
K1
F36
F37
F43
F44
K5

The instrument panel fuse block is on the driver side of the instrument panel. To access the fuses:

1. Pull out at the center of the right edge, and swing the cover out and to the left.
2. Remove the cover.

To reinstall the cover, line up the tabs on the left edge, and press the cover into place.

The vehicle may not be equipped with all of the fuses, relays, and features shown.

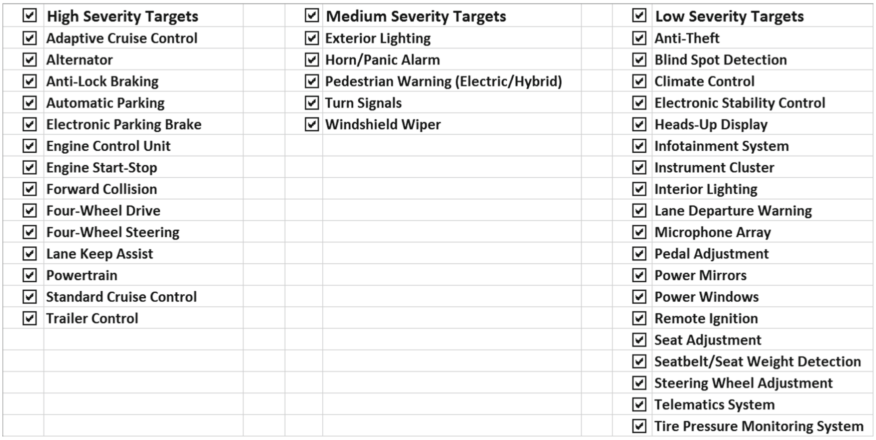
Fuses	Usage
F1	Left power windows
F2	Right power windows
F3	–
F4	Heating, ventilation, and air conditioning blower
F5	Body control module 2 (without Stop/Start option)
F6	Left rear heated seat
F7	Right rear heated seat
F8	Body control module 3
F9	–
F10	Body control module 2 (with Stop/Start option)
F11	–
F12	–

Fig. 6. User manual fuse block diagram.

Figure 5 shows an automobile fuse block. Each fuse block socket is assigned to an electrical system. The engineering documentation specifies the location and purpose of each fuse block socket.

Figure 6 shows a user manual listing each fuse socket location and fuse purpose. If a fuse block socket assigned to a target system in an automobile has a fuse installed, the target can be assumed to be present in the automobile. Otherwise, the target is unlikely to be present.

- **Automobile Status Specification:** This step involves the specification of the automobile status, which includes whether it is stationary or moving and whether or not it is possible to achieve engine ignition. Note that an automobile may be attacked when it is stationary or moving. Also, certain attack vectors are not realizable without automobile engine ignition. Clearly, engine ignition is active in a moving automobile. Engine ignition can be achieved in a stationary automobile by leveraging some of the physical access to automobile interior subvectors or the cyber access to telematics system vector via its connected vehicle services subvector.
- **Attack Vector Chaining and Realization:** This step chains individual attack vectors and subvectors to determine target reachability. This is followed by the specification of the realized attack vectors to identify the automobile targets that become accessible.
- **Attack Specification and Impact Analysis:** This step involves the specification of attacks on the accessible automobile targets. The attacks produce impacts – hazardous operation, non-hazardous operation, unauthorized entry and unauthorized surveillance – depending on target functionality and attack



☑ **High Severity Targets**
- ☑ Adaptive Cruise Control
- ☑ Alternator
- ☑ Anti-Lock Braking
- ☑ Automatic Parking
- ☑ Electronic Parking Brake
- ☑ Engine Control Unit
- ☑ Engine Start-Stop
- ☑ Forward Collision
- ☑ Four-Wheel Drive
- ☑ Four-Wheel Steering
- ☑ Lane Keep Assist
- ☑ Powertrain
- ☑ Standard Cruise Control
- ☑ Trailer Control

☑ **Medium Severity Targets**
- ☑ Exterior Lighting
- ☑ Horn/Panic Alarm
- ☑ Pedestrian Warning (Electric/Hybrid)
- ☑ Turn Signals
- ☑ Windshield Wiper

☑ **Low Severity Targets**
- ☑ Anti-Theft
- ☑ Blind Spot Detection
- ☑ Climate Control
- ☑ Electronic Stability Control
- ☑ Heads-Up Display
- ☑ Infotainment System
- ☑ Instrument Cluster
- ☑ Interior Lighting
- ☑ Lane Departure Warning
- ☑ Microphone Array
- ☑ Pedal Adjustment
- ☑ Power Mirrors
- ☑ Power Windows
- ☑ Remote Ignition
- ☑ Seat Adjustment
- ☑ Seatbelt/Seat Weight Detection
- ☑ Steering Wheel Adjustment
- ☑ Telematics System
- ☑ Tire Pressure Monitoring System

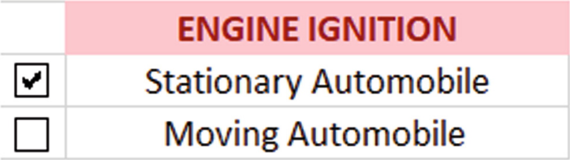
Fig. 7. Automobile target specification view.

type. Section 5.2 presents the potential impacts caused by successful attacks on automobile targets.

6.2 Implementation

A user-friendly visualization tool was written in Microsoft Excel and Visual Basic Scripting Edition to support automobile security analyses. In particular, the tool supports configurable security analyses, conveying automobile risk exposures in terms of their attack surfaces, exploitable targets and impacts. The tool accommodates the specification of diverse automobile cyber anatomies and configurations. The attack vectors, targets and impacts are customizable to support new technologies and advanced safety, convenience and comfort features as they are incorporated in automobiles.

Figure 7 shows the automobile target specification view that lists the available targets and provides options to connect targets (checked boxes) and disconnect targets (unchecked boxes). Because a disconnected target is omitted from further consideration in the methodology, this feature supports what-if analyses.

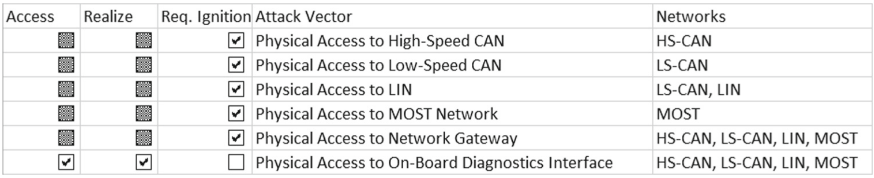


ENGINE IGNITION

☑ Stationary Automobile

□ Moving Automobile

Fig. 8. Automobile status specification view.

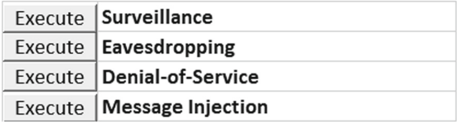


Access	Realize	Req. Ignition	Attack Vector	Networks
▨	▨	☑	Physical Access to High-Speed CAN	HS-CAN
▨	▨	☑	Physical Access to Low-Speed CAN	LS-CAN
▨	▨	☑	Physical Access to LIN	LS-CAN, LIN
▨	▨	☑	Physical Access to MOST Network	MOST
▨	▨	☑	Physical Access to Network Gateway	HS-CAN, LS-CAN, LIN, MOST
☑	☑	□	Physical Access to On-Board Diagnostics Interface	HS-CAN, LS-CAN, LIN, MOST

Fig. 9. Attack vector chaining and realization view.

Figure 8 shows the automobile status specification view, which provides stationary or moving automobile options. A stationary automobile attack is selected (checked box). Note that engine ignition is highlighted, corresponding to the default situation that the stationary vehicle is off.

Figure 9 shows the attack vector chaining and realization view. As mentioned above, individual attack vectors and subvectors are chained to determine target reachability. The require ignition component determines whether engine ignition is needed (checked box) or not needed (unchecked box) before an attack vector may be realized. The black boxes indicate the attack vectors that are not accessible or realizable until engine ignition is gained. The physical access to OBD attack vector is realizable because ignition is not required for physical access.

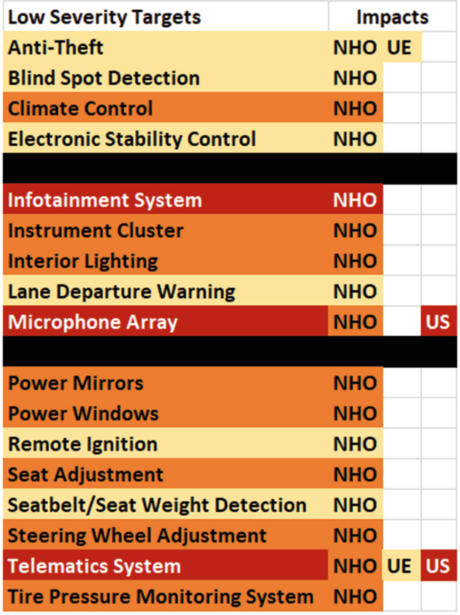


Execute	**Surveillance**
Execute	**Eavesdropping**
Execute	**Denial-of-Service**
Execute	**Message Injection**

Fig. 10. Attack execution subview.

Figure 10 shows the attack execution subview of the attack specification and impact analysis view. Four attacks, surveillance, eavesdropping, denial-of-service and message injection, may be executed on accessible targets to produce impacts:

- **Surveillance Attack:** A surveillance attack collects information about an automobile and/or its occupants. Examples include automobile location and cabin audio, including occupant conversations. A surveillance attack has a non-hazardous operation impact.
- **Eavesdropping Attack:** An eavesdropping attack collects messages transmitted between automobile targets. The messages could be analyzed to develop and execute additional attacks. An eavesdropping attack has a non-hazardous operation impact.
- **Denial-of-Service Attack:** A denial-of-service attack interrupts automobile network communications, resulting in a hazardous operation impact.



Low Severity Targets	Impacts		
Anti-Theft	NHO	UE	
Blind Spot Detection	NHO		
Climate Control	NHO		
Electronic Stability Control	NHO		
Infotainment System	NHO		
Instrument Cluster	NHO		
Interior Lighting	NHO		
Lane Departure Warning	NHO		
Microphone Array	NHO		US
Power Mirrors	NHO		
Power Windows	NHO		
Remote Ignition	NHO		
Seat Adjustment	NHO		
Seatbelt/Seat Weight Detection	NHO		
Steering Wheel Adjustment	NHO		
Telematics System	NHO	UE	US
Tire Pressure Monitoring System	NHO		

Fig. 11. Attack specification and impact analysis view.

- **Message Injection Attack:** A message injection attack transmits crafted messages for a malicious purpose, resulting in a hazardous operation impact.

Figure 11 shows the attack specification and impact analysis view, which displays the targets and their potential impacts. The targets and their associated attack impacts are highlighted in different colors according to their status. The four types of targets and their associated impact status are:

- **Disconnected Target:** A disconnected target is highlighted in black. A disconnected target is not present in the automobile and, therefore, is neither accessed nor attacked.
- **Accessible Target:** A target that is potentially accessible by an (unrealized) attack vector is highlighted in light gray. The impact, highlighted in light gray, indicates that the potential exists for a negative outcome if an attack vector is realized to access the target and a successful attack is executed on the target.
- **Accessed Target:** A target that has been accessed via a realized attack vector is highlighted in gray. The impact, highlighted in gray, indicates that the potential exists for a negative outcome because an attack vector enabling access the target is realized, providing an opportunity to attack the target.
- **Attacked Target:** A target that is successfully accessed and attacked is highlighted in dark gray with white text. The impact, highlighted in dark gray with white text, indicates that the negative outcome is realized.

7 Case Study

The visualization tool was used to conduct security analyses of five representative automobiles that consider their attack surfaces, targets and impacts. The automobiles were selected due to their presence in U.S. Government fleets. The automobiles comprise two vehicle classes, sport utility vehicles and sedans, representing a combined 26% of U.S. Government fleets during the 2020 fiscal year [36]. The five automobile models represent three motor groups with a combined 39% of the U.S. market share in 2021 [32].

The case study considers stationary automobiles in a rental car scenario where the automobile key fobs are available. Physical access to the automobile interior via the key fob attack vector is leveraged and realized, following which engine ignition is achieved. Subsequently, the physical access to the OBD interface and physical access to the network gateway attack vectors are both leveraged and realized. After the attack vectors are realized, surveillance, eavesdropping, denial-of-service and message injection attacks are executed on the automobile targets. The impacts of successful attacks include hazardous operation (HO), non-hazardous operation (NHO), unauthorized entry (UE) and unauthorized surveillance (US).

7.1 Automobile Security Analyses

This section describes the five automobiles in the case study and presents the results of the automobile security analyses that consider surveillance, eavesdropping, denial-of-service and message injection attacks. For security reasons, the makes and models of the five automobiles are not specified. Table 5 shows the high, medium and low severity targets present in the five automobiles.

- **Automobile A:** Automobile A is a domestic, full-size sport utility vehicle with a six-cylinder engine. The spacious automobile seats up to eight passengers.
- **Automobile B:** Automobile B is a domestic, full-size sport utility vehicle with an eight-cylinder engine. The spacious automobile seats up to eight passengers.
- **Automobile C:** Automobile C is a domestic, full-size sedan with a four-cylinder engine. The automobile seats up to five passengers.
- **Automobile D:** Automobile D is a domestic, full-size sedan with a six-cylinder engine. The automobile seats up to five passengers and is optimized for performance.
- **Automobile E:** Automobile E is a domestic, mid-size sport utility vehicle with a six-cylinder engine. The spacious automobile seats up to seven passengers.

Table 5. Automobile targets.

Target	Automobile				
	A	B	C	D	E
High Severity Targets					
Adaptive Cruise Control					✓
Alternator	✓	✓	✓	✓	✓
Anti-Lock Braking	✓	✓	✓	✓	✓
Automatic Parking					✓
Electronic Parking Brake	✓	✓			
Engine Control Unit	✓	✓	✓	✓	✓
Engine Start-Stop	✓	✓	✓		
Forward Collision		✓			✓
Four-Wheel Drive	✓				
Four-Wheel Steering					
Lane Keep Assist	✓				✓
Powertrain	✓	✓	✓	✓	✓
Standard Cruise Control	✓	✓	✓	✓	✓
Trailer Control	✓	✓			
Medium Severity Targets					
Exterior Lighting	✓	✓	✓	✓	✓
Horn/Panic Alarm	✓	✓	✓	✓	✓
Pedestrian Warning					
Turn Signals	✓	✓	✓	✓	✓
Windshield Wiper	✓	✓	✓	✓	✓
Low Severity Targets					
Anti-Theft	✓	✓	✓	✓	✓
Blind Spot Detection					✓
Climate Control	✓	✓	✓	✓	✓
Electronic Stability Control	✓	✓	✓	✓	✓
Heads-Up Display					
Infotainment System	✓	✓	✓	✓	✓
Instrument Cluster	✓	✓	✓	✓	✓
Interior Lighting	✓	✓	✓	✓	✓
Lane Departure Warning	✓				✓
Microphone Array	✓	✓	✓	✓	✓
Pedal Adjustment					✓
Power Mirrors	✓	✓	✓	✓	✓
Power Windows	✓	✓	✓	✓	✓
Remote Ignition	✓	✓	✓	✓	✓
Seat Adjustment	✓	✓	✓	✓	✓
Seatbelt/Seat Weight Detection	✓	✓	✓	✓	✓
Steering Wheel Adjustment	✓	✓			✓
Telematics System	✓	✓	✓		
Tire Pressure Monitoring System	✓	✓	✓	✓	✓

7.2 Risk Comparison

Table 6 shows the numbers of high severity, medium severity and low severity targets in the five automobiles in the case study. The automobiles are listed in rank order based on the total number of targets with ties being broken based on the numbers of high severity, medium severity and, finally, low severity targets.

As expected, Automobile A, a high-end sport utility vehicle with loaded features, is tied for the most number of targets and has the most number of high severity targets. Automobile E, a high-end sport utility vehicle with loaded features, is tied for the most number of targets and has just one high severity target less than Automobile A. A key security feature of Automobile E is that it does not have a telematics system, which reduces its attack surface and exposure to remote attacks due its lack of cellular network connectivity. Automobile B is also a high-end sport utility vehicle but not as loaded as Automobiles A and E; it has just one less high severity target and one less low severity target than Automobile A.

Automobiles C and D are economy sedans with fewer features than Automobiles A, B and E. Automobile D has the fewest targets overall. Also, it does not have a telematics system, which reduces its attack surface and exposure to remote attacks.

Table 7 shows the types and numbers of impacts realized by successful surveillance, eavesdropping, denial-of-service and messaging attacks on the five automobiles in the case study. Note that the automobiles are listed in rank order based on the total number of impacts.

The results in Table 7 parallel those in Table 6. As expected, the high-end sport utility vehicles with loaded features have significantly more negative impacts than the economy sedans. Automobiles D and E that do not have telematics systems have less exposure to surveillance attacks than the other automobiles. Eavesdropping attacks, which are passive in nature, induce non-hazardous operation impacts on all five automobiles.

In contrast, denial-of-service and message injection, which are active attacks, induce numerous hazardous operation impacts. Message injection attacks are more serious than denial-of-service attacks because they target the anti-theft system, microphone array and telematics system, inducing additional impacts.

Table 6. Automobile targets.

Automobile	High Severity Targets	Medium Severity Targets	Low Severity Targets	Total
Automobile A	10	4	16	30
Automobile E	9	4	17	30
Automobile B	9	4	15	28
Automobile C	6	4	14	24
Automobile D	5	4	13	22

Table 7. Automobile attack impacts.

Automobile	Surveillance Attack Impacts	Eavesdropping Attack Impacts	Denial-of-Service Attack Impacts	Message Injection Attack Impacts	Total
Automobile A	NHO: 1; US: 2	NHO: 30	HO: 10; NHO: 30	HO: 10; NHO: 30; UE: 2; US: 2	117
Automobile E	NHO: 1; US: 1	NHO: 30	HO: 9; NHO: 30	HO: 9; NHO: 30; UE: 1; US: 1	112
Automobile B	NHO: 1; US: 2	NHO: 28	HO: 9; NHO: 28	HO: 9; NHO: 28; UE: 2; US: 2	109
Automobile C	NHO: 1; US: 2	NHO: 24	HO: 6; NHO: 24	HO: 6; NHO: 24; UE: 2; US: 2	91
Automobile D	NHO: 1; US: 1	NHO: 22	HO: 5; NHO: 22	HO: 5; NHO: 22; UE: 1; US: 1	80

HO: Hazardous operation, NHO: Non-hazardous operation, UE: Unauthorized entry, US: Unauthorized surveillance

8 Conclusions

The incorporation of highly-networked computing systems that automatically control vital functions in modern automobiles raises significant security concerns. Unfortunately, because modern automobiles have complex and diverse cyber anatomies, configurations and features, it is difficult to comprehend and evaluate their risk exposures.

The security analysis methodology described in this chapter engages generic, albeit configurable, automobile attack surfaces along with configurable targets that may be attacked to cause negative impacts. In particular, the methodology relies on detailed descriptions of the cyber anatomies of modern automobiles that specify their network architectures, underlying systems and networks, network connectivity and the many physical and cyber attack vectors that constitute their attack surfaces. Reachability analysis is employed to chain the realizable attack vectors and determine all the accessible targets. Attack opportunities made possible by the realized physical and cyber attack vectors are identified, following which the impacts on an automobile and its occupants are determined.

The security analysis case study illustrates the advantages of the methodology. In particular, the methodology provides rapid insights into the risk exposures of modern automobiles in terms of attack surfaces, targets and impacts, enabling risk comparisons between automobiles of diverse makes and models. Additionally, the methodology facilitates cyber operations and cyber defense postures on automobiles. Cyber operations analysts can leverage the methodology as a playbook to develop sophisticated targeting of automobiles. Cyber defense analysts can draw on the attack vectors, reachable targets and possible attacks and their impacts to steer efforts directed at reducing risk by helping articulate and prioritize mitigations and security controls. The methodology also supports effective security analyses without drawing on extensive subject-matter knowledge, expensive experimentation and complex computations. Individuals and enterprises can rapidly assess and compare the complex security environments of automobiles as they consider alternatives for purchase, lease or rent. Additionally, it is possible to evaluate the security environments of new automobiles with evolving technologies, systems and features.

Acknowledgment. This research was supported by the National Science Foundation under Grant no. DGE 1501177.

References

1. Alfa Network, AWUS1900, Taipei City, Taiwan (2022). (www.alfa.com.tw/products/awus1900)
2. Allan, M.: "Serious" security flaws expose popular Ford and VW cars to hackers, Banbury Guardian, 13 April 2020
3. Ashworth, J., Staggs, J., Shenoi, S.: Radio frequency identification and tracking of vehicles and drivers by exploiting keyless entry systems. Int. J. Crit. Infrastruct. Prot. **40** (2023). Article no. 100587
4. Blanco, S.: Car hacking danger is likely closer than you think, Car and Driver, 4 September 2021
5. Bosch, CAN Specification, Version 2.0, Stuttgart, Germany (1991). (esd.cs.ucr.edu/webres/can20.pdf)
6. Bozdal, M., Samie, M., Aslam, S., Jennions, I.: Evaluation of CAN bus security challenges. Sensors **20**(8) (2020). Article no. 2364
7. Brewster, T.: Cartapping: How feds have spied on connected cars for 15 years, Forbes, 15 January 2017
8. California Air Resources Board, On-Board Diagnostic II (OBD II) Systems Fact Sheet, Sacramento, California, 19 September 2019. (ww2.arb.ca.gov/resources/fact-sheets/board-diagnostic-ii-obd-ii-systems-fact-sheet)
9. Checkoway, S., et al.: Comprehensive experimental analyses of automotive attack surfaces. In: Proceedings of the Twentieth USENIX Security Symposium, pp. 77–92 (2011)
10. Cowell, K.: Engine stop/start systems on non-hybrid vehicles, Car and Driver, 4 March 2011
11. Dimov, D.: Tesla Model 3 vulnerability: What you need to know about the web browser bug, Infosec Blog, Infosec Institute, Madison, Wisconsin, 5 August 2020. (resources.infosecinstitute.com/topic/tesla-model-3-vulnerability-what-you-need-to-know-about-the-web-browser-bug)
12. Duffy, J.: Modern Automotive Technology. Goodheart-Wilcox Company, Tinley Park (2017)
13. Ettus Research, USRP B210 (board only), Austin, Texas (2022). (www.ettus.com/all-products/ub210-kit)

14. Garbelini, M., Chattopadhyay, S., Bedi, V., Sun, S., Kurniawan, E.: BrakTooth: Causing Havoc on Bluetooth Link Manager, Vulnerability Disclosure Report, Singapore University of Technology and Design, Singapore (2021). (asset-group.github.io/disclosures/braktooth/braktooth.pdf)
15. Great Scott Gadgets, Throwing Star LAN Tap, Lakewood, Colorado (2022). (greatscottgadgets.com/throwingstar)
16. Greenberg, A.: Hackers remotely kill a Jeep on the highway - With me in it, Wired, 21 July 2015
17. Greenberg, A.: After Jeep hack, Chrysler recalls 1.4M vehicles for bug fix, Wired, 24 July 2015
18. HEAVENS Consortium, Healing Vulnerabilities to Enhance Software Security and Safety, Volvo Technology, Goteborg, Sweden (2016). (www.autosec.se/wp-content/uploads/2018/03/HEAVENS_D2_v2.0.pdf)
19. Jeffs, J.: A history of ADAS: Emergence to essential, IDTech-Ex, Cambridge, United Kingdom, 4 January 2022. (www.idtechex.com/en/research-article/a-history-of-adas-emergence-to-essential/25592)
20. Kosche, C.: How many connected cars are sold worldwide? Smartcar Blog, Smartcar, Mountain View, California, 15 April 2021. (www.smartcar.com/blog/connected-cars-worldwide)
21. Koscher, K., et al.: Experimental security analysis of a modern automobile. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 447–462 (2010)
22. Lin, T., Chen, L.: Common attacks against car infotainment systems. Presented at the Automotive Linux Summit (2019)
23. Linder, C.: Five impressive ways criminals use wireless signals to steal everything - Even your car, Popular Mechanics, 27 November 2019
24. McFarland, M.: Teen's Tesla hack shows how vulnerable third-party apps may make cars, CNN, 2 February 2022
25. Mitchell, B.: What is the range of a typical Wi-Fi network? Lifewire, New York (2020). (www.lifewire.com/range-of-typical-wifi-network-816564)
26. Moller, D., Haas, R.: Guide to Automotive Connectivity and Cybersecurity: Trends, Technologies, Innovations and Applications. Springer, Cham (2019)
27. MZD-AIO Contributors, MZD-AIO, GitHub (2020). (github.com/Trevelopment/MZD-AIO)
28. National Highway Traffic Safety Administration, 49 CFR §571.138 - Standard No. 138, Tire Pressure Monitoring Systems, Washington, DC (2011). (www.govinfo.gov/content/pkg/CFR-2011-title49-vol6/pdf/CFR-2011-title49-vol6-sec571-138.pdf)
29. Richards, P.: A CAN Physical Layer Discussion, Application Note AN228, Microchip Technology, Chandler, Arizona (2002). (ww1.microchip.com/downloads/en/appnotes/0228a.pdf)
30. Smith, C.: The Car Hacker's Handbook: A Guide for the Penetration Tester. No Starch Press, San Francisco (2016)
31. Software Radio Systems, srsRAN 22.04 Documentation, Cork, Ireland (2022). (docs.srsran.com/en/latest)
32. Statista, Estimated U.S. market share held by selected automotive manufacturers in 2021, Hamburg, Germany (2022). (www.statista.com/statistics/249375/us-market-share-of-selected-automobile-manufacturers)
33. Taylor, J.: There's no stopping the electric parking brake, Auto Service Professional, 16 February 2018
34. Tutorials Point, Ethical hacking - Wireless hacking, Hyderabad, India (2022). (www.tutorialspoint.com/ethical_hacking/ethical_hacking_wireless.htm)

35. UAB 8 Devices, Korlan USB2CAN, Vilnius, Lithuania (2022). (www.8devices.com/pro ducts/usb2can_korlan)
36. U.S. General Services Administration, FY 2020 Federal Fleet Open Data Set, Washington, DC (2021). (www.gsa.gov/cdnstatic/FY2020FederalFleetReport.xlsx)
37. Valasek, C., Miller, C.: A Survey of Remote Automotive Attack Surfaces. Technical White Paper, IOActive, Seattle, Washington (2014)
38. Valasek, C., Miller, C.: Remote Exploitation of an Unaltered Passenger Vehicle. Technical White Paper, IOActive, Seattle, Washington (2015)
39. Vanhoef, M., Piessens, F.: Key reinstallation attacks: forcing nonce reuse in WPA2. In: Proceedings of the Twenty-Fourth ACM Conference on Computer and Communications Security, pp. 1313–1328 (2017)
40. Wang, Y., Wang, Y., Qin, H., Ji, H., Zhang, Y., Wang, J.: A systematic risk assessment framework of automotive cybersecurity. Autom. Innov. **4**(3), 253–261 (2021)
41. Yarkoni, O.: The danger of connected car mobile apps to OEMs and smart mobility services, Upstream Blog, Novi, Michigan, 19 January 2022. (www.upstream.auto/blog/mobile-apps-pose-major-threat)

Real-Time Attack Detection in Modern Automobile Controller Area Networks

Edward Martin and Sujeet Shenoi(✉)

University of Tulsa, Tulsa, OK, USA
sujeet@utulsa.edu

Abstract. Modern automobiles have numerous sensors, actuators and electronic systems interconnected via internal sub-networks that are not designed with security in mind. This chapter describes a novel real-time system that employs long short-term memory networks to monitor automobile controller area networks, detect attacks and raise alerts. A repeatable design framework is employed to construct and train multiple long short-term memory networks to recognize normal controller area network message timing patterns. The framework lays out the computational resources as well as the data collection and preprocessing and long short-term memory network model development and training steps. Also, it enables new long short-term memory network models to be trained and updated for automobiles of different makes, models and years.

The attack detection system leverages a server-client configuration to monitor an automobile controller area network bus. The server is an inexpensive Raspberry Pi device connected directly to the automobile controller area network bus that captures, logs and transmits controller area network message traffic to a client via a Wi-Fi network. The client, a workstation located outside the automobile, provides the computational resources for real-time attack detection. Trained long short-term memory models executing on the client workstation analyze the received controller area network messages, identify attacks and send alerts via the Wi-Fi network. Experimental results using a 2010 Toyota Prius testbed and a fully-operational 2014 Toyota Prius automobile demonstrate the effectiveness of the real-time attack detection system.

Keywords: Automobiles · Controller Area Networks · Real-Time Attack Detection · Long Short-Term Memory Networks

1 Introduction

Modern automobiles incorporate numerous sensors, actuators and diverse electronic systems that are interconnected by sub-networks to provide safe, convenient and comfortable experiences to drivers and passengers. The principal internal sub-networks, High-Speed Controller Area Network (High-Speed CAN), Low-Speed Controller Area Network (Low-Speed CAN), Local Interconnect Network (LIN) and Media Oriented Systems Transport (MOST) network, support

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, pp. 221–252, 2024.
https://doi.org/10.1007/978-3-031-49585-4_11

automobile functionality [20]. The CAN protocol is employed by all the interconnected automobile applications that provide safety, convenience and comfort [2,3].

Unfortunately, modern automobile networks are not designed with security in mind [4,18]. First, the four principal sub-networks are interconnected via an automobile gateway, which increases the attack surface significantly. The lack of network segmentation makes it possible to gain remote access to the MOST network via the telematics module, pivot to the High-Speed CAN and target critical automobile components such as engine control and brakes. Second, the CAN protocol does not employ message encryption and authentication. The lack of message encryption simplifies reverse engineering as well as message interception, modification and fabrication. The lack of message authentication enables a malicious actor with CAN access to inject harmful messages that interfere with or disable any critical automobile system while remaining undetected. Additionally, the message identifier priority feature enables a malicious actor to flood the High-Speed and Low-Speed CANs with high-priority messages to deny service to all the networked components.

The sorry state of automobile security persists as new safety, convenience and comfort components are installed in models every year – soon, autonomous driving systems will be the norm. Automobile manufacturers are reluctant to segment automobile networks and components for reasons of cost, complexity and practicality (mainly maintenance and repairs). The lack of CAN message encryption and authentication persists due to the cost of implementation and computational resources required by individual automobile components.

A feasible solution is to develop attack detection systems that are incorporated in automobiles as add-on components. The attack detection systems would scrutinize CAN messages in real time and report malicious and anomalous traffic to drivers. Eventually, these attack detection systems could inform attack mitigation systems. Real-time attack detection is imperative because there can be no attack mitigation without detection.

The CAN attack detection system described in this work employs long short-term memory (LSTM) networks [10] to monitor automobile CANs, detect attacks and raise alerts in real time. LSTM networks are leveraged because they can learn patterns with long sequences.

A repeatable design framework is presented for constructing and training multiple LSTM networks that learn normal CAN message timing patterns. The design framework lays out the computational resources as well as the data collection and preprocessing and LSTM model development and training steps. The framework enables new LSTM models to be trained and updated for automobiles of different makes, models and years.

Another key contribution is real-time attack detection. The attack detection system leverages a server-client configuration. The server is an inexpensive Raspberry Pi device connected directly to a monitored automobile CAN bus that captures, logs and transmits CAN message traffic via a Wi-Fi network to a client workstation located outside the automobile. The client workstation provides the computational resources needed for real-time attack detection. Trained LSTM models executing on the client workstation process the transmitted CAN messages, identify attacks and send alerts via the Wi-Fi network.

The attack detection system was evaluated using a 2010 Toyota Prius testbed and a fully-operational 2014 Toyota Prius automobile. An attack device was employed to inject random CAN message identifiers at random times. The attack detection results are very good – LSTM model sensitivity ranged from 0.864 to 1.000 and accuracy ranged from 0.980 to 1.000. Sensitivity and accuracy are the most important metrics because LSTM models must recognize normal traffic and detect as many attacks as possible with high accuracy.

2 Interconnected Automobile Network

An automobile network comprises multiple sub-networks with systems that support safety, convenience and comfort [20]. The interconnected sub-networks include the High-Speed CAN, Low-Speed CAN, LIN and MOST network:

- **High-Speed CAN:** A High-Speed CAN connects critical automobile electronic control units (ECUs) such as the drivetrain, power steering, transmission control, instrument cluster, revolutions per minute (RPM) management, engine control and braking systems. A modern automobile may have multiple High-Speed CANs. Because a High-Speed CAN is responsible for critical automobile functionality, it employs a high-speed version of the CAN protocol that operates at bit rates between 500 Kbps and 1 Mbps to support fast and reliable communications [20].
- **Low-Speed CAN:** A Low-Speed CAN connects convenience and comfort components such as ventilated seats, power windows, lights, heat and air conditioning, and door locks. A Low-Speed CAN employs a low-speed version of the CAN protocol that operates at bit rates in the hundreds of Kbps [20]. An On-Board Diagnostics (OBD-II) interface provides direct access to the High-Speed and Low-Speed CANs via a specialized device. The OBD-II interface, which is located by the steering wheel or instrument cluster, is used to obtain diagnostic information required for automobile service and repair.
- **LIN:** A LIN connects ECUs in a Low-Speed CAN to peripheral components such as lights and door locks. The LIN protocol complements the CAN protocol.
- **MOST Network:** A MOST network connects multimedia components such as an infotainment system and cellular, Bluetooth and Wi-Fi modules in a ring network topology [20]. Telematics service providers such as OnStar interact with a MOST network via its cellular module.

3 Attack Vectors, Vulnerabilities and Attacks

This section lists the attack vectors that target automobile CANs. Also, it describes CAN vulnerabilities and attacks.

3.1 Attack Vectors

A malicious actor would be interested in accessing a High-Speed CAN because it contains critical automobile components. Several attack vectors can be leveraged to target the High-Speed and Low-Speed CANs.

Figure 1 shows the attack vectors that can be leveraged to access High-Speed and Low-Speed CAN components (targets). The attack vectors include the High-Speed CAN, OBD-II interface, Low-Speed CAN, Wi-Fi module, Bluetooth module, cellular module and infotainment system:

- **High-Speed CAN:** A malicious actor can gain direct access to High-Speed CAN targets via a physical connection to the High-Speed CAN (position 1 in Fig. 1). Upon gaining access to the High-Speed CAN via the physical connection, the malicious actor can gain indirect access to Low-Speed CAN targets via the automobile gateway.
- **OBD-II Interface:** A malicious actor can gain direct access to High-Speed and Low-Speed CAN targets via a physical connection to the OBD-II interface (position 2). This is because the OBD-II interface connects directly to the High-Speed and Low-Speed CANs.
- **Low-Speed CAN:** A malicious actor can gain direct access to Low-Speed CAN targets via a physical connection to the Low-Speed CAN (position 3). Upon gaining access to the Low-Speed CAN via the physical connection, the malicious actor can gain indirect access to High-Speed CAN targets via the automobile gateway.
- **Wi-Fi Module:** A malicious actor can gain indirect access to High-Speed and Low-Speed CAN targets via a remote connection to the Wi-Fi module in the MOST network (position 4). This is because the MOST network connects to the High-Speed and Low-Speed CANs via the automobile gateway.
- **Bluetooth Module:** A malicious actor can gain indirect access to High-Speed and Low-Speed CAN targets via a remote connection to the Bluetooth module in the MOST network (position 5). This is because the MOST network connects to the High-Speed and Low-Speed CANs via the automobile gateway.
- **Cellular Module:** A malicious actor can gain indirect access to High-Speed and Low-Speed CAN targets via a remote connection to the cellular module in the MOST network (position 6). This is because the MOST network connects to the High-Speed and Low-Speed CANs via the automobile gateway.
- **Infotainment System:** A malicious actor can gain indirect access to High-Speed and Low-Speed CAN targets via a physical connection to the infotainment system in the MOST network (position 7). This is because the MOST network connects to the High-Speed and Low-Speed CANs via the automobile gateway.

3.2 Vulnerabilities

CAN vulnerabilities arise from the lack of message authentication and message encryption, message identifier priority feature and absence of network segmentation:

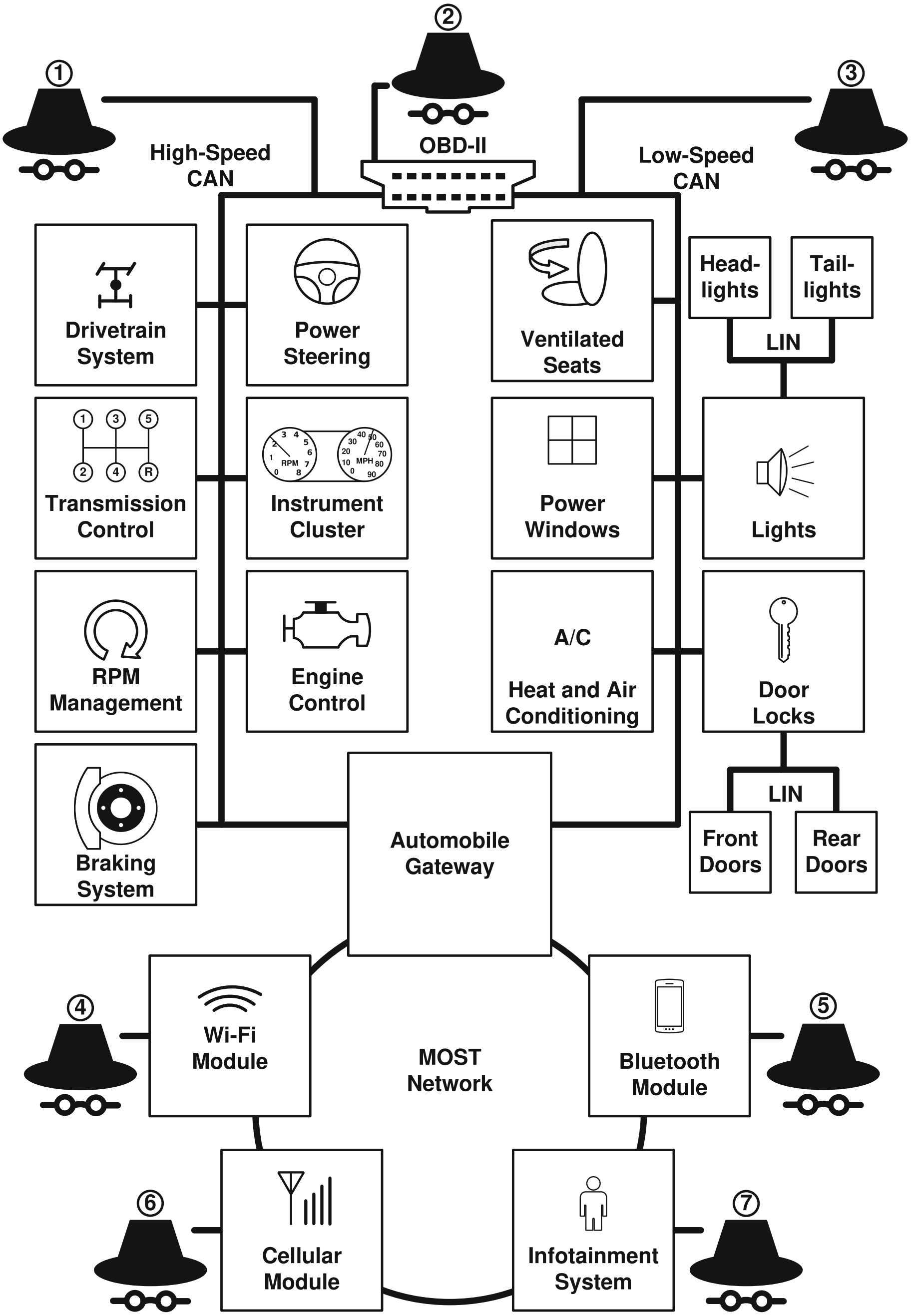


Fig. 1. High-Speed and Low-Speed CAN attack vectors.

- **Message Authentication:** The CAN protocol lacks message authentication. A CAN message only contains an identifier without source and destination addresses [14]. Therefore, receiving nodes cannot verify message source and distinguish between real and fake messages. Thus, a malicious actor with CAN access can transmit fake messages without being detected.
- **Message Encryption:** The CAN protocol lacks message encryption. All CAN messages are transmitted in plaintext [14]. Connecting to a CAN bus directly or via its OBD-II interface provides direct access to all CAN messages. The lack of message encryption simplifies reverse engineering as well as message interception, modification and fabrication. Although automobile manufacturers keep CAN message content proprietary, thorough analysis of CAN traffic can reveal message details. The lack of encryption also enables CAN message replay.
- **Message Identifier Prioritization:** A message identifier with a low binary value has high priority on a CAN bus. A malicious actor can flood a CAN bus with high priority messages to prevent the transmission of legitimate messages. Such denial-of-service attacks are easy to execute and can render critical automobile systems non-operational [14].
- **Network Segmentation:** CANs are not segmented adequately. Nodes in different CANs can communicate with each other via the automobile gateway. A malicious actor with access to an automobile gateway can target nodes in all the connected CANs [14].

3.3 Attacks

Numerous attacks have been demonstrated on CANs in modern automobiles. Absent custom security measures, these attacks are expected to impact practically every High-Speed and Low-Speed CAN.

Hoppe and Dittman [11] describe novel attacks on simulated CANs. They employed CANoe software to create a virtual network comprising connected High-Speed and Low-Speed CAN buses. The CAN buses were connected to a virtual automobile power window system. In one experiment, they captured CAN message frames on the CAN buses and recorded the message frames that opened and closed the power window; the recorded frames were subsequently replayed to control the power window. In another experiment, they used an ECU in a Low-Speed CAN bus to obtain information from the High-Speed CAN bus, demonstrating the lack of segmentation in the virtual CANs. These experiments stimulated research in automobile security.

Koscher et al. [13] employed a custom CarShark CAN bus analyzer and packet injection tool to perform experiments with stationary and moving automobiles. CarShark was used to sniff CAN frames and the message identifiers were subsequently reverse engineered via fuzzing techniques. CAN messages were then injected to control the radio, instrument cluster, engine components, brakes, heating, ventilation and air conditioning, and body control module functions. Denial-of-service attacks were successfully executed on the engine control module of a stationary automobile. Several attacks were executed on a moving automobile, including sounding the horn, killing the engine and preventing the automobile from restarting, and disabling the brakes. The experiments demonstrated that a malicious actor with physical CAN bus access could wreak havoc on stationary and moving automobiles.

Hoppe et al. [12] describe experiments involving a CAN testbed with an electric window lift, instrument cluster, automobile gateway, warning lights and airbag control system. They used a laptop connected to the testbed via the OBD-II interface to directly access the CAN. Fabricated messages were transmitted from the laptop to tamper with various CAN systems.

Checkoway et al. [6] demonstrated several remote exploits that leveraged automobile mechanic tools, media interfaces and wireless communications. In particular, they used the OBD-II interface and infotainment system to obtain indirect physical network access, and Bluetooth and cellular channels to access automobile systems. The research demonstrated that the attack surface expands considerably as an automobile becomes more connected.

Valasek and Miller [24] leveraged custom tools to interact with automobile networks in a 2010 Ford Escape and 2010 Toyota Prius. The automobiles were targeted by connecting a laptop to the OBD-II interfaces. CAN messages were captured and replayed. Also, messages were modified and injected to control the behavior of the automobiles.

Valasek and Miller [25] also demonstrated physical and remote attacks on a 2014 Jeep Cherokee. They targeted its Harman Kardon Uconnect infotainment system that bundles Wi-Fi connectivity, navigation, apps and cellular communications. Specifically, they gained physical access to the infotainment system via a USB connection and were able to jailbreak the system. Next, they gained remote access to the telematics system and exploited it by leveraging an open diagnostics port in the CAN. Using the diagnostics port access, they uploaded modified firmware to the microcontroller connected to the CAN. The resulting direct access to the High-Speed and Low-Speed CAN buses enabled them to send commands to several critical systems. A viral video [8] shows Valasek and Miller remotely turning on the air conditioner of a moving Jeep Cherokee, activating wiper fluid release and even disabling the brakes. The research of Valasek and Miller led Fiat Chrysler to recall 1.4 million automobiles in 2015 [16].

4 Related Work

This section discusses LSTM networks and their applications to CAN attack and anomaly detection.

4.1 LSTM Networks

Creating an anomaly-based detection model for automobile CANs requires a neural network that can learn normal network traffic patterns. This research employs a type of recurrent neural network called a long short-term memory (LSTM) network.

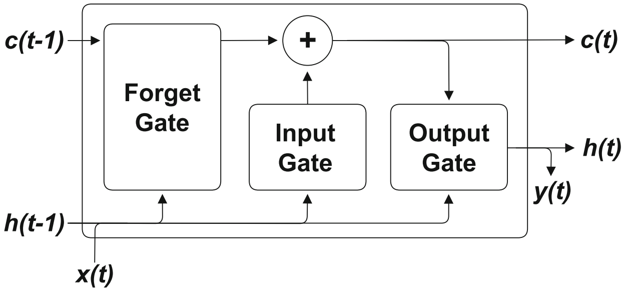


Fig. 2. LSTM cell.

A recurrent neural network comprises connected artificial neurons called cells. Each cell processes inputs using a mathematical function and produces outputs that are sent as inputs to other connected cells. Learning is built into the recurrent neural network cell in the form of a feedback loop [5]. The loop enables information to be passed to the next iteration (time step) of the network loop.

Unfortunately, traditional recurrent neural networks can only learn short patterns. As data propagates through a recurrent neural network, information about older data is discarded. This is problematic when attempting to construct a model that attempts to learn long sequences [7]. Traditional recurrent neural networks have trouble learning patterns in applications such as speech recognition, language translation and network traffic analysis.

Hochreiter and Schmidhuber [10] introduced LSTM networks to address the long-term memory limitation of traditional recurrent neural networks. Their design uses gates within cells that allow certain data to flow through cells. Each cell also has a long-term state called the cell state and a short-term state called the hidden state. These innovations enable an LSTM network to learn patterns with long sequences.

Figure 2 shows a single LSTM cell. All the cell inputs and outputs are vectors. At time step t, input $x(t)$ and the previous hidden state $h(t-1)$ are fed to the cell, which outputs $y(t)$ and the updated hidden state $h(t)$. The horizontal line on the top of the cell (running between $c(t-1)$ and $c(t)$) denotes the cell state, which gives the LSTM cell its long-term memory capability. In an LSTM network implementation, the cell state runs through an entire chain of cells and is updated as needed. Information is added to and removed from the cell state using gates. A cell has three types of gates, forget gate, input gate and output gate:

- **Forget Gate:** A forget gate decides how much information in inputs $x(t)$ and $h(t-1)$ is discarded from the cell state. The output updates the cell state.
- **Input Gate:** An input gate decides how much information from inputs $x(t)$ and $h(t-1)$ is added to the cell state. The output is added to the cell state.

- **Output Gate:** The output gate decides how much information should be read from the cell inputs and cell state, and produces the cell outputs $h(t)$ and $y(t)$. The output yields the cell outputs $h(t)$ and $y(t)$.

The cell state enables an LSTM cell to store important inputs for a period of time. The forget gate in an LSTM cell determines how much information should be discarded. The input gate enables an LSTM cell to learn the important inputs. The output gate produces the LSTM outputs at a specific time step.

An LSTM cell is the fundamental building block of an LSTM network. Each layer in an LSTM network contains tens to hundreds of LSTM cells. The number of layers in an LSTM network depends on the complexity of the problem.

4.2 Attack and Anomaly Detection

Several researchers have employed LSTM networks to detect attacks and anomalies in automobile CANs. The approaches differ in their LSTM model architectures and features used to detect attacks and anomalies.

Taylor et al. [23] were the first to employ an LSTM model to detect attacks leveraging CAN message data field values. An LSTM model was trained to predict data in the next message corresponding to a given CAN identifier. The trained model recognized fabricated CAN messages as anomalous and indicators of attacks.

Zhu et al. [28] developed a multi-dimensional LSTM model for detecting anomalies in CANs. They combined the values in the CAN timestamp and data fields to produce samples with a single feature that were used to train an LSTM model to identify anomalous CAN traffic. The trained model was positioned in a mobile edge computing server to collect and analyze CAN traffic.

Xiao et al. [26] developed a convolutional LSTM model to examine spatiotemporal relationships in CAN traffic. The model was trained using samples with CAN timestamp and data field values. Correlation coefficients between predicted data and real data were computed. A specific range of correlation coefficient values indicated anomalous behavior. Tariq et al. [22] also developed a convolutional LSTM model for predicting normal and abnormal CAN message sequences. Their model is similar to the model of Xiao et al. [26], but it focused on transfer learning, which enabled it to detect novel attacks based on knowledge about previously-seen attacks.

Yang et al. [27] developed an LSTM model to learn the fingerprints of analog CAN signals emanating from ECUs. Their approach differed from others in that it examined CAN message fields. The LSTM model, which was trained using analog CAN signals from ECUs as they processed messages, was implemented using field-programmable gate array hardware for real-time detection.

Hanselmann et al. [9] developed an unsupervised anomaly-based intrusion detection system using LSTM models. An LSTM model was assigned to each CAN identifier and each model was trained to learn the temporal features associated with its CAN identifier. The outputs of all the LSTM models were input to an autoencoder, which produced an anomaly score.

Sun et al. [21] developed a convolutional LSTM model with attention for anomaly-based detection in CANs. The model incorporated a one-dimensional convolution layer for feature extraction and a bidirectional LSTM layer for time characteristic learning in the forward and backward directions. The model was trained using analog CAN signals instead of CAN message fields. Anomaly-based detection was tested on a scaled-down CAN with three ECUs.

Aldhyani and Alkahtani [1] developed a convolutional LSTM model for classifying attacks on automobile CANs. The model, which was trained using CAN message timestamp, identifier, data length and data fields, classified CAN traffic as spoofing, flooding, replay or benign.

Several research efforts have trained LSTM models to learn traffic sequence patterns using the CAN message identifier and data fields, but not the timestamp field. Four efforts stand out as exceptions. Hanselmann et al. [9] focused on learning temporal features of CAN identifiers, but they developed an LSTM model for each CAN identifier, which resulted in a large system. Xiao et al. [26] considered the CAN message timestamp field in addition to other fields, but little information is provided about their implementation. Zhu et al. [28] combined one-bit timestamp and 64-bit data fields. Aldhyani and Alkahtani [1] also combined the timestamp with other features.

Most of the related research efforts did not focus on live systems that monitored CANs in operating automobiles. Three efforts are exceptions. Yang et al. [27] implemented their model using field-programmable gate array hardware for real-time detection. Zhu et al. [28] proposed a mobile-edge computing architecture for real-time detection. However, neither Yang et al. nor Zhu et al. tested their systems on live CAN traffic. The work of Sun et al. [21] stands out because their model was tested on an automobile CAN, although it incorporated only three ECUs.

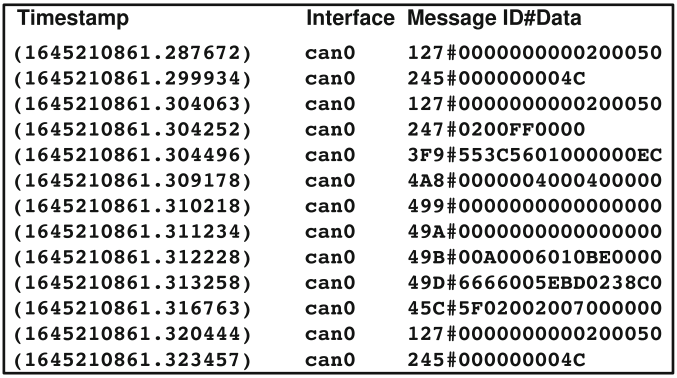
The anomaly-based CAN attack detection approach described in this work advances previous research by focusing on message timing in live automobile CAN traffic. An unsupervised machine learning framework is employed to train LSTM models to recognize normal CAN traffic patterns based on the timestamp and identifier fields. The resulting LSTM models identify mistimed CAN messages as attack indicators. Other unique features of the approach are real-time attack detection in operating automobiles and adaptability to multiple automobile CANs.

5 Attack Detection Design Framework

The attack detection design framework covers the five steps in the machine learning workflow: data collection, data preprocessing, model development and training, model testing, and model enhancement and deployment.

5.1 Data Collection

Tens of thousands of CAN data samples are required to develop LSTM networks for detecting attacks in automobile CANs. During the research, CAN data was



```
Timestamp                   Interface  Message ID#Data
(1645210861.287672)         can0       127#0000000000200050
(1645210861.299934)         can0       245#000000004C
(1645210861.304063)         can0       127#0000000000200050
(1645210861.304252)         can0       247#0200FF0000
(1645210861.304496)         can0       3F9#553C5601000000EC
(1645210861.309178)         can0       4A8#0000004000400000
(1645210861.310218)         can0       499#0000000000000000
(1645210861.311234)         can0       49A#0000000000000000
(1645210861.312228)         can0       49B#00A0006010BE0000
(1645210861.313258)         can0       49D#6666005EBD0238C0
(1645210861.316763)         can0       45C#5F02002007000000
(1645210861.320444)         can0       127#0000000000200050
(1645210861.323457)         can0       245#000000004C
```

Fig. 3. Sample CAN log file.

collected by connecting a Raspberry Pi with a PiCAN interface board to the monitored CAN bus and logging the data.

The Raspberry Pi device captured CAN traffic logs. The `can-utils` package, specifically its `candump` utility [20], facilitated the logging of CAN messages. The Raspberry Pi was set up with the `can0` SocketCAN interface. SocketCAN provides CAN drivers as network devices in a Linux operating system [20]. Application access to the CAN bus was enabled by a network socket programming interface. The `can0` interface provided direct access to the connected automobile CAN bus.

The CAN log files were saved to Google Drive for data preprocessing. Figure 3 shows a sample CAN log file. Each row depicts a single message broadcasted on the connected CAN bus. The first column lists the timestamps, the absolute times connected to the Raspberry Pi system clock. The second column lists the `can0` interface. The third column specifies the CAN message identifiers and associated data. The CAN message identifier is the hexadecimal string to the left of the hash symbol and the data field is the hexadecimal string to the right of the hash symbol. The data field size is between one and eight bytes.

5.2 Data Preprocessing

CAN log data must be preprocessed before it can be used for training and testing. The CAN message timestamp and identifier were selected as features for developing LSTM network models. The feature values are shown in the first column and the left half of the third column in Fig. 3. The CAN data field was not used because attack detection focuses on the timing patterns of CAN messages.

The `pandas` Python library was primarily used for data preprocessing [17]. The library uses high-level data structures called DataFrames and various methods to simplify data conversion. A DataFrame is a tabular data structure with an ordered collection of columns, potentially with different types. Several built-in methods were employed to manipulate DataFrames during data preprocessing.

The CAN log file was read and the feature data was stored in a pandas DataFrame using the read_csv method. Only the CAN message timestamp and identifier features were considered in this research. The timestamp feature was converted from a string to numeric value using the to_numeric method.

The LSTM networks were trained using Δ timestamp values corresponding to CAN identifiers. A Δ timestamp value denotes the frequency at which a CAN identifier is transmitted on a CAN bus. The composite groupby(df.ID).diff method computes the difference between the current and previous timestamps of a given CAN identifier. The difference value replaces the timestamp in the timestamp column in the DataFrame. Because there is no previous timestamp before the first row of a respective CAN identifier, null values are stored in the first differenced rows. The dropna method eliminates rows with null values from the DataFrame.

After the data is loaded in a DataFrame, groups of CAN messages must be binned into smaller DataFrames based on the frequency of CAN identifiers. This is necessary because of potential biasing in a machine learning model. Specifically, CAN message identifiers that appear more frequently in a log file induce bias during model training. Biasing causes a model to make incorrect assumptions, which hinders effective machine learning [7].

The k-means clustering algorithm [15] was used to place CAN messages into bins based on their frequency. The algorithm was selected because it quickly and efficiently clusters datasets [7]. The k-means algorithm clustered the data samples into bins based on the timestamp feature in a loaded DataFrame. A separate DataFrame was subsequently created for each bin.

Figure 4 shows an example of the binning process. A DataFrame is accepted as input to the k-means clustering algorithm. Three bins are generated as outputs by the algorithm. The highest frequency messages are stored in Bin 0, medium frequency messages in Bin 1 and lowest frequency messages in Bin 2. Each bin is treated separately throughout the rest of the design process and a separate LSTM model is trained using each bin.

Feature data is required to be in a numeric format [5]. However, the CAN identifier feature values are hexadecimal strings that correspond to categorical data (i.e., labels that describe their semantics). Therefore, the CAN identifiers were converted to numeric values using an integer encoding that gives a unique integer value to each CAN identifier. Conversions of timestamp feature values were not required because timestamps have a numeric format.

Feature data must be scaled [5]. This was accomplished by normalizing the numeric CAN identifier feature values between zero and one. The timestamp feature values were rescaled so that the mean of the values was zero and standard deviation was one. This was done because the timestamp feature values had a well-behaved mean and standard deviation. The CAN identifier feature values did not have this property, which is why they were normalized.

The data conversion process invokes the LabelEncoder, MinMaxScaler and StandardScaler methods in the Scikit-Learn library [7]. The input is the DataFrame with the CAN timestamp and identifier features. The CAN identifier

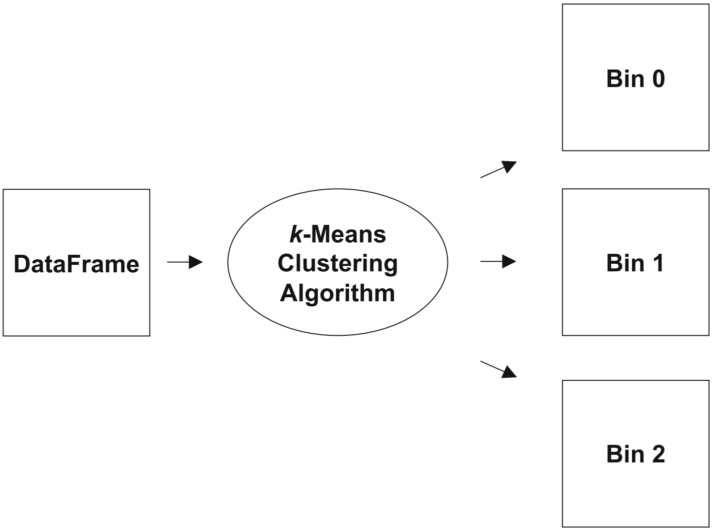


Fig. 4. Binning process.

feature values are encoded as integers and subsequently normalized. Finally, the CAN timestamp feature values are standardized to produce the converted DataFrame.

Note that the encoder objects created during data conversion must be saved for future use. This is because all data input to the attack detection system must be encoded and scaled in the same format.

LSTM networks require three-dimensional data [7]. Therefore, the final data preprocessing step converts the data to a batch of sequences in three dimensions, batch size, time step and input dimensionality:

- **Batch Size:** The batch size is the number of CAN message sequences input to an LSTM network. When training a machine learning model, the batch size is of the order of tens to hundreds of thousands of sequences.
- **Time Step Size:** The time step is the window size used to train an LSTM network. It represents the memory of an LSTM network – given n time steps, the LSTM network is trained to remember the previous n observations.
- **Input Dimensionality:** The input dimensionality is the number of features used to train an LSTM network. Two features, CAN message timestamp and identifier, are employed to develop the LSTM-based attack detection system in this dissertation research.

Figure 5 shows an example of the sequence creation process. In the example, the time step is five, number of samples is seven and number of features is two. The process employs a sliding window approach, where the time step window moves across the rows of the dataset. Sequence 0 takes the first five samples shown in gray. Sequence 1 moves down one row and takes the next five samples.

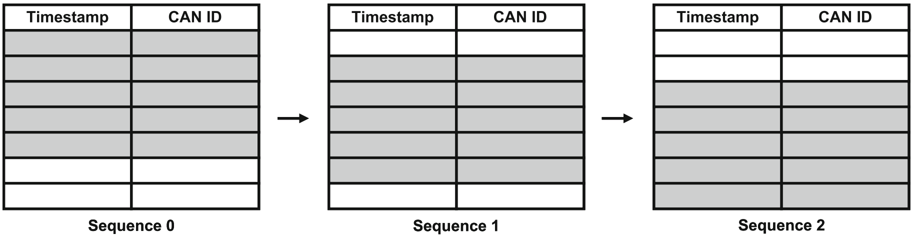


Fig. 5. Sequence creation process.

Sequence 2 moves down another row and takes the next five samples. The input dimensionality in this example is 3 × 5 × 2, which is fed to an LSTM network.

5.3 Model Development and Training

The CAN network attack detection model employs LSTM networks with autoencoder architectures. The LSTM network layers capture temporal relationships between CAN message timestamp and identifier features. The autoencoder architectures capture dense representations of the training data.

An autoencoder learns to copy inputs to outputs using data compression and reconstruction [7]. Specifically, it uses data compression and reconstruction to learn the important characteristics of data. Given CAN data, an autoencoder learns normal CAN traffic patterns.

An autoencoder incorporates an encoder and decoder. The encoder compresses the input data to a fixed-sized vector with less dimensionality. The decoder reconstructs the data from the fixed-size vector of less dimensionality to produce an output. A well-trained autoencoder produces output values close to its input values. However, due to network constraints and data complexity, training an autoencoder is not a trivial task. The autoencoder must learn efficient ways to represent the training data in order to have its outputs resemble its inputs.

Figure 6 shows an LSTM network with an autoencoder architecture. The architecture has two LSTM layers, a repeat vector layer and a time distributed layer.

The first LSTM layer is the encoder. This layer accepts the input with a time step of 15 and input dimensionality of two. It is designed for an arbitrary batch size as input, so the batch size is labeled none. The layer outputs a compressed feature vector of size 1 × 30.

The repeat vector layer serves as the bridge between the encoder and decoder [19]. The layer accepts a feature vector of size 1 × 30 as input and replicates it 15 times. It outputs a 15 × 30 vector that is input to the second LSTM layer.

The second LSTM layer is the decoder layer, which reconstructs the encoding [19]. The layer accepts a 15 × 30 vector as input and outputs a reconstruction of the encoding with size 15 × 30.

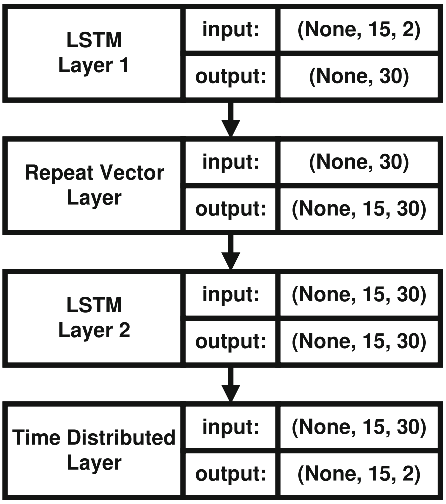


Fig. 6. LSTM network with an autoencoder architecture.

The time distributed layer converts the reconstructed output to the same size as the input. It creates a 30 × 2 output with input dimensionality of two. The output of the second LSTM layer (15 × 30) is matrix-multiplied with the output of the time distributed layer (30 × 2). The resulting output size of 15 × 2 is the same as the input size.

After an LSTM network is constructed, the model must be compiled and fitted. The compilation process, which is a pre-fitting step for the model, converts the sequences of layers in Fig. 6 to matrix transforms that can be executed on central processing units (CPUs), graphics processing units (GPUs) or tensor processing units (TPUs) [5].

A constructed LSTM model is compiled using the optimizer and loss function parameters. The optimizer is an algorithm that updates the model weights during training. Adaptive moment estimation, a popular optimization algorithm due to its overall performance [5], was employed. Additionally, the algorithm does not require constant tuning.

The loss function determines the performance of an LSTM model. Since a model predicts numerical values based on given input, it attempts to solve a regression problem. The mean-squared error (MSE) loss function employed is given by:

$$\text{MSE loss} = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{n}$$

where x_i is the model input, y_i is the predicted model output and n is the number of data samples over which the loss is computed.

An LSTM model is fitted after the compilation step. This is the step where the LSTM model is trained. To solve the regression problem, training data is fed to the model and the weights are adjusted to make the model fit the training data to the desired extent [5].

The following command from the Keras ML library [7] fits a constructed LSTM model:

```
model.fit(x, y, epochs=20, batch_size=15, validation_split=0.2)
```

where x is the preprocessed input training data and y is the expected output. This is equivalent to an autoencoder architecture because the model is trained to make the outputs close to or equal to the inputs.

An epoch corresponds to one pass through the entire training dataset. The batch size specifies how many samples the model processes before the weights are updated.

The validation split holds out a portion of the training data to create the validation dataset. The model.fit command holds out 20% of the training data for validation to improve the problem generalization ability of the LSTM model [7]. The validation dataset is used during the training phase to tune the model.

Model learning curves created during training provide visual indications of how the LSTM model is learning [5]. The learning curves plot the MSE losses with the training and validation datasets during model training. Most well-trained models have curves with exponential decays. Training must be terminated when the training and validation curves taper off because overtraining can cause problems. It is also important to repeat the training process multiple times to verify that the model works well. Neural networks are stochastic, meaning that different predictions are made when the same model is trained on the same data over multiple runs [5].

Data pertaining to a trained model is saved in two files for use in the final two steps of the machine learning workflow, model testing, and model enhancement and deployment. The model architecture is saved in a JSON file. The model weights are saved in an HDF5-formatted file.

5.4 Model Testing

A trained model is tested and verified using unseen CAN data. Data in the testing dataset is preprocessed in the exact same way as the training data in order to enhance accuracy.

Table 1 shows the files created for model testing that are subsequently updated after model enhancement. The model architecture and weights are loaded from the saved JSON and HDF5 files. The encodings as well as the standardizing and normalizing parameters are loaded from the saved PKL files.

The maximum MSE loss during testing is computed as follows. Each test data sequence passes through the LSTM model and an output sequence is generated. The MSE loss is computed between the input and output sequences. The maximum MSE loss is output at the end of the loop. A model with good generalization ability has a low maximum MSE loss, similar to the MSE loss of the training data.

Table 1. Files created for model testing and updated after model enhancement.

File	Description
`model.json`	Stores the model architecture
`weights.h5`	Stores the model weights
`standardizeScaler.pkl`	Stores the standardizing parameters for the timestamp feature
`labelEncoder.pkl`	Stores the integer encodings of the CAN identifier feature
`normalizeScaler.pkl`	Stores the normalizing parameters for the CAN identifier feature

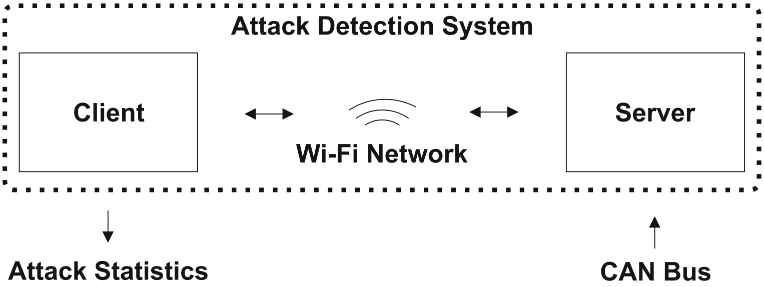


Fig. 7. Attack detection system configuration.

5.5 Model Enhancement and Deployment

A trained and tested LSTM model can always be enhanced. Model enhancement starts with the LSTM model files produced after testing and proceeds to update the LSTM model files for deployment in a production environment. Model enhancement and deployment require updated versions of the files used for testing (Table 1).

6 Real-Time Attack Detection System

This section describes the deployment of the LSTM models in the real-time CAN attack detection system.

6.1 System Configuration

The attack detection system leverages a server-client configuration on a monitored automobile CAN bus. This configuration eliminates the need to have a large system connected to the CAN bus and supports remote attack detection. The attack detection system uses commercial off-the-shelf components.

Figure 7 shows the attack detection system configuration. The server is connected to the monitored CAN bus. The server and client are connected via a

Wi-Fi network. The client processes CAN data received from the server and presents attack statistics.

Server Configuration. The server is an embedded system that is designed to connect to an automobile CAN bus. The embedded hardware was chosen for its lightweight, powerful computing and low cost features.

The server executes on a Raspberry Pi 4 Model B running the 64-bit Raspberry Pi OS Lite. The Raspberry Pi incorporates a PiCAN3 board, which contains an MCP2562 CAN transceiver, MCP2515 CAN controller and 3 A switch mode power supply board. The Raspberry Pi is mounted inside the automobile to be monitored for CAN attacks.

Client Configuration. The client is a rack-mounted computer workstation located outside the automobile being monitored. The workstation was selected to provide the computational resources and processing speed needed for real-time attack detection. The client executes on a Macintosh Pro Rack version 2019 running an Ubuntu virtual machine. The virtual machine is used for all client command and control.

6.2 System Processes

Real-time attack detection involves CAN message logging, processing and prediction. Python scripts were written for the server and client to implement the necessary tasks.

Server Process. The server is responsible for receiving CAN messages, storing the messages in queues and handling client message requests. CAN messages are split into queues depending on the number of LSTM models trained for the CAN bus. A queue is reserved for each LSTM model (bin) created during the model training phase. The server transmits CAN messages from relevant queues upon requests from the client. The server process includes the CAN message handling subprocess and client handling subprocess.

The CAN message handling subprocess determines the number of bins and the CAN messages that go in the bins. Each bin corresponding to an LSTM model contains the CAN identifiers associated with a queue. The server listens for messages on the CAN bus. When a CAN message is received, the difference between the timestamp of the current message and previous message with the same CAN identifier is computed. Following this, the CAN message is placed in the appropriate queue.

The client handling subprocess handles the queues generated by the CAN message handling subprocess. The server listens for a message request from the client. When a request is received from the client, the server parses the request to determine the number of requested messages and the queue in which they exist. If the number of messages requested is greater than the number of messages in the queue, the server waits for the queue to hold the requested number of messages. The server then sends the requested messages to the client.

The CAN message and client handling subprocesses operate concurrently. This ensures that no CAN messages or message requests are dropped. The queues contain data that is shared by the CAN message handling and client handling subprocesses.

Client Process. The client is responsible for examining the received CAN messages and detecting attacks. First, the client requests a set of messages from the server. After receiving the messages, the client preprocesses the messages and feeds them to the appropriate LSTM model. The LSTM model compresses and reconstructs the messages, and computes the mean-squared error. The client process includes the client message requesting and message prediction subprocesses.

An attack detection system user specifies the number of iterations of requests at client startup. The client message requesting subprocess sends a request to the server that includes the number of messages requested and LSTM model (bin number) for processing the messages. The number of requested messages is equal to the time step in the corresponding LSTM model.

Messages received from the server are encoded and scaled for input to the LSTM model. This is an important step because all the messages have to be encoded and scaled in exactly the same manner as the training data. The encoding and scaling files saved after LSTM model training are processed to produce tensors of transformed data. A tensor is a three-dimensional vector of size batch size × time step × input dimensionality. The batch size is one, the time step is dependent on the LSTM model and the input dimensionality is two (CAN timestamp and identifier features). The tensor is placed in a shared data queue for the client message prediction subprocess.

An attack detection system user may specify the number of iterations required for attack prediction at client startup. Alternatively, the client message prediction subprocess can run continuously. The client message prediction subprocess receives a tensor of messages from the shared queue. The tensor is passed to the appropriate LSTM model and returns a reconstructed tensor. Finally, the client message prediction subprocess computes the MSE loss between the input and output tensors.

The client message requesting and prediction subprocesses execute concurrently to support real-time attack detection. The tensor queue is shared by the two subprocesses.

6.3 Client Operation Modes

While the server has a single operation mode, the client has three modes, threshold testing mode, attack detection mode and default execution mode:

- **Threshold Testing Mode:** This mode enables an attack detection system user to determine the appropriate MSE loss threshold for detecting attacks.

The client executes thousands of iterations. Upon completing the iterations, statistics are printed for a user to determine an appropriate MSE loss threshold for attack detection.

- **Attack Detection Mode:** This mode is used for attack detection statistics generation after threshold testing. An attack is indicated when the MSE loss value is greater than the set threshold. The attack detection statistics include the true-positive, false-positive, true-negative and false-negative error rates.
- **Default Execution Mode:** This mode alerts an attack detection system user to attacks. It does not print attack detection statistics.

Multiple client processes must execute concurrently to analyze all the messages transmitted on a CAN bus. A separate client process executes for each LSTM model used in attack detection. A bash script initiates a client process for each model. The client processes leverage the central processing unit cores on the Macintosh Pro workstation to run the LSTM models concurrently.

7 Experimental Testbeds and Results

This section describes the experimental testbeds and the results of evaluating the performance of the real-time attack detection system.

7.1 Experimental Testbeds

Two experimental testbeds were employed to evaluate the performance of the real-time attack detection system for CANs, a 2010 Toyota Prius testbed and a fully-operational 2014 Toyota Prius automobile.

2010 Toyota Prius Testbed. Figure 8 shows the 2010 Toyota Prius testbed. The testbed comprises a CAN test bench with ECUs and a Raspberry Pi attack device (mounted on the wooden board on the table), a black Raspberry Pi server for attack detection (just to the left of the test bench on the table) and a Macintosh Pro Rack version 2019 client running an Ubuntu 22.04 LTS virtual machine using VMware Fusion 12 Pro (on the floor).

The test bench comprises a CAN bus connecting four ECUs from a wrecked 2010 Toyota Prius. The ECUs include the smart key, transmission control, power management control and instrument cluster modules. The accelerator pedal and gear shift mechanism are connected to the power management control and transmission control modules. The Raspberry Pi on the bottom-left of the testbed is the attack device. The black Raspberry Pi just to the top-left of the test bench is the attack detection server. The Raspberry Pi attack device and the Raspberry Pi attack detection server are connected directly to the High-Speed CAN bus in the testbed.

Table 2 specifies the 2010 Toyota Prius test bench LSTM model architecture. The architecture comprises four LSTM models trained to analyze the message

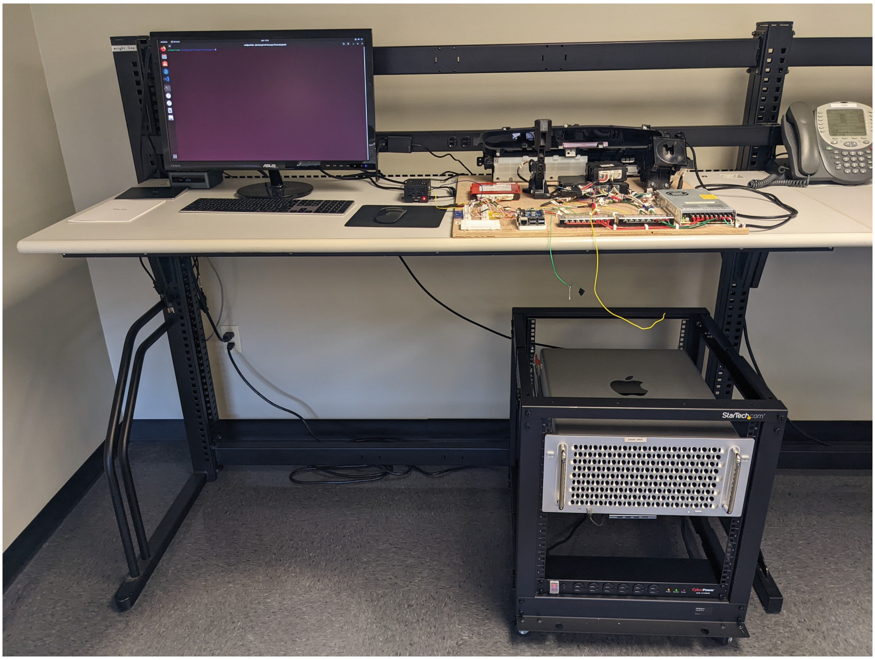


Fig. 8. 2010 Toyota Prius testbed.

timing patterns of the 28 CAN message identifiers listed in the table. Four models were constructed for the 28 CAN message identifiers to prevent model bias towards higher frequency identifiers. CAN message identifiers 612, 613, 616, 619 and 61A were not included due to their low transmission frequencies.

The time step specifies the memory window used by an LSTM model. The LSTM cell configuration corresponding to each model specifies the number of cells used in the LSTM encoder and decoder layers. Each LSTM model has a repeat vector layer between the LSTM encoder and decoder layers, and a time-distributed layer positioned after the LSTM decoder layer. The epochs and batch sizes used for training the LSTM models are also listed.

2014 Toyota Prius Automobile. A fully-operational 2014 Toyota Prius automobile with all the connected ECUs was also employed in the experimental evaluation. Figure 9 shows the Raspberry Pi attack device located below the steering column (left) and the Raspberry Pi server for attack detection located on the center console (right). The attack device is connected directly to the High-Speed CAN bus via the OBD-II diagnostics interface whereas the server is connected to the High-Speed CAN bus via the twisted pair. The Macintosh Pro Rack version 2019 in the 2010 Toyota Prius testbed is also used as the attack detection client for this testbed.

Table 2. 2010 Toyota Prius testbed LSTM model architecture.

LSTM Model	CAN Identifiers	Time Step	Cell Config.	Epoch	Batch Size
0	127, 245, 247	5	10–10	5	5
1	3F9, 6C0, 45C, 45F, 442, 44D	10	20–20	10	10
2	4A8, 499, 49A, 49B, 49D, 3B3, 610	10	20–20	10	10
3	630, 632, 633, 635, 399, 3BB, 3BC, 4A6, 421, 3B6, 611, 3BD	15	30–30	20	15

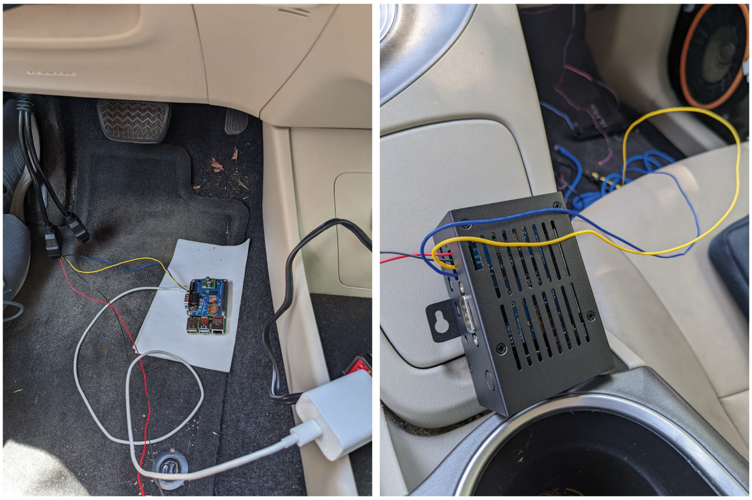


Fig. 9. Attack device (left) and attack detection server (right).

Table 3 specifies the 2014 Toyota Prius automobile LSTM model architecture. The architecture comprises six LSTM models trained to analyze the message timing patterns of the 79 CAN message identifiers listed in the table. Six models were constructed for the 79 CAN message identifiers to prevent model bias towards higher frequency identifiers. CAN message identifiers 383, 381, 382, 3B6, 612, 613, 616, 619 and 61A were not included due to their low transmission frequencies.

The time step corresponds to the memory window used by an LSTM model. The LSTM cell configuration corresponding to each LSTM model specifies the number of LSTM cells used in the LSTM encoder and decoder layers. Each LSTM model has a repeat vector layer between the LSTM encoder and decoder layers, and a time-distributed layer positioned after the LSTM decoder layer. The epochs and batch sizes used for training the LSTM models are also listed.

Table 3. 2014 Toyota Prius automobile LSTM model architecture.

LSTM Model	CAN Identifiers	Time Step	Cell Config.	Epoch	Batch Size
0	0AA, 127, 020, 025, 024, 245, 260, 1C4, 224, 247, 230, 0B4, 235	15	30–30	10	15
1	1AA, 32A, 320, 0B6, 262, 361, 351	10	20–20	15	10
2	6C0, 3F9, 394, 3B7, 620	5	10–10	20	5
3	63B, 4A0, 4A1, 4A2, 610, 4A8, 499, 49A, 49B, 49D, 4A7, 498, 49C, 3B3, 3D3	20	40–40	50	20
4	44D, 45C, 45F, 440, 442, 443	10	40–40	50	10
5	4A6, 4C1, 3B0, 626, 611, 3B1, 420, 4C8, 423, 621, 622, 624, 638, 639, 680, 3B9, 4C3, 3BC, 38E, 4C7, 387, 4C6, 38F, 4DD, 3BD, 3BB, 630, 399, 632, 421, 42F, 633, 635	35	200–200	100	35

All the parameters are the same as those used in the LSTM models for the 2010 Toyota Prius testbed.

Test Environment

Figure 10 shows the test environment. The attack device and attack detection server are connected to the monitored CAN bus. The client and server are connected via a Wi-Fi network. During testing, random synthetic attacks were injected by the attack device into the monitored CAN bus.

The synthetic CAN message injection process implemented by the attack device executes for a duration specified in seconds. This process chooses a random CAN message identifier from a specified bin. If no bin number is specified, then the process chooses a bin at random to obtain a CAN message identifier. The attack device injects the CAN message identifier along with a flag that

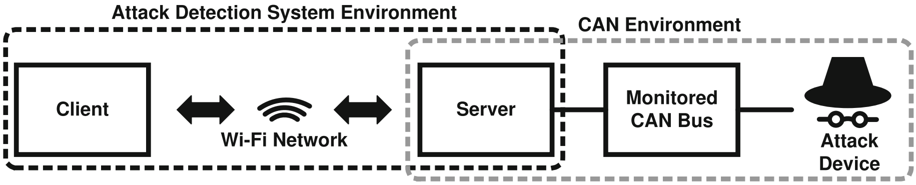


Fig. 10. Test environment.

indicates the absolute truth corresponding to an attack. At this point, the process displays that a particular message was injected along with the bin associated with its identifier. The process then waits for a randomly-generated time in seconds between a specified lower bound and an upper bound.

Table 4. Attack detection data.

MSE Loss Above Threshold	Flag Set	Result
Yes	Yes	TP
Yes	No	FP
No	No	TN
No	Yes	FN

The attack detection system uses the attack detection mode to raise alerts about attacks and collect historical real-time attack data. Table 4 shows the results generated by the attack detection system for High-Speed CAN message sequences. A true positive (TP) output is received when the MSE loss is above the set threshold and a CAN message is received with an attack flag set. A false positive (FP) output is received when the MSE loss is above the set threshold and a CAN message is received with no attack flag set. A true negative (TN) output is received when the MSE loss is below the set threshold and a CAN message is received with no attack flag set. A false negative (FN) output is received when the MSE loss is below the set threshold and a CAN message is received with an attack flag set. When the attack detection system is terminated, the historical real-time attack data is presented to the user.

Note that error propagation often occurs when an attack message is injected. Specifically, when an attack occurs, two message sequences that result in MSE loss above the set threshold are received. The first corresponds to the attack message sequence (true positive) and the second is the next message sequence, which is recognized as part of the attack.

7.2 Experimental Results

This section presents the real-time attack detection results. The true positive, false positive, true negative and false negative detection values are used to compute the attack detection performance metrics.

Performance Metrics. Five metrics are employed to characterize attack detection system performance, precision, sensitivity, specificity, accuracy and F1 score.

The precision metric P describes how well LSTM models detect actual attacks relative to the total number of predicted attacks:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

The sensitivity (recall) metric Se describes the ability of LSTM models to correctly determine attacks:

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

The specificity metric Sp describes how well LSTM models recognize normal activity without attacks:

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

The accuracy metric A describes how well LSTM models can determine all observations correctly:

$$\mathrm{A} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

The F1 score metric is the harmonic mean of precision and sensitivity (recall):

$$\mathrm{F1} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

While it is desirable for all five metrics to be high during testing, sensitivity and accuracy are the principal metrics for evaluating LSTM models. This is because it is most important that LSTM models detect as many attacks as possible with high accuracy.

2010 Toyota Prius Testbed Attack Detection Results. Real-time attack testing was conducted on the 2010 Toyota Prius testbed. During the one-hour test, the attack device injected CAN message identifiers at random (28 different identifiers). The tests evaluated the four LSTM models concurrently using the specified MSE loss thresholds.

Table 5 shows the precision, sensitivity, specificity, accuracy and F1 score for each LSTM model. All the LSTM models, except for LSTM Model 0, achieved the maximum precision, sensitivity, specificity, accuracy and F1 score. LSTM

Table 5. 2010 Toyota Prius testbed concurrent test results.

LSTM Model	MSE Loss Threshold	Messages Analyzed	Attacks Injected	Attacks Detected	P	Se	Sp	A	F1
0	0.005	490,015	192	192	0.985	1.000	1.000	1.000	0.992
1	0.300	62,890	209	209	1.000	1.000	1.000	1.000	1.000
2	1.000	50,770	174	174	1.000	1.000	1.000	1.000	1.000
3	1.000	44,010	205	205	1.000	1.000	1.000	1.000	1.000

Model 0 generated three false positives, which reduced its precision and F1 score. Accuracy and specificity were affected negligibly due to the small number of false positives. LSTM Model 0 achieved the maximum sensitivity.

The MSE loss threshold for LSTM Model 0 could be increased to reduce the number of false positives. Also, the model could be improved by model tuning and retraining, or model restructuring. Overall, the four models were able to distinguish between normal CAN traffic and attack traffic with the 28 CAN message identifiers.

2014 Toyota Prius Automobile Attack Detection Results. Four tests were conducted on the 2014 Toyota Prius automobile. The first and second tests evaluated the six LSTM models concurrently for half-hour periods. These two real-time tests involving CAN attacks were performed when the automobile was stationary for safety and network configuration reasons.

The third real-time test evaluated the six LSTM models concurrently when the automobile was moving forward and reversing. The fourth test, which was not performed in real time, evaluated the six LSTM models separately under normal driving conditions. No attacks were launched during the third and fourth tests for safety reasons.

Table 6 shows the results of the first test, which evaluated all six LSTM models executing concurrently. During the half-hour test, the attack device injected CAN message identifiers selected at random (79 different identifiers). The concurrency test evaluated the six LSTM models using the specified MSE loss thresholds. The precision, sensitivity, specificity, accuracy and F1 score are presented for each LSTM model.

In the first test, LSTM Model 0 generated 121 false positives, which reduced its precision, specificity, accuracy and F1 score. Note that the precision (0.142) and F1 score (0.248) are very low. This is due to the decreased rate of true positives caused by the attack device selecting identifiers from random bins instead of identifiers only from the bin associated with LSTM Model 0. Nevertheless, LSTM Model 0 yielded good attack detection accuracy of 0.997. The model achieved maximum attack detection sensitivity at the cost of detecting false positives.

LSTM Models 2 and 4 generated six and two false positives, respectively, which reduced their precision, specificity, accuracy and F1 scores. However, the models achieved the maximum sensitivity. LSTM Models 1, 3 and 5 achieved the maximum precision, sensitivity, specificity, accuracy and F1 scores.

Table 6. 2014 Toyota Prius automobile concurrent test results (Run 1).

LSTM Model	MSE Loss Threshold	Messages Analyzed	Attacks Injected	Attacks Detected	P	Se	Sp	A	F1
0	0.025	707,520	20	20	0.142	1.000	0.997	0.997	0.248
1	0.250	256,620	48	48	1.000	1.000	1.000	1.000	1.000
2	0.500	31,415	43	43	0.878	1.000	0.999	0.999	0.935
3	0.450	55,220	42	42	1.000	1.000	1.000	1.000	1.000
4	0.300	18,420	45	45	0.978	1.000	0.999	0.999	0.989
5	0.100	60,655	46	46	1.000	1.000	1.000	1.000	1.000

Overall, LSTM Models 1 through 5 were able to distinguish between normal CAN traffic and attack traffic with the 79 CAN message identifiers. LSTM model 0 achieved high accuracy for attack detection, but it generated several false positives. The MSE loss threshold for LSTM Model 0 could be increased in an attempt to reduce the number of false positives. Also, LSTM Model 0 could be improved with model tuning and retraining, or model restructuring.

Table 7. 2014 Toyota Prius automobile concurrent test results (Run 2).

LSTM Model	MSE Loss Threshold	Messages Analyzed	Attacks Injected	Attacks Detected	P	Se	Sp	A	F1
0	0.025	703,800	19	19	0.158	0.864	0.998	0.998	0.268
1	0.250	257,200	47	47	0.979	1.000	1.000	1.000	0.959
2	0.750	31,490	48	48	0.941	1.000	1.000	1.000	0.970
3	0.450	55,180	56	56	0.982	1.000	1.000	1.000	0.991
4	0.500	18,470	38	38	1.000	1.000	1.000	1.000	1.000
5	0.100	60,795	35	35	0.854	1.000	0.996	0.996	0.921

Table 7 shows the results of the second test with all six LSTM models executing concurrently. During the half-hour test, the attack device injected CAN message identifiers at random (79 different identifiers). The concurrency test evaluated the six LSTM models using the specified MSE loss thresholds. The precision, sensitivity, specificity, accuracy and F1 score were computed for each LSTM model.

In the second test, LSTM Model 0 generated 101 false positives, which reduced its precision, specificity, accuracy and F1 score. Note that its precision (0.158) and F1 score (0.268) are very low. This is due to the decreased rate of true positives caused by the attack device selecting identifiers from random bins instead of only identifiers from the bin associated with LSTM Model 0. The model generated three false negatives, which reduced its sensitivity to 0.864. However, LSTM Model 0 yielded good attack detection accuracy of 0.998 despite the false positives.

LSTM Models 1, 2, 3 and 5 generated one, three, one and six false positives, respectively, which reduced their precision and F1 scores. Specificity and accuracy were affected negligibly due to the small numbers of false positives. All four

Table 8. 2014 Toyota Prius automobile forward/reverse motion test results.

LSTM Model	MSE Loss Threshold	Messages Analyzed	Sp
0	0.025	40,680	0.998
1	0.250	13,910	0.999
2	0.750	1,635	0.997
3	0.450	3,340	1.000
4	0.500	1,170	1.000
5	0.100	60,795	0.772

models achieved the maximum sensitivity. LSTM Model 4 fared best, achieving the maximum precision, sensitivity, specificity, accuracy and F1 score.

Overall, LSTM Models 1 through 5 were able to distinguish between normal CAN traffic and attack traffic with the 79 CAN message identifiers. LSTM Model 0 achieved high accuracy for attack detection, but generated several false positives. Because the model generated false positives and false negatives, adjusting the MSE loss threshold would likely not improve its performance. However, LSTM Model 0 could be improved with model tuning and retraining, or model restructuring.

Although attacks could not be launched when the automobile was moving, it was important to test the attack detection system under normal operating conditions. Table 8 shows the results of the forward/reverse motion test. During the two-minute test, the automobile moved forward ten feet and reversed ten feet. Because no attacks were launched, only the specificity metric was computed using the true negatives and false positives.

LSTM Models 0, 1, 2 and 5 generated five, one, one and 18 false positives, respectively, negatively affecting their specificity. LSTM Models 3 and 4 achieved the maximum specificity. Overall, the LSTM models were able to recognize normal CAN traffic. LSTM Model 5 likely yielded a lower specificity of 0.772 due to CAN traffic interruptions when engaging the gear shift mechanism.

The final test involved the execution of the six LSTM models under normal driving conditions. CAN traffic was logged while driving around campus and the log file was input to the six LSTM models. Table 9 shows the results of the normal driving test. Because no attacks were launched, only the specificity metric was computed using the true negatives and false positives.

LSTM Models 0, 2, 3 and 5 generated 28, 1, 2 and 27 false positives, respectively, which reduced their specificity. LSTM Models 1 and 4 achieved the maximum specificity. Overall, the six LSTM models were able to recognize normal CAN traffic. Model 5 likely yielded a lower specificity of 0.892 due to CAN traffic interruptions during normal driving conditions.

Table 9. 2014 Toyota Prius automobile normal driving test results.

Model	MSE Loss Threshold	Messages Analyzed	Sp
0	0.025	171,615	0.998
1	0.250	32,670	1.000
2	0.750	4,040	0.999
3	0.450	6,940	0.994
4	0.500	2,330	1.000
5	0.100	7,770	0.892

Potential Model Improvements. For the 2010 Toyota Prius testbed, precision ranged from 0.985 to 1.000, sensitivity, specificity and accuracy were a perfect 1.000, and the F1 score ranged from 0.992 to 1.000. For the 2014 Toyota Prius automobile, precision ranged from 0.142 to 1.000, sensitivity ranged from 0.864 to 1.000, specificity ranged from 0.772 to 1.000, accuracy ranged from 0.980 to 1.000, and the F1 score ranged from 0.248 to 1.000. The low precision and F1 scores were due to high false positive to true positive rates in just the LSTM Model 0 in the 2014 Toyota Prius automobile tests. Nevertheless, the attack detection results are very good. Specifically, sensitivity ranged from 0.864 to 1.000 and accuracy ranged from 0.980 to 1.000 for the LSTM models in the 2014 Toyota Prius automobile tests. Sensitivity and accuracy are the most important metrics because the LSTM models must recognize normal traffic and detect as many attacks as possible with high accuracy.

LSTM model performance could be improved by reducing the false positives and/or false negatives. One approach is to adjust the MSE loss threshold to make the model more or less sensitive; this may be done using a trial-and-error procedure or using a receiver operating characteristic (ROC) curve. Another approach is to redesign the LSTM model or retrain it using additional data and epochs, but this would be more time consuming than MSE loss threshold adjustment. Alternatively, the LSTM model may be divided into smaller models to simplify the message sequences to be learned. This would involve dividing a CAN message identifier bin into smaller bins. Bin subdivision, while effective, would be more time consuming than the MSE loss threshold adjustment and model redesign/retraining approaches.

It is possible that some CAN message identifiers do not have periodic timing patterns. Since an LSTM model cannot not be trained effectively with aperiodic timing patterns, the corresponding CAN message identifiers would have to be discarded.

8 Conclusions

The CAN attack detection system described in this work employs LSTM networks to monitor automobile CANs, detect attacks and raise alerts in real time. LSTM networks are leveraged because they can learn patterns with long sequences. To address LSTM network bias, binning is employed as a novel concept in LSTM model development for automobile CANs. In this process, CAN message identifiers are separated into bins depending on their relative message frequencies. A separate LSTM model is trained to recognize the traffic patterns of the CAN message identifiers in each bin.

A repeatable design framework is presented for constructing and training multiple LSTM networks that learn normal CAN message timing patterns. The design framework lays out the computational resources as well as the data collection and preprocessing and LSTM model development and training steps. The framework enables new LSTM models to be trained and updated for automobiles of different makes, models and years.

Another key contribution is the implementation of real-time attack detection. The attack detection system leverages a server-client configuration on a monitored automobile CAN bus. The server is an inexpensive Raspberry Pi device connected directly to an automobile CAN bus that captures, logs and transmits CAN message traffic to a client via a Wi-Fi network. The client, a workstation located outside the automobile, provides the computational resources needed for real-time attack detection. Trained LSTM models executing on the client workstation process the transmitted CAN messages, identify attacks and send alerts via the Wi-Fi network.

The attack detection system was evaluated using a 2010 Toyota Prius testbed and a fully-operational 2014 Toyota Prius automobile. An attack device was employed to inject random CAN message identifiers at random times. For the 2010 Toyota Prius testbed, the attack detection precision ranged from 0.985 to 1.000, sensitivity, specificity and accuracy were perfect 1.000, and the F1 score ranged from 0.992 to 1.000. For the 2014 Toyota Prius automobile, the attack detection precision ranged from 0.142 to 1.000, sensitivity ranged from 0.864 to 1.000, specificity ranged from 0.772 to 1.000, accuracy ranged from 0.980 to 1.000, and the F1 score ranged from 0.248 to 1.000. The low precision and F1 scores were due to high false positive to true positive rates in just one of the six LSTM models in the attack detection system used in the 2014 Toyota Prius automobile experiments. Nevertheless, the attack detection results are very good – LSTM model sensitivity ranged from 0.864 to 1.000 and accuracy ranged from 0.980 to 1.000. Sensitivity and accuracy are the most important metrics because LSTM models must recognize normal traffic and detect as many attacks as possible with high accuracy.

Acknowledgment. This research was supported by the National Science Foundation under Grant no. DGE 1501177.

References

1. Aldhyani, T., Alkahtani, H.: Attacks on autonomous vehicles: a deep learning algorithm for cybersecurity. Sensors **22**(1), article no. 360 (2022)
2. Bosch, CAN Specification Version 2.0, Technical Specification, Stuttgart, Germany (1991)
3. Bosch, CAN with Flexible Data-Rate Version 1.0, Technical Specification, Gerlingen, Germany (2011)
4. Bozdal, M., Samie, M., Aslam, S., Jennions, I.: Evaluation of CAN bus security challenges. Sensors **20**(8), article no. 2364 (2020)
5. Brownlee, J.: Long Short-Term Memory Networks with Python. Machine Learning Mastery, San Juan, Puerto Rico (2020)
6. Checkoway, S., et al.: Comprehensive experimental analysis of automotive attack surfaces. In: Proceedings of the Twentieth USENIX Security Symposium (2011)
7. Geron, A.: Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems. O'Reilly Media, Sebastopol (2019)
8. Greenberg, A.: Hackers remotely kill a jeep on the highway – with me in it, Wired (2015)
9. Hanselmann, M., Strauss, T., Dormann, K., Ulmer, H.: CANet: an unsupervised intrusion detection system for high-dimensional CAN bus data. IEEE Access **8**, 58194–58205 (2020)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Hoppe, T., Dittman, J.: Sniffing/replay attacks on CAN buses: a simulated attack on the electric window lift classified using an adapted CERT taxonomy. In: Proceedings of the Second Workshop on Embedded Systems Security (2007)
12. Hoppe, T., Kiltz, S., Dittmann, J.: Security threats to automotive CAN networks - practical examples and selected short-term countermeasures. Reliab. Eng. Syst. Saf. **96**(1), 11–25 (2011)
13. Koscher, K., et al.: Experimental security analysis of a modern automobile. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 447–462 (2010)
14. Liu, J., Zhang, S., Sun, W., Shi, Y.: In-vehicle network attacks and countermeasures: challenges and future directions. IEEE Network **31**(5), 50–58 (2017)
15. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
16. Matthews, C.: Jeep hack: Fiat recalls 1.4 million vehicles for software fix, Fortune (2015)
17. McKinney, W.: Python for Data Analysis - Data Wrangling with Pandas, NumPy and IPython. O'Reilly Media, Sebastopol (2018)
18. Möller, D.P.F., Haas, R.E.: Automotive cybersecurity. In: Guide to Automotive Connectivity and Cybersecurity. CCN, pp. 265–377. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-73512-2_6
19. Ranjan, C.: Step-by-step understanding of LSTM autoencoder layers. Towards Data Science (2019). https://towardsdatascience.com/step-by-step-understanding-lstm-autoencoder-layers-ffab055b6352
20. Smith, C.: The Car Hacker's Handbook: A Guide for the Penetration Tester. No Starch Press, San Francisco (2016)
21. Sun, H., Chen, M., Weng, J., Liu, Z., Geng, G.: Anomaly detection in in-vehicle networks using CNN-LSTM with attention mechanism. IEEE Trans. Veh. Technol. **70**(10), 10880–10893 (2021)

22. Tariq, S., Lee, S., Woo, S.: CANtransfer: transfer-learning-based intrusion detection in a controller area network using a convolutional LSTM network. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Applied Computing, pp. 1048–1055 (2020)
23. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: Proceedings of the IEEE International Conference on Data Science and Advanced Analytics, pp. 130–139 (2016)
24. Valasek, C., Miller, C.: Adventures in Automotive Networks and Control Units. Technical White Paper, IOActive, Seattle, Washington (2014)
25. Valasek, C., Miller, C.: Remote Exploitation of an Unaltered Passenger Vehicle. Technical White Paper, IOActive, Seattle, Washington (2015)
26. Xiao, J., Wu, H., Li, X.: Robust and self-evolving IDS for in-vehicle networks by enabling spatiotemporal information. In: Proceedings of the Twenty-First IEEE International Conference on High Performance Computing and Communications, Seventeenth IEEE International Conference on Smart City and Fifth IEEE International Conference on Data Science and Systems, pp. 1390–1397 (2019)
27. Yang, Y., Duan, Z., Tehranipoor, M.: Identifying a spoofing attack on an in-vehicle CAN bus based on the deep features of an ECU fingerprint signal. Smart Cities **3**(1), 17–30 (2020)
28. Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile edge-assisted literal multi-dimensional anomaly detection in an in-vehicle network using LSTM. IEEE Trans. Veh. Technol. **68**(5), 4275–4284 (2019)

Author Index

Published by Springer Nature Switzerland AG 2024
J. Staggs and S. Shenoi (Eds.): ICCIP 2023, IFIP AICT 686, p. 253, 2024.
https://doi.org/10.1007/978-3-031-49585-4