

Habtamu Abie · Silvio Ranise ·
Luca Verderame · Enrico Cambiaso ·
Rita Ugarelli · Gabriele Giunta ·
Isabel Praça · Federica Battisti (Eds.)

LNCS 12618

Cyber-Physical Security for Critical Infrastructures Protection

First International Workshop, CPS4CIP 2020
Guildford, UK, September 18, 2020
Revised Selected Papers



Springer

Founding Editors

Gerhard Goos

Karlsruhe Institute of Technology, Karlsruhe, Germany

Juris Hartmanis

Cornell University, Ithaca, NY, USA

Editorial Board Members

Elisa Bertino

Purdue University, West Lafayette, IN, USA

Wen Gao

Peking University, Beijing, China

Bernhard Steffen 

TU Dortmund University, Dortmund, Germany

Gerhard Woeginger 

RWTH Aachen, Aachen, Germany

Moti Yung

Columbia University, New York, NY, USA

More information about this subseries at <http://www.springer.com/series/7410>


Habtamu Abie · Silvio Ranise ·
Luca Verderame · Enrico Cambiaso ·
Rita Ugarelli · Gabriele Giunta ·
Isabel Praça · Federica Battisti (Eds.)


Cyber-Physical Security for Critical Infrastructures Protection


First International Workshop, CPS4CIP 2020
Guildford, UK, September 18, 2020
Revised Selected Papers


Editors


Habtamu Abie 
Norwegian Computing Center
Oslo, Norway

Luca Verderame 
Università degli Studi di Genova
Genoa, Italy


Rita Ugarelli 
SINTEF A.S.
Oslo, Norway

Isabel Praça 
Instituto Superior de Engenharia do Porto
Porto, Portugal

Silvio Ranise 
University of Trento and Fondazione
Bruno Kessler
Trento, Italy

Enrico Cambiaso 
IEIIT Institute
Consiglio Nazionale delle Ricerche (CNR)
Genoa, Italy

Gabriele Giunta
Engineering Ingegneria Informatica S.p.A.
Rome, Italy

Federica Battisti 
University of Padua
Padua, Italy

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-69780-8 ISBN 978-3-030-69781-5 (eBook)
<https://doi.org/10.1007/978-3-030-69781-5>

LNCS Sublibrary: SL4 – Security and Cryptology

© Springer Nature Switzerland AG 2021

Chapters 5, 6, 10, 11, 13 and 14 are licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapters.

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Preface

CPS4CIP 2020 is a forum for researchers and practitioners working on cyber-physical security for critical infrastructures protection that supports finance, energy, health, air transport, communication, gas, and water. The secure operation of critical infrastructures is essential to the security of nations and, in an increasingly interconnected world, of unions of states sharing their infrastructures in order to develop their economies, and to public health and safety. Security incidents in critical infrastructures can directly lead to a violation of users' safety and privacy, physical damage, interference in the political and social life of citizens, significant economic impact on individuals and companies, and threats to human life while decreasing trust in institutions and questioning their social value. Because of the increasing interconnection between the digital and physical worlds, these infrastructures and services are more critical, sophisticated, and interdependent than ever before. The increased complexity makes each infrastructure increasingly vulnerable to attacks, as confirmed by the steady rise of cyber-security incidents, such as phishing or ransomware, and cyber-physical incidents, such as physical violation of devices or facilities in conjunction with malicious cyber activities. To make the situation even worse, interdependency may give rise to a domino effect with catastrophic consequences on multiple infrastructures.

To address these challenges, the workshop aimed to bring together security researchers and practitioners from the various verticals of critical infrastructures (such as the financial, energy, health, air transport, communication, gas, and water domains) and rethink cyber-physical security in the light of the latest technological developments (e.g., Cloud Computing, Blockchain, Big Data, AI, Internet-of-Things) by developing novel and effective approaches to increase the resilience of critical infrastructures and the related ecosystems of services.

The workshop attracted the attention of the critical infrastructures protection research communities and stimulated new insights and advances with particular attention to the integrated cyber and physical aspects of security in critical infrastructures. The first International Workshop on Cyber-Physical Security for Critical Infrastructures Protection (CPS4CIP 2020) was held online. The workshop was organized in conjunction with the 25th European Symposium on Research in Computer Security (ESORICS 2020), Guildford, the United Kingdom on 14–18 September 2020. The format of the workshop included two keynotes and technical presentations. The workshop was attended by around 28 people on average.

The workshop received 24 submissions, of which one was withdrawn and 23 were sent for reviews, from authors in 15 distinct countries. After a thorough peer-review process, 14 papers were selected for presentation at the workshop. The review process focused on the quality of the papers, their scientific novelty, and their applicability to the protection of critical financial infrastructure and services, and the acceptance rate was 58%. The accepted articles represent an interesting mix of techniques for security threat intelligence, data anomaly detection (predict and prevent), computer vision and

datasets for security, security management and governance, and impact propagation and power traffic analysis. The workshop was proactive with two important and stimulating keynotes in the areas of “Digital twins in industrial ecosystems: challenges, security issues, and countermeasures”, and “Cyber-physical security in automotive: the new challenge for smart cities”.

The workshop was supported by projects of the ECSCI (European Cluster for Securing Critical Infrastructures) cluster (<https://bit.ly/35YKnyE>), mainly FINSEC (www.finsec-project.eu), ANASTACIA (www.anastacia-h2020.eu/), DEFENDER (<https://defender-project.eu/>), InfraStress (www.infrastress.eu/), RESISTO (www.resistoproject.eu/), SAFECARE (www.safecare-project.eu/), SATIE (<http://satie-h2020.eu>), SecureGas (www.securegas-project.eu/), SPHINX (sphinx-project.eu/), and STOP-IT (stop-it-project.eu/). The organizers would like to thank these projects for supporting the CPS4CIP 2020 workshop.

Finally, the organizers of the CPS4CIP 2020 workshop would like to thank the CPS4CIP 2020 Program Committee, whose members made the workshop possible with their rigorous and timely review process. We would also like to thank the University of Surrey for hosting the workshop and the ESORICS 2020 workshop chair for valuable help and support.

December 2020

Habtam Abie
Silvio Ranise
Luca Verderame
Enrico Cambiaso
Rita Ugarelli
Gabriele Giunta
Isabel Praça
Federica Battisti

Organization

General Chairs

Habtamu Abie
Silvio Ranise

Norwegian Computing Center, Norway
Fondazione Bruno Kessler (FBK), Italy

Program Committee Chairs

Luca Verderame
Enrico Cambiaso
Rita Ugarelli
Gabriele Giunta
Isabel Praça
Federica Battisti

University of Genoa, Italy
National Research Council (CNR), Italy
SINTEF, Norway
Engineering Ingegneria Informatica S.p.A., Italy
GECAD/ISEP, Portugal
Università degli Studi Roma Tre, Italy

Program Committee

Dieter Gollmann
Sokratis Katsikas

Hamburg University of Technology, Germany
Norwegian University of Science and Technology,
Norway

Javier Lopez
Fabio Martinelli
Einar Snekkenes

University of Malaga, Spain
IIT-CNR, Italy
Norwegian University of Science and Technology,
Norway

Omri Soceanu
Stamatis Karnouskos
Reijo Savola
Alessandro Armando
Alessio Merlo
Cristina Alcaraz
Giovanni Livraga
Gustavo G. Granadillo
Stefan Poslad
Shouhuai Xu
Christos Xenakis
Mauro Conti
Denis Čaleta
Ali Dehghantanha
Dušan Gabrijelčič
Nikolaus Wirtz
Theodore Zahariadis

IBM Research, Israel
SAP Research, Germany
VTT Technical Research Centre of Finland, Finland
University of Genoa, Italy
University of Genoa, Italy
University of Malaga, Spain
University of Milan, Italy
Atos Spain, Spain
Queen Mary University of London, UK
University of Texas at San Antonio, USA
University of Piraeus, Greece
University of Padua, Italy
Institute for Corporate Security Studies, Slovenia
University of Guelph, Canada
Jožef Stefan Institute, Slovenia
RWTH Aachen University, Germany
National and Kapodistrian University of Athens,
Greece

Adrien Bécue	AIRBUS CyberSecurity, France
Lorenzo Sutton	Engineering Ingegneria Informatica S.p.A., Italy
Harsha Ratnaweera	Norwegian University of Life Sciences, Norway
Volodymyr Tarabara	Michigan State University, USA
Christos Makropoulos	National Technical University of Athens, Greece
Alessandro Neri	Università degli Studi Roma Tre, Italy
Stefano Panzieri	Università degli Studi Roma Tre, Italy
David Tipping	DeftEdge, USA
Dionysis Nikolopoulos	National Technical University of Athens, Greece
Véronique Legrand	Cnam, France
Ioan Constantin	Orange Romania, Romania
Tim Stelkens-Kobsch	German Aerospace Center (DLR), Germany
Matteo Mangini	Network Integration and Solutions S.r.l., Italy
Mirjam Fehling-Kaschek	Fraunhofer Inst. for High-Speed Dynamics, Germany
Vasileios Kazoukas	Center for Security Studies (KEMEA), Greece

Contents

Security Threat Intelligence

Privacy-Preserving CCTV Analytics for Cyber-Physical Threat Intelligence	3
<i>Jürgen Neises, Adrien Besse, and Jean-Baptiste Rouquier</i>	
TLSAssistant Goes FINSEC A Security Platform Integration Extending Threat Intelligence Language	16
<i>Salvatore Manfredi, Silvio Ranise, Giada Sciarretta, and Alessandro Tomasi</i>	
Cyber Threat Monitoring Systems - Comparing Attack Detection Performance of Ensemble Algorithms	31
<i>Eva Maia, Bruno Reis, Isabel Praça, Adrien Becue, David Lancelin, Samantha Dauguet Demailly, and Orlando Sousa</i>	
FINSTIX: A Cyber-Physical Data Model for Financial Critical Infrastructures	48
<i>Giorgia Gazzarata, Ernesto Troiano, Luca Verderame, Maurizio Aiello, Ivan Vaccari, Enrico Cambiaso, and Alessio Merlo</i>	

Data Anomaly Detection: Predict and Prevent

Inferring Anomaly Situation from Multiple Data Sources in Cyber Physical Systems.	67
<i>Sara Baldoni, Giuseppe Celozzi, Alessandro Neri, Marco Carli, and Federica Battisti</i>	
Fusing RGB and Thermal Imagery with Channel State Information for Abnormal Activity Detection Using Multimodal Bidirectional LSTM.	77
<i>Nikolaos Bakalos, Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Kassiani Papatotiriou, and Matthaïos Bimpas</i>	
A Cloud-Based Anomaly Detection for IoT Big Data	87
<i>Omri Soceanu, Allon Adir, Ehud Aharoni, Lev Greenberg, and Habtamu Abie</i>	

Computer Vision and Dataset for Security

An Advanced Framework for Critical Infrastructure Protection Using Computer Vision Technologies	107
<i>Krishna Chandramouli and Ebroul Izquierdo</i>	

A Comprehensive Dataset from a Smart Grid Testbed for Machine Learning Based CPS Security Research 123
Chuadhry Mujeeb Ahmed and Nandha Kumar Kandasamy

Security Management and Governance

Cross-Domain Security Asset Management for Healthcare 139
Federico Stirano, Francesco Lubrano, Giacomo Vitali, Fabrizio Bertone, Giuseppe Varavallo, and Paolo Petrucci

Towards a Global CIs’ Cyber-Physical Security Management and Joint Coordination Approach 155
Vasiliki Mantzana, Eftichia Georgiou, Anna Gazi, Ilias Gkotsis, Ioannis Chasiotis, and Georgios Eftychidis

Toward a Context-Aware Methodology for Information Security Governance Assessment Validation 171
Marco Angelini, Silvia Bonomi, Claudio Ciccotelli, and Alessandro Palma

Impact Propagation and Power Traffic Analysis

Impact Propagation in Airport Systems 191
Corinna Köpke, Kushal Srivastava, Louis König, Natalie Miller, Mirjam Fehling-Kaschek, Kelly Burke, Matteo Mangini, Isabel Praça, Alda Canito, Olga Carvalho, Filipe Apolinário, Nelson Escravana, Nils Carstengerdes, and Tim Stelkens-Kobsch

A Comparative Analysis of Emulated and Real IEC-104 Spontaneous Traffic in Power System Networks 207
C.-Y. Lin and Simin Nadjm-Tehrani

Author Index 225

Security Threat Intelligence



Privacy-Preserving CCTV Analytics for Cyber-Physical Threat Intelligence

Jürgen Neises¹(✉), Adrien Besse², and Jean-Baptiste Rouquier²

¹ Fujitsu Technology Solutions GmbH, Düsseldorf, Germany
juergen.neises@fujitsu.com

² Fujitsu Technology Solutions SAS, Paris, France
{adrien.besse, jeanbaptise.rouquier}@fujitsu.com

Abstract. This paper describes the FUJITSU CCTV Analytics service (FCAS), a privacy-preserving CCTV surveillance module using AI based video analytics for cyber-physical threat intelligence. The design and architecture of a privacy preserving and thus GDPR friendly way to improve security by automatically analysing video feeds and sending events that can be interpreted as a threat to further analytics is illustrated. The system has been applied to several scenarios like data centre and at ATMs. The developments can also be applied to general public safety requests and may be utilized coping with the COVID-19 impacts. Finally, the solution shall be adapted to edge computing. First steps illustrate the capabilities of small form factor systems.

Keywords: CCTV · Cyber-physical-threat-intelligence · Privacy · Human pose detection

1 Introduction

This document is about physical threat intelligence provided by the FUJITSU CCTV Analytics service (FCAS). Video surveillance is by essence dedicated to monitor assets and interactions with them within a camera scene. Today most security systems are not designed for proactive use but for forensics when a breach has already happened. In our case, the goal is a privacy preserving and thus GDPR friendly way to improve security by interpreting video feeds and sending events that can be interpreted as a threat to further analytics. FCAS will not store any biometric data. Moreover, FCAS is by design agnostic of the security or business use cases it is supposed to support pushing events. The goal is to be able to detect bodies and body parts (e.g. heads, hands, etc.) and capture their interactions with each other and with physical objects (e.g. ATMs or racks). The observations generate events that shall be pushed to data collection for further analytics and correlation.

FCAS provides a set of AI models, able to detect objects out of the box, by using human pose estimation. They are able to detect series of human actions, e.g., opening a door or interacting with an ATM. These interactions result in events indicating potential physical threats. These events can be correlated by AI and analytics identifying physical

threats and indicate to the human operator where to focus his attention or automatically initiate mitigation actions. This way, CCTV systems can be leveraged in order to improve security.

The FCAS CCTV probe relies on one or more CCTV cameras watching a scene, i.e. the complete camera picture. Moreover, relevant static areas – so-called zones – can be easily defined by drawing a polygon around them, thus defining a separated detection area. Then the FCAS can track interactions of dynamic objects within those zones.

FCAS combines the detection of objects, e.g. body parts, and the detection of body poses for analysing the CCTV images. Based on the results FCAS generates action related events. Some models are already available for those tasks, but they often need to be retrained for the specific purpose. Such existing technologies for object detection and pose analysis are combined and retrained to process the video feed in real-time using CPUs or preferably GPUs as in FCAS and to generate events for further security analysis.

1.1 Object Detection

Object detection aims to detect all instances of objects from a known class, such as people, or body parts as heads in an image. Typically, only a small number of instances of the object are present in the image, but they can occur at a large number of possible locations and scales that need to somehow be explored.

Each detection is reported with some form of pose information. This could be as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. Providing even more information, a face detector might compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face. The pose could also be defined by a three-dimensional transformation specifying the location of the object relative to the camera.

Object detection systems construct a model for an object class, e.g. heads, from a set of training examples. In the case of a fixed rigid object, only one example may be needed, but more generally, multiple training examples are necessary to capture certain aspects of class variability.

We therefore focus on typical security elements that appear in common video surveillance:

- **Counting People:** The probe sends timestamped events when someone enters or exits the area, allowing getting the number of people in each area.
- **Discriminating Adults from Children:** This element is not implemented in the probe yet but could be added in later versions of FCAS. Most predictions on age are based on face image analysis and require a sufficient level of details for the model to give a reliable age prediction.
- **Face Detection:** The FCAS uses a pose estimation model to detect and track different face attributes such as the nose, eyes and ears of people (cf. Fig. 1). One of the advantages of using pose estimation models instead of pure face detection model is that it can detect people even when the faces are hidden (somebody wearing a balaclava or somebody facing in the opposite direction) from the presence of other body parts.

- Weapon Detection:** The current version of FCAS focuses on human body parts positions in the scene. FCAS models detecting particular weapons in the hands of somebody are not assessed yet. Major challenges of this approach are creating the training datasets gathering a suitable amount of weapon annotated images extracted from similar security CCTV camera visual context (images from movies contain plenty of weapons but are quite different from CCTV images).

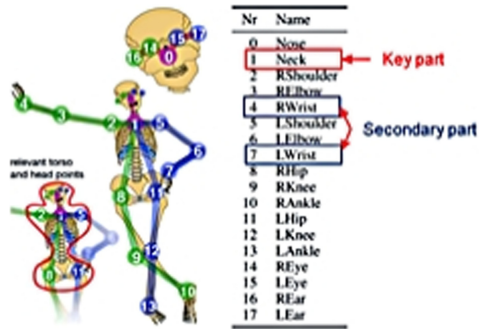


Fig. 1. Body parts definition used by the pose estimation model. Image source Platte et al. [1]

1.2 Human Pose Detection

Human Pose Estimation attempts to find the orientation and configuration of human body parts. 2D Human Pose Estimation, or Keypoint Detection, generally refers to localize body parts of humans, e.g. finding the 2D location of the knees, eyes, feet, etc. Body parts, which are typically used are depicted in Fig. 1.

Pose estimation is a difficult problem and an active research subject partly because the human body has 244 degrees of freedom with 230 joints. Although not all movements between joints are needed, a common approach is to model the human body by 10 large parts with 20 degrees of freedom [1]. Algorithms must account for large variability introduced by differences in appearance due to clothing, body shape, size, and hairstyles. Additionally, the results may be ambiguous due to partial occlusions from self-articulation, such as a person's hand covering their face, or occlusions from external objects. Other issues include varying lighting and camera configurations. In the FINSEC case, we estimate pose from monocular (two-dimensional) images, taken from a normal camera.

Human pose estimation is crucial for video surveillance [2]. Recently, significant progress has been made in solving the human pose estimation problem in unconstrained single images. However, human pose estimation in videos is a relatively new and promising problem and needs much further improvement.

Obviously, single image-based pose estimation methods can be applied to each video frame to get an initial pose estimation as illustrated in Fig. 2. A further refinement through frames can be applied to make the pose estimation consistent and more accurate and reliable.

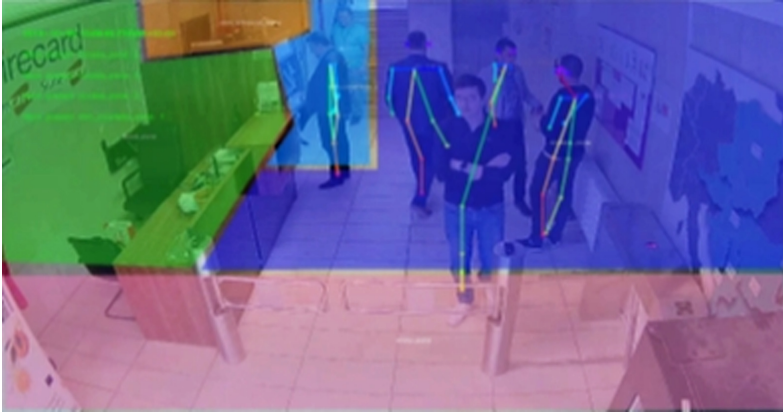


Fig. 2. Human Pose detection.

2 CCTV Analytics Technology Description

Within the global architecture in the FINSEC project, FCAS is a probe working in the edge: the FCAS probe uses local CCTV surveillance for analytics and generating events. Converting videos to events is done locally. The events occurring in the local scene are generated in the edge by the FCAS analytics and pushed to a data layer. Anomaly detection and threat prediction utilize the events stored in the data layer for security analysis.

The CCTV FCAS probe architecture is composed of modules that hierarchically process the information contained in the video stream. Figure 3 illustrates the processing flow by the probe from the camera video stream to the events sent to the data layer of the FINSEC platform.

The main modules:

- The **video reader** receives the video frames from the camera
- The **detection** module detects defined body parts and infers their coordinates in those video frames.
- The **tracking** module matches new detections in the last frame to detections in the previous ones, for each body part, thus keeping track of the body part positions. The tracking algorithm uses distances between the current detection and the predicted position (derived from a Kalman filter). (Currently the configuration takes into account frames from the last 12 s). The position of a human body is taken as the position of one key body part (cf. Fig. 4) which in our case is the ‘Neck’ for the pose estimation model or the head for the head detection model.
- The **event generation** module looks for tracking results and generates events about body parts entering or exiting configured areas in an open STIX related format to a security analytics platform.

The body parts illustrated in Fig. 4 can be detected and tracked by the pose estimation model and the human tracker.

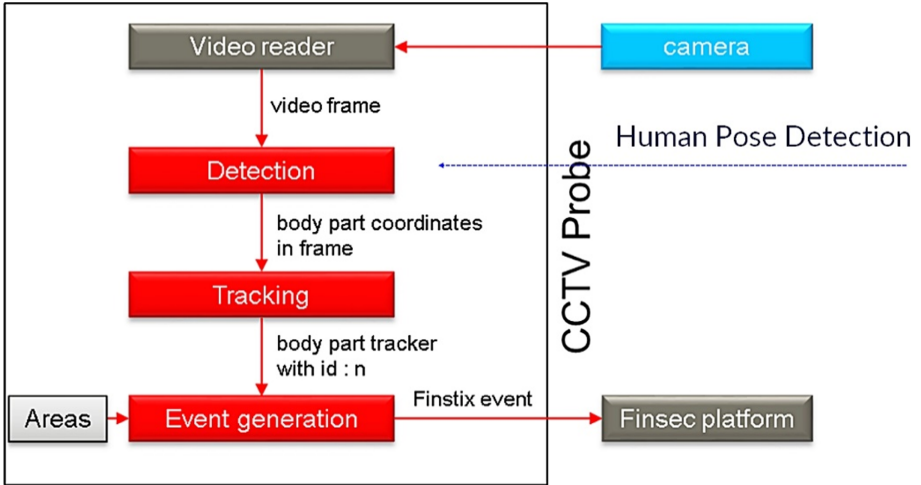


Fig. 3. High-level architecture of the probe.

To control the number of emitted events, the body parts effectively taken into account for event generation can be set through the FCAS configuration API. By this configuration, the lists of body parts taken into account at event level can be defined to be used for detection of a suspicious event.

In the current version of the FCAS probe with pose estimation, the ‘Neck’ position is selected as the key body part, representative of the body position. Secondary parts are other tracked body parts, e.g. right and left wrists.



Fig. 4. Body parts for pose estimation

2.1 Detection

The detection module uses a pose estimation model or a head detection model to infer the coordinates of body parts in video frames received from the camera. The detection module is responsible for processing a frame and outputs the positions of body parts. The module detects the position of heads, neck or right and left wrists depending on a parameter indicating which body part to track.

The models to detect human bodies are obtained from a pose estimation library, in this case the NVIDIA library “Real-time pose estimation accelerated with NVIDIA TensorRT” [3]. The library authors follow the approach in references [4] and [5], based on part affinity fields and key point detections, for pose estimation. That library provides:

- a pre-trained model zoo for human pose estimation, resulting from several experiments on the MSCOCO Keypoints Challenge 2017 [6]. The experiments are based on several backbone network architectures such as densenet121, densenet169 [7], resnet18 and resnet50 [8].
- a training script allowing to fine-tune the pre-trained model on our specific datasets
- a module to compile model, e.g. using the NVIDIA TensorRT library as described above, allowing to optimize model inference on NVIDIA Hardware such as the Jetson platforms. The library authors report throughputs on Jetson Nano and Jetson AGX Xavier respectively of 22 FPS and 251 FPS with a resnet18 network architecture [3]. The range of available backbone architectures in the library zoo allows optimizing performances versus throughput and latency of the finally deployed model.

The model for head detection uses a MobileNet [9] deep neural network architecture trained on two datasets, namely the Open images dataset [9–11] and the Brainwash dataset [12].

2.2 Tracking

The tracking module is responsible for initializing and updating the following information:

- The object id (here an object is a human body)
- The positions (x, y) in pixels of its body parts
- The area names in which the body parts are present
- The timestamp of entry in the area
- The current timestamp

The tracking algorithm applied in FCAS is based on the SORT algorithm [13]. This algorithm is effective and uses position extrapolation. The first approach of this algorithm is that objects do not move a lot between frames: if the FCAS system detects an object in one frame, 100 ms seconds later it cannot be far from where it was before. In addition, it computes a direction and predicts the new location in the current frame. The FCAS probe searches for the object next to where it is predicted to be. Predicting the path of the object allows following the object even if someone else temporarily occludes it, which means that only the occluding person is detected. Note that this algorithm preserves privacy as the FCAS tracks objects by predicting their future position and not by recognizing them.

In order to efficiently address the tracking of human bodies through video, we have developed a human model using individual trackers for each body parts and combining the information of all the body part trackers to keep track of the human body. The building blocks are trackers based on Kalman filters [14] for each body part position

prediction. The matching between tracker predictions and detection is performed following the Hungarian algorithm [15]. Open source implementation of those are Python packages `filterpy.kalman` and `scipy.optimize.linear_sum_assignment`. From the building block consisting in one tracker, we have developed a global human tracker combining a set of body part trackers. Each human model tracker has a unique ID. This way pose estimation outputs based on various body parts make human model tracking more robust than a tracker based on a single human detection, as relies on more information than just on a unique bounding box detection.

The Human model tracker has internal states such as centre positions and velocities for each part of the body. The event generator to monitor human bodies in the video can access the human model internal state. In particular, left and right ankle positions can be used to map the human positions on a floor map of the scene.

2.3 Event Generation

The FCAS collects and processes the video stream of a camera and stops if the video stream ends. The event generator module is responsible for sending events generated from the tracker inputs in a STIX like format. We outline in this section the various events that can be generated by the FCAS.

New Agent Events

When a new body is detected in the camera field or scene for more than a defined period (e.g. 2 s), the FCAS creates a new agent instance with new id and sends an ENTER event.

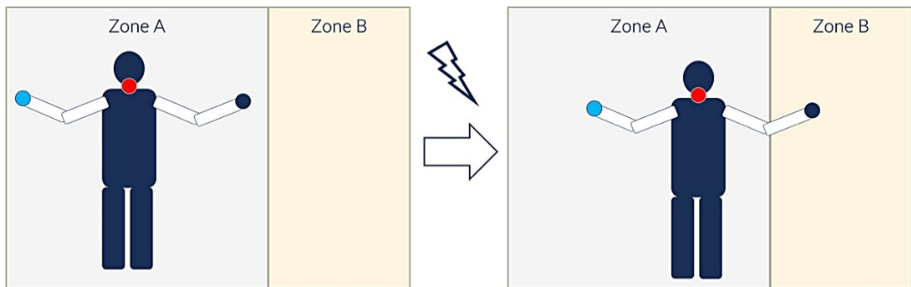


Fig. 5. Enter/Exit Event

When a body disappears from the camera field of view, the FCAS generates and sends an EXIT event for this from the last area it appeared in, and cleans the information of this body.

Enter/Exit Events

When a body part enters a scene area, e.g. Zone A for the time, FCAS sends an “entering area ‘Zone A’” event. Later, when a body part (e.g. LWrist as illustrated in Fig. 5),

exits from the ‘Zone A’ area and enters into the ‘Zone B’ an area, FCAS sends the corresponding events “ENTER (Zone B)” and “EXIT (Zone A)”.

Velocity Related Events

These events are related to the velocity of bodies moving through the scene. The neck is the default reference point defining the body and thus its velocity. The events are illustrated in Fig. 6.

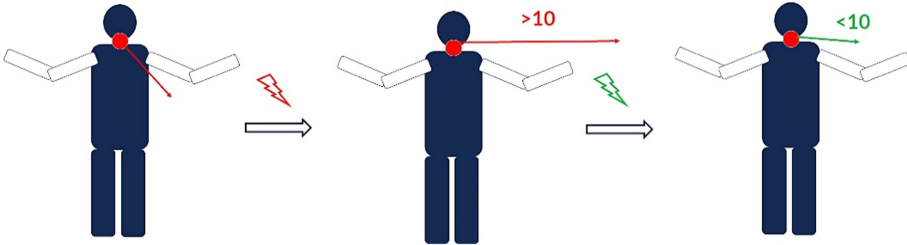


Fig. 6. High Velocity and Slow Down

The configurable value `body_speed_threshold` defines the threshold above which to trigger **high velocity** events.

If the velocity of a reference point, e.g. 10 cm per second is shrinking an event indicating **slow down** of the agent is issued.

Proximity Related Events

These events consider the proximity of two bodies. The value `parts_body_proximity` specifies a list of body parts used to determine if two human bodies are closer than a given threshold (cf. Fig. 7). The values defined in `body_radius_proximity` set the corresponding thresholds defining body proximity.

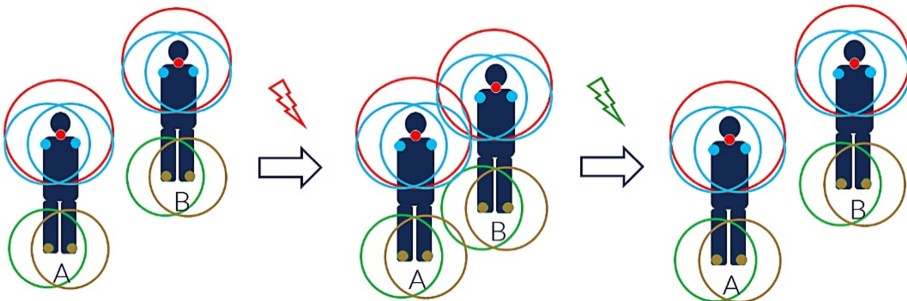


Fig. 7. Approaching and Distancing

Two events are related to body proximity:

- APPROACHING when the distance is below the set threshold and shrinking
- DISTANCING when the distance is above the defined threshold and increasing

Trajectory Related Events

Loitering is a major security event and is worth monitoring. Thus, the body trajectories are tracked on a floor map within the scene. Additionally, their length is compared with `length_floormap_trajectory` and `length_trajectory_threshold`. When the length of the trajectory in the floor map exceeds the defined threshold (cf. Fig. 8), the TRAJECTORY event is triggered. In combination with the time an agent is in the scene this event indicates loitering,

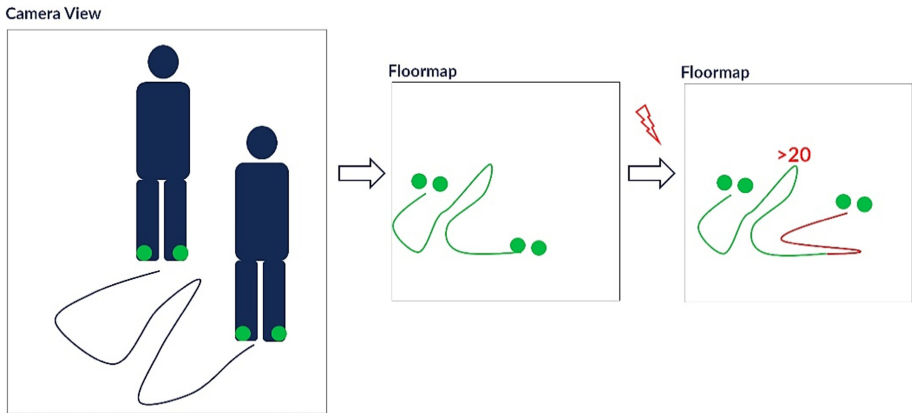


Fig. 8. Trajectory Event

3 Application to Common Physical Security Scenarios

In this section, the impact of the FCAS are outlined related to two common physical security scenarios. These comprise physical security in the data centre and at ATMs.

3.1 Scenario Data Centre

A well known security scenario in the data centre is the protection of access toracks, containing secured servers. By a common security policy, a secure rack has to be opened

by two people, each one using an authorized key. All the other configurations (one person using two keys, two people coming sequentially, etc.) are unauthorized in this case.

A camera on top of the rack may survey the scene. This way it can be detected, when someone enters the area in front of a rack and if the person uses the keyhole with one of their wrists. The FCAS probe allows tracking interactions of humans with key holes, counting the number of people, and tracking their positions in different areas. Events indicating the activities and potential protocol violations are submitted to a security platform and may be used by anomaly detection to trigger security actions at the data centre scale. In case, an anomaly triggers alarms and pushes them to the data centre operations.

3.2 Scenario ATMs

ATMs continue to be a profitable target for criminals. While some rely on physically destructive methods with metal cutting tools or explosives (including injecting gas), others choose malware infections. Criminals use different tools and techniques (physical, logical or combined) to access ATMs as a “black box” and bypass all security controls forcing the cash-out of all its money, cut communication with the ATM host in order to avoid any interference from operational personnel monitoring the smooth operation of the ATMs.

In this scenario, two camera feeds shall be utilized in the FCAS solution. Typically, a dome camera is monitoring the scene at an ATM. Additionally, a portrait camera on top of the ATM can be used to ensure detection of a person in the zones close to the ATM using head detection (cf. Fig. 9). This way, the precise position of the person related to the ATM left, centre, right can be detected.



Fig. 9. CCTV Analytics in the ATM Pilot

In addition, FCAS can monitor the number of people, the time spent in front of the ATM on in the monitored area and if they arrived in the same time in front of the ATM.

Among others, there are five well-known use cases in this scenario and several CCTV related events can be used for combined cyber-physical threat intelligence:

- **Attack to a person at the ATM.** To indicate such an attack, two or more persons need to be in the scene and they need to get close to each other.

- **Attack to an ATM.** To attack an ATM, one or more people need to be in the ATM zone.
- **Loitering.** A person staying in the scene for longer than a specific time indicates this. In addition, a trajectory longer than a defined length may add confidence to this.
- **Introduction of malware to the ATM.** From CCTV point of view, this is analogous to Attack to an ATM since the ATM needs to be manipulated
- **Jackpotting.** This also relates to the second use case “Attack to an ATM” considering CCTV events Since the ATM needs to be manipulated.

Moreover, events of cyber and physical probes can be correlated by an anomaly detection and when attacks are detected, alarms can be issued or other mitigating actions can be initiated.

3.3 Application to Other Segments

The FCAS development is based on an early very general Proof-of-Concept in the retail segment. Beyond the described scenarios the ongoing FCAS development has been considered for retail based use cases.. Among those, use cases counting people entering shops, monitoring how long people stay in front of items, and interactions with sellers.

With the spring 2020 COVID-19 crisis, the requests for counting people inside an area and monitoring their distances generated further application scenarios for this technology. Especially limited access to shops can be controlled counting people. Moreover, utilizing floor maps of critical areas, e.g. at the cashiers desk, distances between people can be monitored. The latter might be applied also to critical public spaces.

4 Deployment – from Server to Edge Computing

During the FCAS development, a trend towards application of AI in the edge has been observed. Therefore, a deployment in the edge has been explored.

The development starting point was applying existing technology based on a system with a powerful GPU, e.g. an X64 system and at least a GPU as powerful as an NVIDIA 1080GTX. This limits the utilization of the FCAS due to the system requirements and related cost. Therefore, the system was not easily deployed in the edge.

This is the reason to evaluate small form factor systems, in this case the NVIDIA Jetson platform, namely the Jetson AGX Xavier, which is expected to provide the required performance at lower cost than a typical server-based GPU and at a small form factor, as a first step towards edge computing or even embedded application, e.g. inside an ATM.

With this process applying a further specialized computer architecture, i.e. ARM in addition to $\times 64$, we faced common issues in the dynamic development of new hardware and open source software as depicted in Fig. 10.

Even between servers with the same operating system (Ubuntu 18.04) the libraries and versions of the underlying software change. This implies problems in finding the right Python packages version in repositories for the specific ARM architecture. Moreover, conflicts occurred as some parts required different versions of a given Python library.

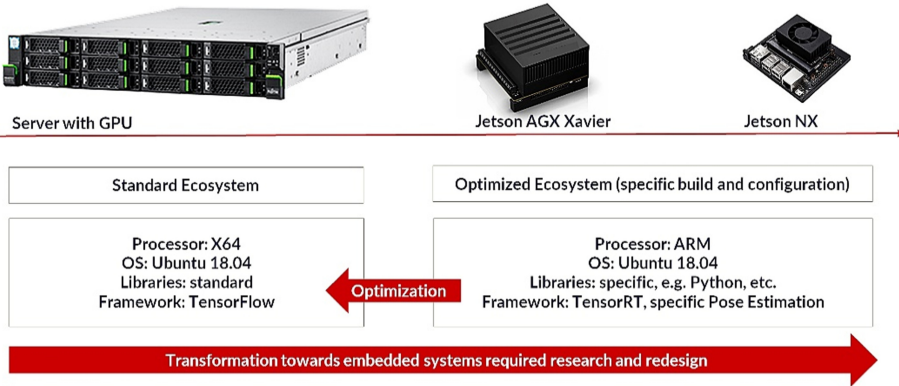


Fig. 10. From Server to Edge Computing

The precise list of packages and versions would be fastidious to list. The main libraries that faced such issues are scikit-learn and scipy (both needed for the tracking algorithm).

To speed up deep learning inference as compared to TensorFlow, we moved to the TensorRT framework published by NVIDIA. It provides optimizations for GPU-based platforms. This requires usage of a distinct, dedicated open source code repository for pose estimation instead of the original one (https://github.com/NVIDIA-AI-IOT/trt_pose). The latter is written in pytorch and needs to be compiled to be optimized on each hardware (Jetson GPU or common GPU), while the former one used an open source TensorFlow implementation of pose estimation. However, this replacement may be advantageous for both hardware environments (edge and server) as it the optimisation through TensorRT may be used for underlying hardware compatible with the library. This way an optimization can be achieved for both computing environments by this pose estimation implementation.

5 Conclusion

This paper shows a privacy-preserving CCTV surveillance module using AI based video analytics for cyber-physical threat intelligence. It is based on common technologies for detection of body parts and human pose detection. There is an ongoing development to move the solution towards and edge based intelligent sensor of potential physical threats.

Acknowledgement. A substantial part of this work has been carried out in the scope of the HORIZON 2020 FINSEC project, which is funded by the European Commission in the scope of its HORIZON 2020 programme (contract number 786727). The authors gratefully acknowledge the contributions of the funding agency and of all the project partners.

Besides the authors, Sebastien Iooss, Moïse Valvassori, Ivaylo Petkanchin, Thierry Lefort, Arthur Cahu and Thomas Walloschke were involved in the work of Fujitsu side.

References

1. Platte, B., et al.: Person tracking and statistical representation of person movements in surveillance areas. *Int. J. Des. Anal. Tools Integrated Circ. Syst.* 7(1), 10 (2018)
2. Zhang, D., Shah, M.: Human pose estimation in videos. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 2012–2020 (2015). <https://doi.org/10.1109/iccv.2015.233>
3. NVIDIA: “Real-time pose estimation accelerated with NVIDIA TensorRT”. https://github.com/NVIDIA-AI-IOT/trt_pose
4. Zhe, C.G.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, arxiv, 1812.08008 (2018)
5. Xiao, B., Haiping, W., Yichen, W.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
6. MSCOCO Keypoints Challenge (2017). <https://cocodataset.org/#keypoints-2017>
7. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261–2269 (2017). <https://doi.org/10.1109/cvpr.2017.243>
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770–778 (2016). <https://doi.org/10.1109/cvpr.2016.90>
9. Howard, A.G., et. al.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv e-prints, [arXiv:1704.04861](https://arxiv.org/abs/1704.04861). April 2017
10. Krasin, I., et. al.: OpenImages: A public dataset for large-scale multi-label and multi-class image classification (2017)
11. Kuznetsova, A.: The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale, arXiv, [arXiv:1811.00982](https://arxiv.org/abs/1811.00982) (2018)
12. Brainwash data set. <https://megapixels.cc/brainwash/>
13. SORT: SIMPLE ONLINE AND REALTIME TRACKING. <https://arxiv.org/pdf/1602.00763.pdf>
14. Kalman, R.: A new approach to linear filtering and prediction problems, *J. Basic Eng.* **82**, 35–45 (1960)
15. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**, 83–97 (1955)



TLSAssistant Goes FINSEC A Security Platform Integration Extending Threat Intelligence Language

Salvatore Manfredi^{1,2}(✉) , Silvio Ranise^{1,3} , Giada Sciarretta¹ ,
and Alessandro Tomasi¹ 

¹ Security and Trust, FBK, Trento, Italy
{`smanfredi, ranise, sciarretta, altomasi`}@fbk.eu

² DIBRIS, University of Genoa, Genoa, Italy

³ Department of Mathematics, University of Trento, Trento, Italy

Abstract. We present the integration of TLSAssistant, a tool for TLS vulnerability scanning and mitigation, with an online platform of services for cybersecurity in critical infrastructure. We highlight the added value of intelligence sharing and synergies with other services on the platform, as well as the non-trivial challenges encountered in the process.

Keywords: Integrated cybersecurity · Threat intelligence exchange · STIX · TLS

1 Introduction

Sharing threat intelligence, evaluating risk exposure and responding to threats in a timely manner are an open challenge - particularly in complex systems and critical infrastructures. Security platforms are emerging as innovative answers. There is no wide consensus on a common definition of the term security platform. Three crucial capabilities that seem to be recurring in several descriptions [12] are the following: *(i)* run a number of different security tools simultaneously across as many environments as possible; *(ii)* integrate the various tools in a uniform solution so to share threat intelligence information, contribute to the timely evaluation of the risk exposure, and participate to respond to threats; *(iii)* automate as much as possible security operations so to keep up with the sheer number and speed of transactions and attacks in complex systems and critical infrastructures.

While there exist many tools able to provide security-related functionalities, such as penetration testing and intrusion detection, being able to integrate them in a security platform and guarantee the features described above is a non-trivial task. This is so because individual tools are designed in isolation with little or no attention to information sharing in both input and output. This is not surprising as the tools are often developed by different vendors to maximize their effectiveness when used stand-alone.

In summary, the main challenges to re-using the wealth of security techniques available in stand-alone tools and effectively integrate them in a security platform are the following:

1. merging heterogeneous sources of security intelligence in a coherent way makes integration a complex task;
2. the results returned by the various tools must be integrated in a coherent and meaningful way to be able to calculate a risk exposure of the system under analysis.

With the arrival of the revised *Payment Services Directive* (PSD2) [3] and its standardization effort by the Berlin Group [1], the focus on securing TLS has become more and more important and relevant in financial contexts. In this paper, we present the integration of TLSAssistant [25], an open source tool that combines state-of-the-art TLS analyzers with a report system that suggests appropriate mitigations, in the FINSEC platform [5], which is an integrated framework for predictive and collaborative security of financial infrastructures.

The goals and added value in transitioning TLSAssistant from a command-line tool to a cloud service integrated in the FINSEC platform were *(i)* sharing the long-term mitigation intelligence provided by TLSAssistant reports; *(ii)* sharing the immediate per-scan intelligence of vulnerabilities sighted at specific URLs; and *(iii)* potentially enhancing the functionality of other services, based on individual integration. Achieving these goals let us face the two aforementioned challenges. In particular, the first challenge allows us to introduce in TLSAssistant the use of STIX, a language for exchanging cyber threat intelligence to simplify the integration; while the second challenge allows us to introduce the importance of defining appropriate risk metrics and associate them to the vulnerabilities detected by the tools so that they can be integrated and used by the Risk Assessment Engine available in the platform.

Plan of the Paper. Section 2 provides the necessary background notions on TLS, its vulnerabilities and the impact of its use when securing Financial APIs. The section also introduces TLSAssistant, a tool developed to assist system administrator to secure their TLS deployments. Section 3 gives an overview of the FINSEC platform by describing its structure, how it works and the parties involved. Section 4 describes how we addressed the challenges arising from the integration of TLSAssistant in FINSEC, and Sect. 5 further details the integration with the Risk Assessment Engine as deployed in FINSEC. Section 6 concludes the paper and highlights future work.

2 Assisted Hardening of TLS Configuration

Transport Layer Security (TLS) was first introduced in 1999 [15] to provide both confidentiality and integrity between communicating entities. Unfortunately, the secure configuration of TLS instances is far from trivial and requires time and security insight. In this section, we first give a concise description of two known

vulnerabilities (Sect. 2.1); then we present `TLSAssistant`, a tool proposed to assist system administrators and app developers to deploy resilient instances of the TLS protocol (Sect. 2.2); finally we describe the role of TLS for financial services (Sect. 2.3).

2.1 Vulnerabilities on TLS

TLS has been updated several times [9, 16, 17] to improve its security. The latest version, TLS 1.3, removes several now-deprecated cipher suites that made the protocol prone to a large number of vulnerabilities. Nevertheless, the most widely supported version is currently TLS 1.2. On this protocol version or prior, TLS suffers from a wide set of vulnerabilities [26]. In this paper, we focus on two representative examples:

Bar Mitzvah. [10] By exploiting the invariance weakness of the RC4 stream cipher, an attacker is able to retrieve the session cookie by guessing the least significant bits of the keystream. RC4 is a very widely used cipher suite. It has been known since 2013 that some RC4 weaknesses also affect SSL/TLS; nevertheless, almost 10% of the most popular sites in the world (based on Alexa’s list) is still supporting it [22].

ROBOT. [2] (*Return Of Bleichenbacher’s Oracle Threat*): due to the availability of the PKCS#1v1.5 padding algorithm in RSA, an attacker is able to extract the private key of the session and breaking the message confidentiality. By using an adaptive chosen-ciphertext attack, based on Daniel Bleichenbacher’s chosen-ciphertext attack, the victim is forced to leak information that help the attacker to guess the key. The key can then be used to decrypt HTTPS traffic sent between the TLS server and the user’s browser. Even though the most popular sites in the world are not vulnerable to ROBOT [22], researchers are still discovering different ways to exploit the same vulnerability with small variations to the Bleichenbacher attack, some of them affecting even TLS 1.3 [23].

2.2 TLSAssistant

`TLSAssistant` is an open-source tool [13] proposed to assist system administrators and app developers to deploy resilient instances of the TLS protocol. By bringing together different powerful analyzers, `TLSAssistant` is able to cover a full-range of analysis on all the parties involved in a secure communication and to provide a set of mitigation measures that aim to mitigate the impact of the identified vulnerabilities.

Architecture. `TLSAssistant` is written in Bash and can thus be invoked via command-line. The tool takes as input the target to be evaluated (e.g., the IP address of a server) and outputs a report file. The content of the report depends on the detected weaknesses and on the level of verbosity the user chose.

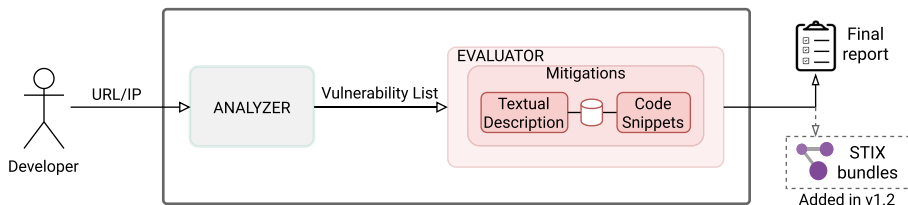


Fig. 1. TLSAssistant architecture.

Figure 1 shows the architecture with its two main components: ANALYZER and EVALUATOR.

ANALYZER takes as input the address of the target webserver. TLSAssistant v1.2, the version we integrated in the FINSEC platform, included the following command-line tools: testssl.sh [29], MalloDroid [4] and tlsfuzzer [11], HTTP/HSTS checker. Once loaded, the ANALYZER module will run each of the tools related to the required scan, collect their reports and forward them to the EVALUATOR.

EVALUATOR is responsible for the enumeration of the detected vulnerabilities and the generation of the report that will guide the system administrator to apply the appropriate mitigations. It is currently able to generate reports that contain the vulnerabilities detected by the ANALYZER and the related mitigations described at various levels of abstraction:

Textual Description: natural language description of the TLS vulnerability, identification of the respective CVE ID, CVSS score and related mitigations (brief explanation of the actions to perform).

Code Snippet: a fragment of code that can be copy-pasted into the webserver’s configuration to seamlessly fix the weakness. TLSAssistant can detect any webserver but is currently only able to provide snippets for Apache and nginx HTTP server. Together with the snippet, the report will provide a set of steps on how to find the correct file/line to edit.

For example, in the case of an Apache server vulnerable to Bar Mitzvah, the report will contain the information shown in Fig. 2.

As described in Sect. 4.1, from version v1.2, TLSAssistant is able to export the analysis result in STIX (Structured Threat Information eXpression) language, which supports cyber threat intelligence activities and facilitates integration within security platforms (e.g., FINSEC).

2.3 TLS and Financial API

Under the revised *Payment Services Directive* (PSD2) [3], *Account Servicing Payment Service Providers* (ASPSP) are to provide an interface for third parties to access account information and perform operations (e.g., payments) on behalf of the account holder.

Bar Mitzvah

By exploiting the invariance weakness of the RC4 stream cipher, an attacker is able to retrieve the session cookie by guessing the LSBs (least significant bits) of the keystream. After a phase in which the attacker sniffs the connection between two parties, it detects a weak key usage and tries to exploit the weakness.

CVE: 2015-2808

CVSSv2 score: 4.3 (Medium)

Mitigation

Disable the RC4 stream cipher.

Code Snippet

1. open your Apache configuration file (default: `/etc/apache2/sites-available/default-ssl.conf`);
2. find the line starting with: **SSLCipherSuite**;
3. add the string `!RC4` at the end.

N.B. restart the server by typing: `sudo service apache2 restart`.

Fig. 2. TLSAssistant report for Bar Mitzvah - Apache webserver.

The Berlin Group standards and harmonization initiative proposes several possible approaches in its detailed “Access to Account (XS2A) Framework” [1], providing a detailed description of RESTful *Application Programming Interfaces* (APIs) and their usage for the purposes of authentication of involved parties and authorization to access service resources, such as account information, payment initiation, and confirmation of funds. The security of these APIs is based on both the transport and application layers. The first core technology explicitly identified by the guidelines is the TLS protocol: in particular, “the communication between the Third Party Provider (TPP) and the ASPSP is always secured by using a TLS-connection using TLS version 1.2 or higher.” [1].

Recommendations for security and identification standards in XS2A have been published by Open Banking Europe [21], which explicitly assume the use of TLS to guarantee confidentiality and integrity. The deployment of TLS is also assumed by the Financial-grade API (FAPI) specification [19] for authentication and authorization.

While PSD2 APIs are one of the most recent examples of online financial services relying on the security of TLS, there are several other examples such as home banking web and mobile applications; is also explicitly cited as an example of how to comply with the *Payment Card Industry Data Security Standard* (PCI-DSS) [20] Requirement 4.1, to “Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks”.

Beyond online services, TLS is deployed in many IoT devices [24] and client-end TLS proxies (e.g., in anti-virus products) [28]. Vulnerabilities in their TLS configuration and implementation can significantly downgrade the overall cyber and physical security, affecting many different enterprise sectors. For example, in the financial sector, the CCTVs are essential for the ATM surveillance.

3 FINSEC Platform Overview

FINSEC (Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures) is a Horizon 2020-funded project being developed by a consortium of 23 international partners [5]; its aim is to provide an integrated service platform for the cyber-physical security of critical financial infrastructures.

Figure 3 shows a simplified version of the FINSEC architecture [6], highlighting the main components related with TLSAssistant, which can be seen as a three-layered set of components with different roles:

Presentation Layer composed by a *Dashboard* and any *External Services* with access to the APIs. In particular, the *Dashboard* is an interactive interface developed within the FINSEC project capable of showing the overall security status of the infrastructure;

Service Layer contains the services integrated in the FINSEC platform. The services can be either called on-demand (like TLSAssistant) or have an ongoing monitoring activity (e.g., the Risk Assessment Engine);

Data Layer hosts a collection of security policies, vulnerabilities (imported CVE), system logs and other intel. It is where all the knowledge is stored and retrieved by upper layer entities. Its content is written in FINSTIX, a proprietary extension of the STIX language (see Sect. 4.1 and [7]).

From a developer’s perspective, the FINSEC platform has a microservice architecture. This means that each integrated service is completely virtualized, independent from the others and must therefore handle its own dependencies and deployment. In FINSEC, this has been achieved by using the containerization technology provided by Docker. Thanks to the high level of independence, this approach aims to simplify both the overall management and each maintainer’s work on condition that they handle their containers as if they were partially cloud-deployed. This because, while each service can maintain its own private state, it has to store all the relevant information in the shared database and handle its own access queue.

The communication among containers is managed through the use of REST APIs that each maintainer must provide. The set of APIs exposed by the services is called *SECaaS API* while the one provided to access the *Data Layer* is called *Data Access API*.

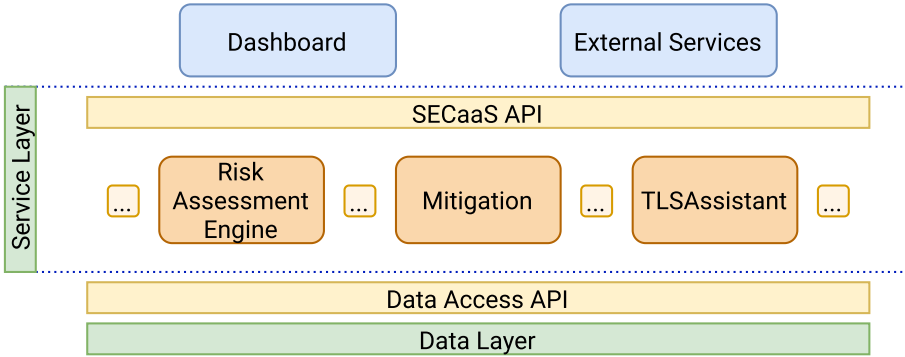


Fig. 3. Simplified FINSEC architecture.

4 Integration with FINSEC Platform

Having been developed outside the FINSEC context, *TLSAssistant* is the result of design choices driven by the use-case of a system administrator wanting to check his own TLS deployment by downloading *TLSAssistant* and being able to use it even when offline. This specific use case, notably different from the FINSEC platform, led to (i) the creation of an internal database containing the vulnerability mitigations and other information that help address the issues, (ii) the ability to run a single analysis at a time, and (iii) the generation of a human-readable report.

To maintain *TLSAssistant*'s autonomy while being able to satisfy all the FINSEC requirements, the integration was divided in three parts:

1. extension of *TLSAssistant*'s output capabilities to allow a STIX-generated report option;
2. creation of a *Connector* able to translate *TLSAssistant*'s STIX output in FINSTIX and to link our tool with the FINSEC's *Data Layer*, avoiding data redundancy and to maintain consistency across multiple analysis;
3. creation of a set of REST APIs and of a queue manager to allow concurrent requests to be served sequentially.

4.1 STIX Output In *TLSAssistant*

Sharing intelligence with automated services required producing structured data that are consumable by other services through a persistent data store. For this reason, from version v1.2, *TLSAssistant* is able to export the analysis result in STIX (Structured Threat Information eXpression) [18], a language used to share cyber threat intelligence (CTI) that can be represented with objects and their descriptive relationships. The *STIX Domain Objects* (SDOs) and *STIX Relationship Objects* (SROs) are visually summarized in Fig. 4.

An implementation of *TLSAssistant* with STIX version 2.0 is available on GitHub [25]. To clarify how the *TLSAssistant* report has been modified to map

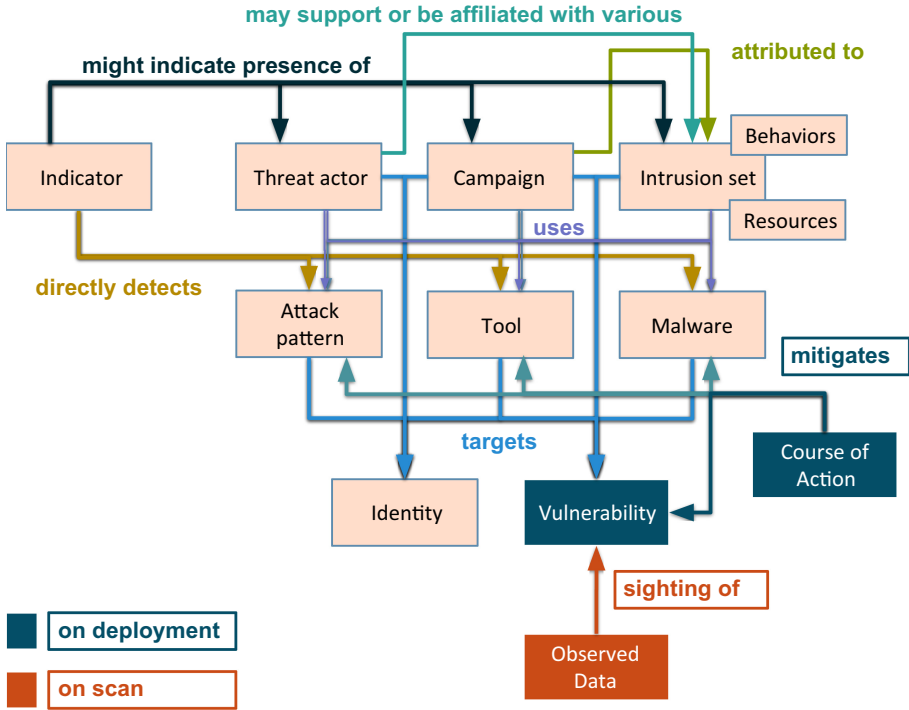


Fig. 4. A simplified SDO ecosystem with its possible relationships.

the vulnerability and mitigation output to STIX objects and relationships, Fig. 5 shows an example for the Bar Mitzvah attack. In general, after every scan and for each discovered vulnerability, TLSAssistant generates a JSON file containing the following entries:

vulnerability is an SDO that indicates a weakness that can be used by an attacker to compromise a system. The CVE ID [14] that provides a common name for known vulnerabilities is usually present in the `external_references` property. In TLSAssistant, the `vulnerability` SDO contains the name and description of the detected vulnerability.

course of action is an SDO used to give a recommendation on the actions that might be taken in response to a CTI. It describes technical responses (e.g., patches) or higher level actions (e.g., policy changes). In TLSAssistant, the `course of action` SDO contains the textual description of the mitigation (in the `description` field) and the actionable mitigation in the form of code snippets (in the `x_actions` custom field).

relationship is an SRO used to link together two SDOs or STIX Cyber-observable Objects (SCOs) in order to describe how they are related to each other. STIX defines many relationship types, e.g. `uses`, `targets`, `mitigates`.

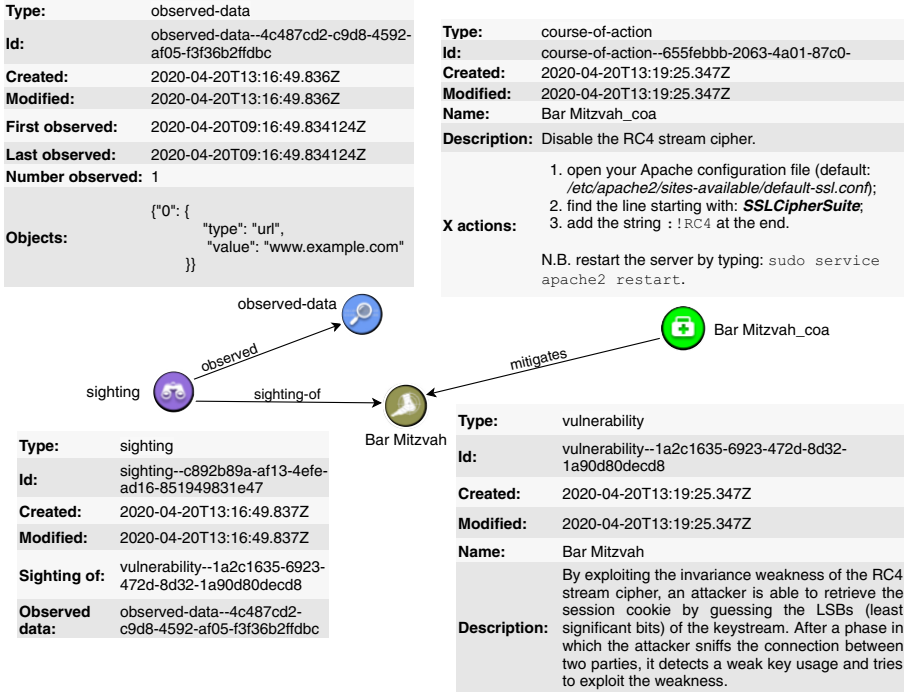


Fig. 5. STIX output for the Bar Mitzvah attack.

In TLSAssistant, a course of action is linked with a vulnerability through the mitigates relationship type.

observed data is an SDO that contains information about entities (e.g., files and systems) using the SCOs to provide supporting context. It is not an intelligence assertion, it is simply the raw information without any context for what it means. In TLSAssistant, the observed data SDO contains info about the asset observed (e.g., the URL analyzed and the timestamp of the scan).

sighting is an SRO that denotes the belief that something in CTI was seen. In TLSAssistant, the sighting SRO links the vulnerability SDO and the observed data SDO through the observed and sighting of relationship types.

4.2 FINSEC Connector

The Connector is a Python script written to integrate TLSAssistant in the FINSEC platform. Figure 4 shows its two modes of operation: on_deployment and on_scan.

on_deployment. The script is called once during the container’s deployment phase. To avoid data inconsistencies or duplicates, once invoked it checks if a

deployment had already occurred. In this mode, the connector has the role of linking TLSAssistant’s intelligence to the *Data Layer* content. In particular:

1. it exports TLSAssistant’s internal database. This will create a set of STIX bundles (see Sect. 4.1) containing three objects: a **vulnerability**, a **course of action**, and a **relationship** of type **mitigates**;
2. using the exported vulnerabilities, it retrieves their IDs from the *Data Layer* then edits all the **relationship** objects. By doing this, each **course of action** will be linked with the proper object within the shared database. Two edge cases can occur:
 - if a single **course of action** is able to mitigate more than one vulnerability, the connector will create an “aggregated” **vulnerability** object and link it to the mitigation (to avoid SRO redundancy);
 - if a vulnerability extracted from TLSAssistant does not have a CVE (hence the *Data Layer* will not contain its object), a new **vulnerability** object will be created and uploaded;
3. it extends the structure of each **course of action** by adding FINSTIX properties (see Sect. 3). These (e.g., ‘x_subtype’ = ‘to_dashboard’) are used to extend the STIX language and manage the integration with the *Dashboard*;
4. lists all the created and linked vulnerabilities in a file stored locally (this is the file whose existence will be checked on startup) and uploads all **course of action** and **mitigates** objects.

on_scan. The script is called after every completed scan. Once started, the connector retrieves the JSON files generated by TLSAssistant (see Sect. 4.1), and performs the following operations (for each file):

1. extracts the **sighting**, the **observed data** (see Fig. 5) and the name of the detected **vulnerability**;
2. matches the name of the vulnerability and links the **sighting** to its ID in the *Data Layer* (value retrieved or generated during the deployment);
3. tags each **sighting** object with a custom **scan_id** field so that the initiating service could retrieve the results;
4. finally, it uploads both the **sighting** and the **observed data** to the *Data Layer*, allowing the *Dashboard*, the initiating service, and any other services to retrieve the results of the scan, now available within the shared database.

4.3 API and Queue Handling

Each TLSAssistant scan can take anywhere between 1 and 5 min, and an installation of TLSAssistant was never designed to handle concurrent requests. Making the service available therefore required not only the definition of an API to initiate scans, but also a message queue for requests to be passed to worker threads, and a data store for states and results to be queried.

Figure 6 shows a component diagram of TLSAssistant integrated in the FINSEC platform, with our contribution (colored) composed of the following:

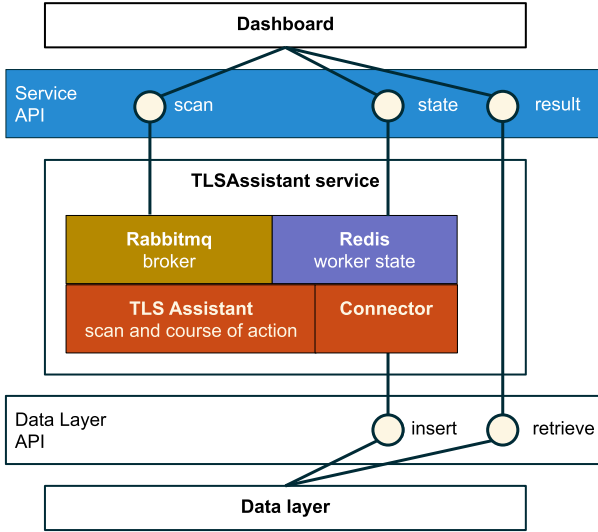


Fig. 6. Integration of TLSAssistant in the FINSEC architecture.

- TLSAssistant+*Connector* to analyze the deployments and interact with the *Data Layer* (see Sects. 2.2 and 4.2);
- a set of Service APIs exposed using Flask and leveraging the [Celery](#) task queue;
- `rabbitmq` broker for scan requests;
- `redis` to store scan states.

In detail, our service expose the following Service APIs:

POST `/scan` a JSON object with a `url` key on which to initiate a vulnerability scan; returns a `scan_id` - a UUID as per RFC 4122.

GET `/state/{scan_id}` returns the current scan state as reported by Celery (e.g. `pending`, `success` and `failure`).

GET `/result/{scan_id}` retrieves all `sighting` for the given `scan_id`, and all associated objects such as `vulnerability` and `course of action`, from the *Data Layer*.

Finally, enhancing the functionality of other services to achieve a synergy beyond their consumption of the generated STIX objects required individual integration, e.g. Risk Assessment - see Sect. 5.

5 Integration with Risk Assessment

As discussed in the introduction, one of the most important advantages in integrating a tool like TLSAssistant in the FINSEC platform is to enhance the functionality of other services. Particularly relevant to any security platform is to

provide a comprehensive evaluation of the security risks of an infrastructure or system. We thus consider how TLSAssistant can contribute information to refine the calculation of the Risk Assessment Engine (RAE) available in the FINSEC platform in order to support a continuous risk evaluation process.

The RAE is designed to support a continuous monitoring of assets; this allows a Security Officer at a financial institution to maintain a list, for instance, of TLS servers in their infrastructure, including but not limited to Online Banking or PSD2 API servers, and have them regularly scanned for vulnerabilities. While changes in server configurations should only occur infrequently and as a consequence of manual intervention by a system administrator, a regular scan also mitigates against undetected malicious changes, unintended consequences e.g. of upgrades to dependencies, or automated changes. The regular scans may return observations of vulnerabilities that trigger a re-evaluation of the current risk level. If the vulnerabilities are tied to a specific CVE, this can be cross-referenced with those potentially present on the asset by manufacturer and model, and its impact can be assessed based on CVSS scores.

The RAE integrated in the FINSEC platform [8] takes an approach based on graphical risk modeling described in [27]. To summarize, this entails the creation of a CORAS model, based on an understanding of the overall risk pattern to be modeled, and the definition of risk assessment algorithms - Bayesian networks in R and decision diagrams in DEXi - for an automated quantitative and qualitative risk assessment.

The RAE has been adapted to consume STIX `sighting` objects to trigger re-evaluation of risk models. As noted in Sect. 4.1, TLSAssistant produces `sighting` objects linked to cyber observables; the RAE however measures risks associated with an `x-asset` object, developed specifically for the FINSTIX data model [7]. This object links specific cyber-physical entities, e.g. servers, with their cyber-physical address (e.g. LAN IP) and/or physical location, as well as parameters relevant to risk assessment. A synergy between our vulnerability scanning service and the Risk Assessment Engine can be achieved by (a) linking each scan of an address with a known `x-asset`, on the part of the scanning service, and (b) adding `sighting` of the relevant `vulnerability` objects to the RAE risk models.

The following modifications were made specifically to integrate with the RAE.

`x-asset` objects in the *Data Layer* had a `url` property added

`scan/` API endpoint calls can be made with an `x-asset-id` JSON element; the API will retrieve the corresponding object from the *Data Layer* and read its `url` property.

`x-asset-id` provided to the API is added to the `x-asset_refs[]` property in `sighting` objects produced by TLSAssistant.

CORAS risk models were updated to include some of the specific CVE that TLSAssistant can produce `sighting` of. A simplified model of how `sighting` of `vulnerability` may be represented in CORAS is shown in Fig. 7.

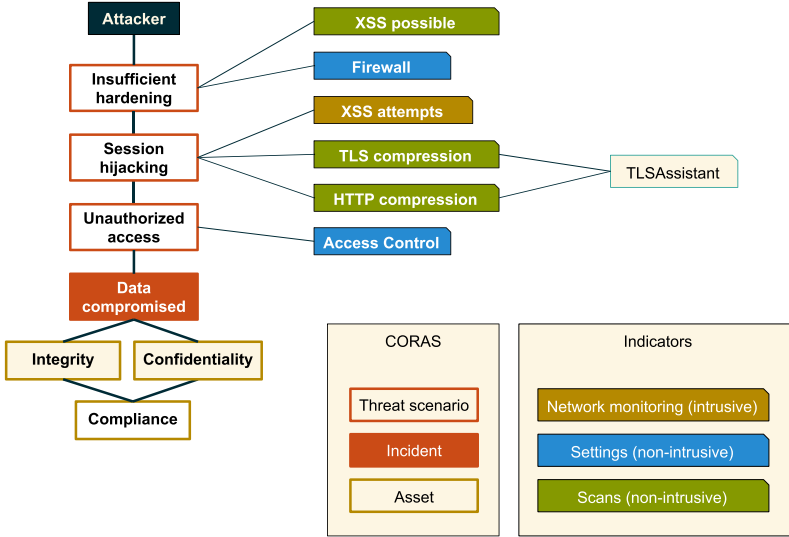


Fig. 7. Integration of TLSAssistant in a Risk Assessment Engine model.

6 Conclusions and Future Work

Combining different security services and merging their results in a coherent view presents some interesting challenges and opportunities. In this paper, we present how we addressed them in the integration in the FINSEC platform of an open-source tool (TLSAssistant) proposed to assist system administrators to deploy resilient instances of the TLS protocol. Firstly, for sharing TLSAssistant’s output with FINSEC services, we extended our tool with STIX 2.0, a language used to share structured cyber threat intelligence data. Secondly, as TLSAssistant was never designed to handle concurrent requests, we provided – together with the APIs to initiate scans and retrieve the results – an API to request the state of the scan and a data store for states and results to be queried. Finally, we showed how the TLSAssistant service integrated in the FINSEC platform can be easily modified to support the integration with other FINSEC services (e.g., the risk assessment engine).

As future work, we plan to further improve the TLSAssistant integration (e.g., by managing the edge cases) and the orchestration of TLSAssistant requests (e.g., by adding a scan scheduling to manage both on-demand and scans repeated on a regular basis).

Acknowledgments. This work has been supported by the research project “Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures” (FINSEC), which receives funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant agreement 786727.

References

1. Berlin Group: NextGenPSD2 Access to Account Interoperability Framework - Implementation Guidelines V1.3.4. <https://www.berlin-group.org/nextgenpsd2-downloads>
2. Böck, H., Somorovsky, J., Young, C.: Return of Bleichenbacher's oracle threat (ROBOT). In: 27th USENIX Security Symposium (USENIX Security 18), pp. 817–849 (2018)
3. European Parliament: Directive (EU) 2015/2366 of the European Parliament and of the Council on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC. <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015L2366&from=EN>
4. Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., Smith, M.: Why eve and mallory love android: an analysis of android SSL (in)security. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 50–61 (2012). <https://doi.org/10.1145/2382196.2382205>
5. FINSEC: Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures. <https://www.finsec-project.eu/>
6. FINSEC D2.5: FINSEC Reference Architecture II (October 2019). <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ce3a941d&appId=PPGMS>
7. FINSEC D3.9: Finance Sector Security Knowledge Base I (October 2019), <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c8e14437&appId=PPGMS>, due to be updated in Deliverable D3.10 in 2021
8. FINSEC D4.5: Risk Assessment Engine for Critical Infrastructures in the Financial Sector II (March 2020). <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ccabbc21&appId=PPGMS>, due to be updated in Deliverable D4.6 in 2021
9. IETF: The Transport Layer Security (TLS) Protocol: Version 1.3. <https://tools.ietf.org/pdf/rfc8446.pdf>
10. IMPERVA: Attacking SSL when using RC4. https://www.imperva.com/docs/HII-Attacking_SSL_when_using_RC4.pdf
11. Kario, H.: tlssfuzzer: SSL and TLS protocol test suite and fuzzer. <https://github.com/tomato42/tlssfuzzer>
12. Maddison, J.: Defining the security platform. <https://www.csoonline.com/article/3527843/defining-the-security-platform.html>
13. Manfredi, S., Ranise, S., Sciarretta, G.: Lost in TLS? no more! assisted deployment of secure TLS configurations. In: 33th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), pp. 201–220 (2019). https://doi.org/10.1007/978-3-030-22479-0_11
14. MITRE: Common Vulnerabilities and Exposures. <https://cve.mitre.org/>
15. Network Working Group: The TLS Protocol: Version 1.0. <https://tools.ietf.org/pdf/rfc2246.pdf>
16. Network Working Group: The Transport Layer Security (TLS) Protocol: Version 1.1. <https://tools.ietf.org/pdf/rfc4346.pdf>
17. Network Working Group: The Transport Layer Security (TLS) Protocol: Version 1.2. <https://tools.ietf.org/pdf/rfc5246.pdf>

18. OASIS Open: STIX - A structured language for cyber threat intelligence. <https://oasis-open.github.io/cti-documentation/stix/intro>
19. OpenID Foundation: Financial-grade API (FAPI). <https://fapi.openid.net/>
20. PCI Security Standards Council: Requirements and security assessment procedures v3-2-1. https://www.pcisecuritystandards.org/document_library (2018)
21. PRETA Open Banking Europe: Security and Identification Standards for APIs & Communications. <https://www.openbankingeuropa.eu/media/1398/oasis-obe-api-identification-and-security-standards-for-apis-and-communications.pdf>
22. Qualys: SSL Pulse. <https://www.ssllabs.com/ssl-pulse/>
23. Ronen, E., Gillham, R., Genkin, D., Shamir, A., Wong, D., Yarom, Y.: The 9 lives of Bleichenbacher’s CAT: new cache ATtacks on TLS implementations. IEEE Symposium on Security and Privacy, SP **2019**, 435–452 (2019). <https://doi.org/10.1109/SP.2019.00062>
24. Samarasinghe, N., Mannan, M.: Short paper: TLS ecosystems in networked devices vs. web servers. In: Financial Cryptography and Data Security - 21st International Conference, FC 2017, pp. 533–541 (2017). https://doi.org/10.1007/978-3-319-70972-7_30
25. Security & Trust Research Unit: TLSAssistant. <https://github.com/stfbk/tlsassistant>
26. Sheffer, Y., Holz, R., Saint-Andre, P.: Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). <https://tools.ietf.org/html/rfc7457.pdf>
27. Černivec, A., Erdogan, G., Gonzalez, A., Refsdal, A., Alvarez Romero, A.: Employing graphical risk models to facilitate cyber-risk monitoring - the WISER approach. In: Graphical Models for Security (GraMSec) 2017. LNCS, vol. 10744, pp. 127–146 (2018). https://doi.org/10.1007/978-3-319-74860-3_10
28. Waked, L., Mannan, M., Youssef, A.M.: The sorry state of TLS security in enterprise interception appliances. CoRR abs/1809.08729 (2018). <http://arxiv.org/abs/1809.08729>
29. Wetter, D.: /bin/bash based SSL/TLS tester: testssl.sh. <https://testssl.sh>



Cyber Threat Monitoring Systems - Comparing Attack Detection Performance of Ensemble Algorithms

Eva Maia¹ , Bruno Reis¹, Isabel Praça¹ , Adrien Becue²,
David Lancelin², Samantha Dauguet Demailly², and Orlando Sousa¹

¹ GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, School of Engineering of the Polytechnic of Porto (ISEP), Porto, Portugal

{egm,misr,icp}@isep.ipp.pt

² Airbus Defence and Space, Paris, France

Abstract. Cyber-attacks are becoming more sophisticated and thereby more difficult to detect. This is a concern to all, but even more to Critical Infrastructures, like health organizations. A Cyber Threat Monitoring System (CTMS), providing a global approach to detect and analyze cyber-threats for health infrastructures is proposed by combining a set of solutions from Airbus CyberSecurity with a machine learning pipeline to improve detection and provide awareness from cyber side to a more global approach that will combine them with physical incidents. The work is being carried out in the scope of SAFECARE project. In this work, we present the CTMS architecture and present our experimental findings with ensemble learning methods for intrusion detection.

Several parameters of six different ensemble methods are optimized, using Grid Search and Bayesian Search approaches, in order to detect intrusions as soon as they occur. Then, after the determination of best set of parameters for each algorithm, the attack detection performance of these six different ensemble algorithms using the CICIDS 2017 dataset are calculated and discussed. The results obtained identified Random Forest, LightGBM and Decision Trees as the best algorithms, with no significant difference in the performance, using a 95% confidence interval.

Keywords: Machine Learning · Intrusion Detection System (IDS) · Ensemble learning · Parameters optimization · Random Forest · Decision Trees · LightGMB · Adaboost · Rusboost · CICIDS2017

1 Introduction

Over the last decade, the European Union has faced numerous threats that quickly increased in their magnitude, changing the lives, the habits and the fears of hundreds of millions of citizens. The sources of these threats have been heterogeneous, as well as weapons to impact the population. Health services are

at the same time among the most critical infrastructures and the most vulnerable ones. They are widely relying on information systems to optimize organization and costs, whereas ethics and privacy constraints severely restrict security controls and thus increase vulnerability. The aim of SAFECARE project is to bring together the most advanced technologies from the physical and cyber security spheres to achieve a global optimum for systemic security and for the management of combined cyber and physical threats and incidents, their interconnections and potential cascading effects. SAFECARE cyber security solutions include a IT threat detection system, an Advanced file analysis system, a threat detection system for Building Monitoring Systems (BMS) and a E-health device security analytics system, all monitored in an overall Cyber Threat Monitoring System (CTMS) that feeds a data exchange layer, where all physical and cyber security incidents are analysed in a combined way through an impact propagation model. These detected incidents are then available through a Threat Response and Alert system, providing awareness to different stakeholders, from SOC operators to National health agency, Police, Firefighters, etc. In this paper, we will describe the CTMS and detail our experimental findings in using ensemble techniques for intrusion detection. The main objective of the IT Threat detection system is to improve network traffic incident/threat detection and investigation. Machine Learning methods have been widely used for this analysis since it can understand the behavior of an attack using known traffic datasets, and then can detect attacks in the network. The first experimental findings we will share in this paper rely on the application of Ensemble learning techniques. Ensemble learning [4] is a machine learning paradigm where multiple models are trained independently to solve the same problem, and then combined to get better results. As they often outperform single models for many types of problems [16], this technique has been widely used in intrusion detection [8, 11, 15]. Six different ensemble algorithms were considered for this study. In order to determine the best set of parameters for each ensemble algorithm, different parameter optimization techniques (Grid Search and Bayesian optimization) were applied. Then, after the determination of best set of parameters for each algorithm, their attack detection performance was calculated using a macro averaged score of the recall, precision and F1-score metrics. This way it is possible to understand the performance of each algorithm but also the trade-offs between precision and recall, so important in the attack detection field. The results showed that Random Forest and LightGBM are the best algorithms with no significant difference in the performance. The rest of the algorithms can be classified according with its attack detection performance in the following order: Decision Tree, Rusboost, Balanced Random Forest and Adaboost.

2 Cyber Threat Monitoring System

One of the main outcomes of the SAFECARE project is a cyber threat monitoring system (CTMS) that aims at improving the detection of Advanced Persistent Threats (APTs) [2] and zero-day attacks on IT and BMS systems, as

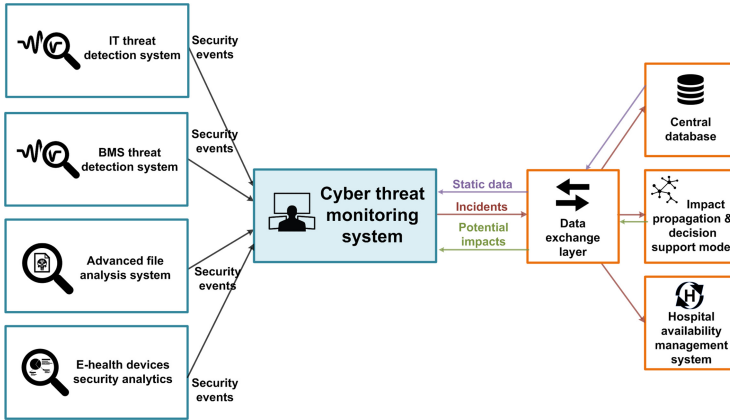


Fig. 1. SAFECARE - CTMS global structure.

shown in Fig. 1. The e-health device analytics solution collects data from medical devices, combines it with other (public) data sources and performs analytics to derive meaningful security data to help identify, assess and manage threats and risks affecting the e-health devices. The advanced file analysis system performs an in-depth analysis of the files extracted from network traffic by the IT and BMS threat detection systems, and the cyber threat monitoring systems receives all security events produced by the IT and BMS threat detection system. The advanced file analysis system detects malicious files based on different approaches: static analysis such as signature matching, heuristic analysis and dynamic analysis [3]. Signature matching is a deterministic method only effective if the malware is already known. Heuristic methods are able to identify several variants of a virus but can generate false-positive events. Dynamic analysis consists in sand boxing the file in a suitable environment in order to highlight suspicious behaviours. IT threat detection system supports the objective of improving incident detection by providing network monitoring and producing relevant information. The IT threat detection system captures the network traffic and performs near-real time analysis in order to detect suspicious behaviour and scale up security events to the cyber threat monitoring system. The IT threat detection system uses a hybrid approach combining both non-supervised methods and supervised methods in order to improve threat investigation and threat detection on the network traffic. The current solution is based on the network threat detection engine Suricata and is associated with a correlation engine Graylog. A machine learning module includes different techniques combined in a hybrid approach. In this paper we study, analyze and compare the performance of Ensemble Techniques and our experimental findings based on the training with public datasets.

3 Ensemble Methods in Intrusion Detection

Ensemble models are particularly adept at dealing with large datasets due to the ability to partition the data to train different classifiers with different samples and combining their results. Diversity is also essential since it allows classifiers to correct each other's errors. Random Forest, Adaboost, Rusboost and LightGBM are some of the most used ensemble methods in intrusion detection systems. Decision trees are the basic method of all these ensemble methods.

Stefanova and Ramachandran [20] proposed a two-stage classifier for network intrusion detection. Experimental tests led the authors to state that the approach presented is a superior method compared to the existing data mining models in network security, since the time for performing the analysis is relatively short, and the accuracy is remarkable.

Due to its nature, intrusion detection evaluation datasets are composed by imbalanced data where the proportion of attack events across all data is not evenly distributed. Random Forest is more robust than other widely known methods when considering imbalanced datasets. Even then, in some situations, the imbalance may influence the accuracy. Balanced Random Forest improves the ability of Random Forest models deal with imbalanced data [1].

Adaboost has been employed in several intrusion detection approaches but it is most commonly used in signature detection. Mazini et al. [14] proposed a hybrid solution where the Artificial Bee Colony (ABC) algorithm is used to optimize the search for the best feature space (feature selection) and Adaboost. M2 (a multiclass adaboost [23]) is used in a multiclass classification setting. In order to validate their results, the authors used False Positive Rate (FPR), recall and accuracy on the NSL-KDD and ISCXIDS2012 datasets. It was concluded that the proposed solution outperformed other methods with 99.61% detection rate, 0.01 FPR, and 98.90% accuracy.

Latah and Toker [13] presented a comparative study on the choice of an efficient anomaly-based intrusion detection method. The authors focused on supervised machine learning approaches, using several typical classifiers, such as decision trees, bagging trees, AdaBoost, Rusboost, etc. Using the well-known NSL-KDD dataset and based on experimental studies, the authors concluded that decision trees approach shows the best performance in terms of accuracy, precision, F1-score, area under the curve and McNemar's test. In addition, approaches like bagging trees, AdaBoost and Rusboost outperformed other conventional machine learning methods with a confidence level over 99.5%. Yulianto et al. [22] improved AdaBoost-based IDS Performance on CICIDS 2017 Dataset. The method proposed by the authors outperforms the performance of previous works with the accuracy of 81,83%, precision of 81,83%, recall of 100%, and F1 Score of 90,01%.

4 Datasets

Data are the most valuable asset to develop an efficient intrusion detection system. In the literature, exist several public available datasets that are intended

to resemble real data traffic, such as KDD-99 [5], DARPA 98/99 [21] and ISCX2012 [19]. In this work we will use the CICIDS2017 dataset [6], which was created to overcome the issues of existing datasets. This dataset is available in eight different (csv) files, containing five days of normal and 3 days of intrusive traffic.

Despite ensemble learning techniques are capable of handling large datasets, we have decided to sample the dataset for training and testing purposes, since it is more efficient and cost-effective than surveying the entire dataset. Thus, 30% of the data were chosen using a stratified sampling method, i.e., the new dataset had the same proportion of attacks and benign traffic as the complete dataset. Nevertheless, this proportional cut affected attacks with less representation, namely infiltration, heartbleed, and SQL injections, which had less than 40 instances each. To solve this problem it was decided to include, all these attacks in the new dataset. It is important to note that this addition changed the original distribution of traffic, and can create an undesired bias in the data. However, this bias it would be similar to the bias present in any other oversampling or undersampling method. Moreover, since we are only resample a small amount of instances (around 100 in total) the bias is even less significant. Therefore, it can be said that the dataset was sampled twice, in a first instance it was sampled to 30%, using a stratified method, making the underrepresented classes almost nonexistent; in a second instance all the underrepresented classes were included. Lastly, before starting the training phase, the new dataset was partitioned in train (77%) and test (23%) sets using again a stratified sampling strategy.

5 Ensemble Optimization

Most machine learning algorithms have several parameters that should be adjusted properly, otherwise the selected algorithm will not achieve optimal results. Several studies have been successfully proposed in parameter optimization to obtain the most accurate classification models [7]. In this paper, we mainly used three different methods: Grid Search, Bayesian Search and manual tuning. Grid Search is an exhaustive search based on defined subsets of the parameter space. Ramadhan et al. [18] applied the Grid Search method for tuning parameters in the well-known classification algorithm Random Forest. Also in this work, we have tuned Random Forest, Rusboost and Decision Tree parameters using Grid Search. The process was done as follows: first, a range to test the parameters was chosen by examining previous works and estimating the boundaries; second, a Grid Search was applied using the boundaries and some values in between. This process was accompanied by a 3-fold cross-validation to ensure that the results were stable in the training data and did not depend on random chance. k-fold cross-validation is a re-sampling procedure used to evaluate machine learning models, increasing the consistency and the quality of the results. In this procedure the training test is divided into k non-overlapping parts of equal proportion (randomized). Following this division k models are trained, each using its part as a validation set, and the other $k - 1$ parts as a

training set. Finally, all k models' performance is averaged to give a final result without overfitting the test set.

A common alternative to Grid Search is Bayesian Optimization which could be employed when the number of parameters and, consequently, the computational cost of doing a Grid Search are high. Bayesian optimization is an iterative algorithm. In each iteration, a probabilistic surrogate model is fitted to all observations of the target function made so far. Then, an acquisition function, which uses the predictive distribution of the probabilistic model, determines the utility of different candidate points, trading off exploration and exploitation [9]. We will use Bayesian optimization to tune LightGBM parameters. In the following sections we will describe the tuning of each individual algorithm as well as the parameters that are being tuned.

5.1 Random Forest and Balanced Random Forest

The optimization of parameters for Random Forest and Balanced Random Forest are very similar. Moreover the parameters that needs to be tuned are exactly the same:

- **Number of trees:** number of trees that are part of the Random Forest;
- **Max depth:** maximum depth of each tree;
- **Max features:** number of features that should be used to train each tree;
- **Splitting Criterion:** the criterion to test the quality of the splits.

Random Forest is a bagging method that ensembles several decision trees. So, as for any other bagging method, random samples of training sets (bootstrap samples) for each Random Tree are produced. The number of trees is then an important parameter to analyse. Thus, we built two plots (Fig. 2). As can be seen, the number of trees does not increase the F1-score beyond a small number of trees, in this case only about 100 trees are needed for performance improvements to be halted. This behavior is expected and in accordance with the literature [17]. According to the results, we can conclude that if F1-score was the only performance metric of interest any number of trees greater than 50 would be an acceptable choice. However, in the case of anomaly detection, execution time is also a concern. By analyzing the plot on the right (see Fig. 2), it is possible to

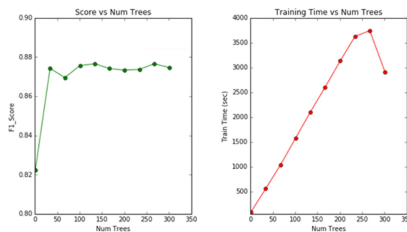


Fig. 2. Plot of F1-score (left) and execution time (right) against the number of trees.

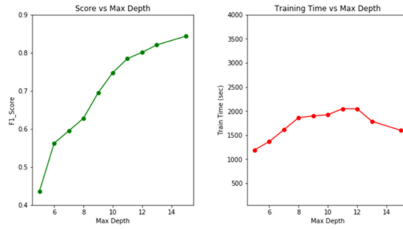


Fig. 3. Plot of F1-score (left) and execution time (right) against the maximum depth of the trees.

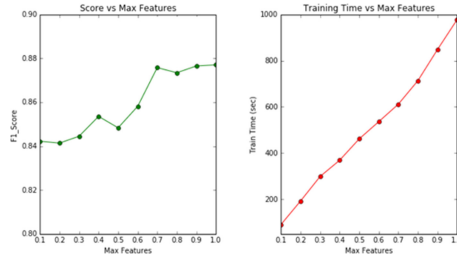


Fig. 4. Plot of F1-score (left) and execution time (right) against maximum number of features.

conclude that time and number of trees are positively correlated, which suggests, together with the graph on the left, that the ideal number of trees necessary to reach a plateau in performance could be any number above 50 trees. However, to account for the inherent randomness of the process 100 trees will be considered.

Another important parameter to tune is the depth of each individual tree, which is the only mechanism that Random Forest has to control the bias. Decision trees when grown sufficiently deep have relatively low bias. However, this can bring high variance, since the learned parameters, such as the structure of the decision tree, will vary considerably with the training data. The problem is that the model learns not only the actual relationships in the training data,

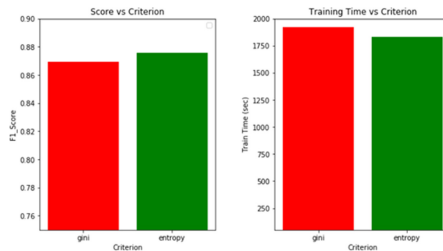


Fig. 5. Pilot of F1-score (left) and execution time (right) against the splitting criteria.

but also any noise that is present. In Random Forest, this variance is controlled by the bagging of the various trees, which means that by averaging out each prediction it is possible to reduce the error of the model on the test data, comparatively to the error of a single tree. Thus, the entire forest will have lower variance but not at the cost of increasing the bias. As can be seen in Fig. 3, the performance improves with tree depth, and execution time stays somewhat stable for each value. This indicates that growing trees to the maximum depth (until all leaves are pure) has a small computation cost but can bring good performance improvements. In the Random Forest only a subset of all features are considered for splitting each node in each decision tree. Therefore, the maximum number of features to use in every single split is one of, if not the most, important parameter in a Random Forest. With the analysis of the Fig. 4 it is possible to observe that the performance plateaus after 70%. This means that the algorithm could only produce meaningful splits with a high number of features probably due to a large amount of noise variables. This type of values point to the need to select the more relevant features.

Finally, a not so important parameter when building a Random Forest is the splitting criteria. Although the small importance for Random Forest due its ensemble nature, split criterion is a fundamental issue in decision trees. There are two main criteria: Gini Impurity and Information Gain (entropy). The Gini Impurity represents the probability that a randomly selected sample from a node will be incorrectly classified according to the distribution of samples in the node [12].

Changes in splitting criteria parameter rarely cause significant performance differences in Random Forest. Figure 5 shows that using Gini Impurity results slightly lower performance and slightly longer training time. In this case entropy was chosen, but the choice of Gini Impurity would not significantly affect the final results. Therefore, the final parameters chosen were:

- **Number of trees:** 100;
- **Max depth:** None;
- **Max features:** 70%;
- **Splitting Criterion:** Entropy.

Note that the **Max depth** parameter is None in order to expand the nodes to the maximum depth.

5.2 Decision Tree

In the case of decision trees, the parameters are very similar to those of Random Forest: **Max depth**, **Max features** and **Splitting Criterion**. This happens because the parameters used in Random Forest focus on controlling each individual tree instead of controlling the interaction between them. As such the descriptions will be similar to those used in the previous section.

In Fig. 6, it is easily observable that the maximum F1-score occurs when the maximum depth is about 17. Training time stabilizes when the maximum depth

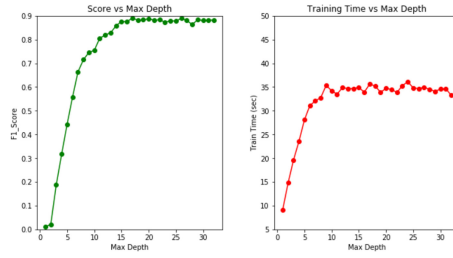


Fig. 6. Plot of F1-score (left) and execution time (right) against maximum depth.

approaches 10. This means that, contrarily to Random Forest choosing a higher maximum depth will not increase training time. However, there is no mechanism for reducing the variance of deep trees, which can make deep trees to outperform training data and to perform poorly on test data. As such, deep trees are not recommended and a depth of around 17 should be enough to meet performance needs.

Figure 7 presents the graphs for maximum number of features. As in Random Forest case, it is possible to notice two plateaus in the both functions, one between 40% and 60% and another between 80% and 100% features. This suggests a high number of noise variables, which indicates that using a low number of features, such as 10% or 20%, would highly contaminate the samples, leading to bad splits and consequently to poor results.

Interestingly, the best splitting criterion in decision trees is not the same as Random Forest. As can be observed in Fig. 8, although the training time is very similar, the criterion that leads to the best F1-score is Gini Impurity instead of entropy, with a 2% increase. Despite it seems very negligible, it can make a difference by reducing metrics like false positives and false negatives, that already have very low values. Thus, the final parameters for decision trees are:

- **Max depth:** 17;
- **Max features:** 80%;
- **Splitting Criterion:** Gini Impurity.

5.3 Adaboost and Rusboost

Adaboost and Rusboost, due to its nature, have almost the same parameters that can be tuned:

- **Max Number of estimators:** number of trees to fit in the model (can be reduce by early stopping);
- **Learning rate:** Rate of convergence of evaluation metric (higher converges faster);
- **Max depth:** depth of each boosted tree;
- **Sampling strategy:** which classes to undersample (exclusive to rusboost).

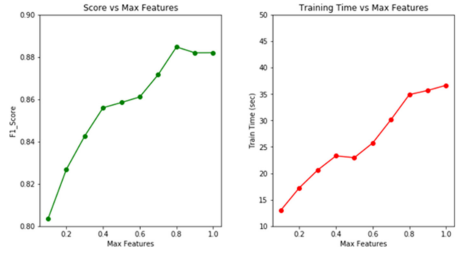


Fig. 7. Plot of F1-score (left) and execution time (right) against maximum number of features.

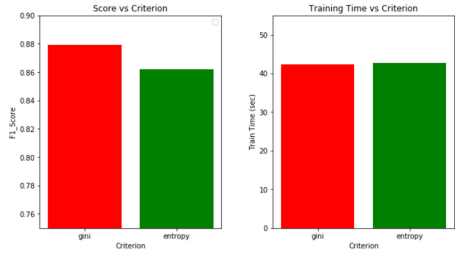


Fig. 8. Plot of F1-score (left) and execution time (right) against splitting criteria.

The number of estimators of a boosting model is one of the most important parameters. It represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably. Moreover, the number of estimators or trees needs to be balanced with the learning rate to tune the model.

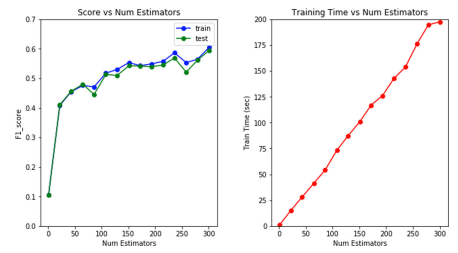


Fig. 9. Plot of F1-score (left) and execution time (right) against number of estimators.

In Fig. 9, it can be seen that for a given learning rate (0.05) the F1-score seems to increase monotonically when the number of trees is increased. This means that the F1-score increases rapidly in the early stages and stabilizes later. Therefore, choosing the number of trees is not so important as long as it is not too small or too high (may result in overfitting). A good starting point would be around 100 trees.

Another extremely important parameter is the learning rate or shrinkage. This parameter can be interpreted as the contribution scale of each tree to the prediction. This means that when there is a low shrinkage the model takes longer to converge, and as such will need more trees. Therefore, the learning rate and the number of trees are often seen together, as a tradeoff: when one is decreased the other must be increased, or the model is at risk of overfitting. A good way to ensure stability is to choose a low learning rate value and then select the number of trees. Another reason for adjusting shrinkage first is the computational cost. Usually, the algorithm looks for the convergence point when training. Thus, if the shrinkage is selected beforehand, the algorithm can select an optimal number of trees, reducing the computational cost.

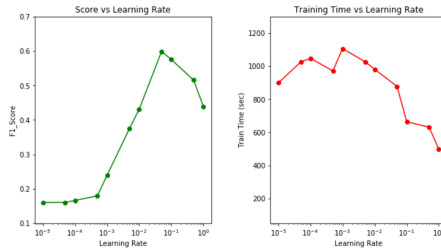


Fig. 10. Plot of F1-score (left) and execution time (right) against learning rate.

Figure 10 shows that shrinkage does not improve after 0.05, due to the low number of estimators (100), that limits the extent to which the convergence can be delayed. If the number of trees was increased, the learning rate would be expected to decrease. In terms of training time the model has erratic computational costs with expected monotonic trends when randomness is accounted for.

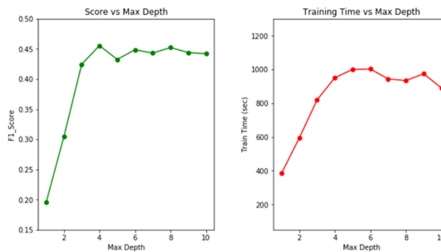


Fig. 11. Plot of F1-score (left) and execution time (right) against max depth.

Normally, the depth of the boosted trees is not a concerning fact when training a boosting algorithm. Tree stumps (trees with one root and two leaves) are often good models for weak learners and rarely need adjustment. When using trees deeper than one, there is a risk of increasing the variance, since models make fewer mistakes in each iteration. Thus, it is hard to train subsequent models using the misclassified samples of the previous model. Therefore, Hastie et al. [10] recommend a depth between 3 and 7, and mention that rarely more is needed and may result in a significant variance addition. As can be seen in Fig. 11, the model seems to stabilize the depth around 5. However, the score would be expected to fall if the depth continued to increase more than the chart boundaries. Finally, the training time follows a similar function and plateaus the depth around 5.

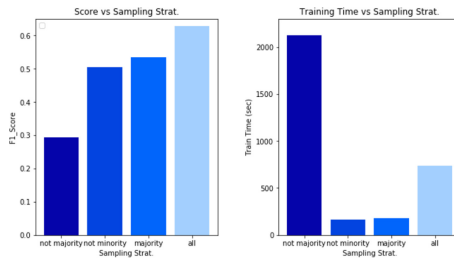


Fig. 12. Plot of F1-score (left) and execution time (right) against sampling strategy.

In the case of RusBoost there is one more parameter to take into account: the sampling strategy. Sampling strategy dictates which of the classes will be undersampled in each iteration of the boosting algorithm. There are 4 options for undersampling: undersample the majority class, all classes but not the majority class, all classes but not the minority class and finally it is possible to undersample all classes. In the case of intrusion detection, it is intuitive to want to undersample the majority class and leave the few samples of attack without any sampling. However, in the left plot of Fig. 12, it is possible to observe that the most successful undersampling technique is not *majority* as expected but undersampling every single class. This type of sampling is unexpected and as such would need a more thorough study. Finally, when it comes to the computation cost undersampling every class but the *not majority* displays the worst performance with around 200% more training time when compared to undersampling every class. Therefore, in this case the final parameters are:

- **Max Number of estimators:** 150;
- **Learning rate:** 0.05;
- **Max depth:** 5;
- **Sampling strategy:** all.

5.4 LightGBM

In the case of gradient boosting machines like LightGBM the tuning of parameters is not as straight forward as in the algorithms previously described. The solution is use Bayesian optimization, which uses an underlying gaussian process as a heuristic to find the parameters that optimize a black box model, i.e., it constructs a posterior distribution of functions (gaussian process) that best describes the function that needs to be optimized. There are many parameters that can be used in Bayesian optimization, nonetheless, the most important ones are: the number of iterations and the number of random explorations. The first dictates how many steps the model will execute to try to find the best parameters and the second tries to prevent the process from getting stuck in local minima by exploiting random solutions.

To find the best number of boosted trees a LightGBM model with 600 max trees was ran using 10-fold cross-validation together with an early stopping parameter of 200 iterations. It was possible to conclude that in all folds the model did not improve after 350 trees. As such, this number was chosen as an immutable parameter in the optimization process, along with other LightGBM parameters such as the objective function as *multiclass*, the number of classes as 15, class weight as *balanced* and bagging frequency as 5.

The parameters eligible for optimization were:

- **Bagging Fraction:** percent of data to sample (a value of 1 means all the data are used at each iteration, so no bagging is used), ranging from 0.8 to 1;
- **Feature Fraction:** fraction of features to select in each iteration, ranging from 0.1 to 0.9;
- **Max depth:** depth of each boosted tree, ranging from 5 to 9;
- **Number of leaves:** maximum number of leaves of each boosted tree, ranging from 50 to 80 leaves;
- **Learning rate:** rate of convergence of evaluation metric (higher converges faster), ranging from 0.001 to 0.1.

Table 1 presents the results of Bayesian optimization for these parameters. The line with the best results is highlighted.

Table 1. Bayesian optimization of LightGBM model.

Iteration	F1 Score	Bagging Fraction	Feature Fraction	Learning Rate	Max Depth	Number Leaves
0	0.918	0.875	0.861	0.073	7.0	55.0
1	0.908	0.831	0.146	0.086	7.0	71.0
2	0.914	0.804	0.876	0.083	6.0	55.0
3	0.778	0.808	0.162	0.005	9.0	50.0
4	0.932	1.0	0.9	0.1	9.0	80.0
5	0.576	1.0	0.9	0.001	5.0	80.0
6	0.712	0.8	0.1	0.001	9.0	63.0
7	0.737	1.0	0.9	0.001	9.0	75.0
8	0.923	1.0	0.9	0.1	5.0	66.0
9	0.923	1.0	0.9	0.1	5.0	50.0
10	0.874	0.8	0.1	0.1	5.0	71.0
11	0.667	1.0	0.1	0.001	5.0	53.0
12	0.923	1.0	0.9	0.1	5.0	61.0

Bagging fraction is normally used to speed training and/or reduce overfitting, since it resamples the data for a specific iteration using only the percentage specified in the parameter. Since LightGBM is a gradient boosting machine implementation, it is needed to define how often boosted trees will be created in a bootstrap. In this case, 5 will be used, which means that every five trees one would be created using a bootstrap containing between 80% to 100% of the data. This bagging fraction parameter, as can be seen in the Table 1, was optimized to 1, i.e., 100% of the data in each bootstrap. This behavior is expected since the Bayesian process was optimizing for F1-score and not for training time.

Feature fraction behaves similarly to the **Max features** parameter in Random Forest. Its goal is to reduce the number of predictors used in each iteration to improve training times and reduce the variance across all trees. As can be seen in the Table 1, feature fraction parameter has a behavior similar to Max features in Random Forest, tending to higher values, 0.7 in the case of Random Forest, 0.9 in this case. This usually means that most features constitute noise and offer no additional predictive value.

Following the previous parameters, comes the parameter that most influences the model’s performance, the **maximum depth** of the tree. LightGBM uses a leaf-wise (best-first) tree growth, i.e., it chooses to grow the branch whose split leads to maximum reduction of impurity. By doing so, LightGBM can often achieve better results than methods that use depth-wise growth. Nevertheless, the algorithm still provides a maximum depth parameter, which means that the leaf-wise growth is limited in height by this parameter, in order to reduce overfitting by controlling the variance of each tree. Looking at Table 1 it can be seen that the maximum depth can vary between 5 and 9, with the maximum value being chosen (9). Another common parameter is the **number of leaves** which, contrarily to maximum depth parameter, regulates the growth of the tree in a leaf-wise manner. However, the objective of this parameter is the same, it is used to control overfitting. The analysis of Table 1 suggests again that this parameter should be maximized using 80 as the maximum number of leaves. This points to the potential increase in the model score, by increasing the number of leaves. However, the computational costs of increment the number of leaves would increase an already long training time of 8 min. Finally, the last parameter is **learning rate** which, as in Rusboost, is used to delay the convergence of the model, since the slower the model learns the less likely it is to get stuck in local minima. This parameter is easier to optimize when there are many trees. Thus, in this case, with 350 trees, only lower learning rates can be explored. As such, Bayesian optimization chose the highest value with which the model could converge in only 350 trees - 0.1 (see Table 1).

5.5 Experimental Results and Analysis

To better understand the performance of the algorithms, a macro-averaged score of the recall, precision and F1-score was calculated for each model (see Fig. 13). With this information it is possible not only to gain insight into the performance of the algorithm but also into the trade-offs between precision and recall.

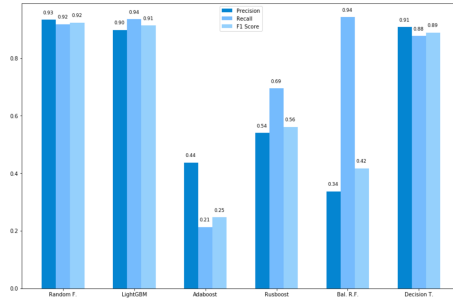


Fig. 13. Comparison of precision, recall and F1-score for all algorithms.

It is possible to notice that Random Forest, LightGBM and decision trees stand as the best performing models with 92%, 91% and 89% F1-score, respectively. This can be attributed, in the case of Random Forest and decision trees, to the depth of the trees used, since deepest trees achieve better results. In the case of LightGBM the trees depth is still high (9 levels), but the most important factor is the 80 leaf nodes that increase the method's ability to learn complex behavior together with boosting strategies. On the other hand, the rest of the boosting algorithms (Adaboost and RusBoost), performed poorly, attaining only 25% and 56% F1-score, respectively. This could be explained, as previously noted, for two reasons: first the depth of the trees, since both algorithms were modeled using only trees of depth 3 to not incur in unnecessary variance; and the low number of trees that did not provide enough iterations for the model to converge. As such, the solutions to these problems would be to increase the number of trees from 150 to 600, for example, to try to regulate the learning rate in order to converge the solution; and use deeper trees, preferably not exceeding 10 in depth, with a risk of performance degradation. Another remarkable property is the recall of the balanced Random Forest algorithm, which is highest among all algorithms along with LightGBM, despite having an exceedingly low precision. This indicates that the undersampling of the majority class had significant impact on reducing the number of false negatives but it also increased the amount of benign traffic labeled as an attack (false positives). This trade-off could be better regulated by adjusting the probability thresholds of the balanced Random Forest.

6 Conclusion

In this paper we present SAFECARE Cyber Threat Monitoring System and detail the IT threat detection systems, by comparing the attack detection performance of six different ensemble algorithms (Adaboost, Rusboost, Random Forest, Balanced Random Forest, Decision Tree and LightGBM) using the CICIDS2017 dataset. For each algorithm a study of the optimal parameter was made using Grid Search and Bayesian search approaches. After, the parameter optimization three different metrics were chosen to compare the algorithms

performance: recall, F1-score and precision. The results identified, using a 95% confidence interval, Random Forest, LightGBM and Decision Trees as the best algorithms with no significant difference in performance and the rest of the algorithms in the following order: Rusboost, Balanced Random Forest, Adaboost. The selected techniques are now deployed and being used for attack detection on data resultant from the simulation of different attacks under Airbus cyber range tool.

Acknowledgements. This work has received funding from European Union’s H2020 research and innovation programme under SAFECARE Project, grant agreement no. 787002.

References

1. Agusta, Z., Adiwijaya, A.: Modified balanced random forest for improving imbalanced data prediction. *Int. J. Adv. Intell. Inform.* **5**(1), 58–65 (2019). <https://doi.org/10.26555/ijain.v5i1.255>. <http://ijain.org/index.php/IJAIN/article/view/255>
2. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutor.* **21**(2), 1851–1877 (2019)
3. Aslan, A., Samet, R.: A comprehensive review on malware detection approaches. *IEEE Access* **8**, 6249–6271 (2020)
4. Brown, G.: *Ensemble Learning*, pp. 393–402. Springer, Boston (2017). https://doi.org/10.1007/978-1-4899-7687-1_252
5. University of California, I.: KDD cup 1999 data, March 2018. <http://mlexplained.com/2018/01/05/lightgbm-and-xgboost-explained/>
6. for Cybersecurity, C.I.: Intrusion detection evaluation dataset (CICIDS 2017), March 2018. <https://www.unb.ca/cic/datasets/ids-2017.html>
7. Dewancker, I., McCourt, M., Clark, S., Hayes, P., Johnson, A., Ke, G.: A strategy for ranking optimization methods using multiple criteria. In: *AutoML@ICML* (2016)
8. Dhaliwal, S.S., Nahid, A.A., Abbas, R.: Effective intrusion detection system using XGBoost. *Information* **9**(7) (2018). <https://doi.org/10.3390/info9070149>
9. Feurer, M., Hutter, F.: *Hyperparameter Optimization*, pp. 3–33. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_1
10. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd edn. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
11. Hu, W., Hu, W., Maybank, S.: AdaBoost-based algorithm for network intrusion detection. *IEEE Trans. Syst. Man Cybernet. Part B, Cybernet.* **38**, 577–83 (2008). <https://doi.org/10.1109/TSMCB.2007.914695>. A Publication of the IEEE Systems, Man, and Cybernetics Society
12. Koehrsen, W.: An implementation and explanation of the random forest in python, August 2018. <https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76>
13. Latah, M., Toker, L.: Towards an efficient anomaly-based intrusion detection for software-defined networks, March 2018

14. Mazini, M., Shirazi, B., Mahdavi, I.: Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *J. King Saud Univ. Comput. Inf. Sci.* (2018). <https://doi.org/10.1016/j.jksuci.2018.03.011>
15. Parr, T., Turgutlu, K., Csiszar, C., Howard, J.: Beware default random forest importances, March 2018. <https://explained.ai/rf-importance/>
16. Polikar, R.: Ensemble based systems in decision making. *IEEE Circuit Syst. Mag.* **6**, 21–45 (2006). <https://doi.org/10.1109/MCAS.2006.1688199>
17. Probst, P., Boulesteix, A.L.: To tune or not to tune the number of trees in random forest. *J. Mach. Learn. Res.* **18**(1), 6673–6690 (2017). <http://dl.acm.org/citation.cfm?id=3122009.3242038>
18. Ramadhan, M., Sitanggang, I., Rizky Nasution, F., Ghifari, A.: Parameter tuning in random forest based on grid search method for gender classification based on voice frequency. *DEStech Trans. Comput. Sci. Eng.* (2017). <https://doi.org/10.12783/dtcse/cece2017/14611>
19. Shiravi, A., Shiravi, H., Tavallae, M., Ghorbani, A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **31**, 357–374 (2012). <https://doi.org/10.1016/j.cose.2011.12.012>
20. Utic, Z., Ramachandran, K.: Network attribute selection, classification and accuracy (NASCA) algorithm for intrusion detection systems, April 2017. <https://doi.org/10.1109/THS.2017.7943463>
21. W. Haines, J., P. Lippmann, R., J. Fried, D., Zissman, M., Tran, E.: 1999 DARPA intrusion detection evaluation: design and procedures, p. 188, February 2001
22. Yulianto, A., Sukarno, P., Suwastika, N.: Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. *J. Phys. Conf. Ser.* **1192**, 012018 (2019)
23. Zhu, J., Rosset, S., Zou, H., Hastie, T.: Multi-class AdaBoost. *Stat. Interface* **2** (2006). <https://doi.org/10.4310/SII.2009.v2.n3.a8>



FINSTIX: A Cyber-Physical Data Model for Financial Critical Infrastructures

Giorgia Gazzarata^{1,2(✉)}, Ernesto Troiano³, Luca Verderame^{1,2},
Maurizio Aiello^{1,2,3,4}, Ivan Vaccari^{1,4}, Enrico Cambiaso⁴, and Alessio Merlo¹

¹ Department of Informatics, Bioengineering, Robotics and System Engineering,
University of Genoa, Genoa, Italy

² Consorzio Interuniversitario Nazionale per l'Informatica, Catania, Italy
{giorgia.gazzarata,luca.verderame}@dibris.unige.it

³ GFT Italia S.r.l., Genoa, Italy
ernesto.troiano@gft.com

⁴ Consiglio Nazionale delle Ricerche, IEIIT Institute (CNR-IEIIT), Genoa, Italy
{ivan.vaccari,enrico.cambiaso}@ieiit.cnr.it

Abstract. Cyber-physical security of financial institutions is a critical and sensitive topic. In this context, the FINSEC project aims to design and build a reference architecture for the integrated physical and cyber security of financial institutions. To make feasible, the interactions among the different services of the FINSEC platform, a proper data model defining the exchanged information semantic is fundamental. One of the objectives of the FINSEC project is to integrate cyber and physical security measures in the financial services industry. To do so, the data model must consider both cyber and physical systems. In this paper, the authors present FINSTIX, namely the data model adopted in the FINSEC platform. In particular, they extended the Structured Threat Information eXpression (STIX) standard creating custom objects to describe the financial organization's infrastructure and then to integrate cyber and physical security measures. The paper also reports an example of the use of FINSTIX in a relevant use case scenario.

Keywords: Data model · FINSEC · STIX · Cyber-Physical Threat Intelligence

1 Introduction

In the last few years, the number of cybersecurity incidents against financial institutions has been kept growing. According to the CLUSIT (the Italian association for cybersecurity) 2019 report, the number of financial attacks has increased by 33% from 2017 to 2018¹, thereby underlining how financial institutions are a primary target for cyber-attacks nowadays. This is mainly due to the growing sophistication of the IT technologies and the complex processes involving

¹ The report is available at the following link: <https://clusit.it/publicazioni/>.

multiple organizations. It is clear that financial institutions must increase their robustness against attack vectors. To deal with such a problem, the European Commission funded the Integrated Framework for Predictive and Collaborative Security of Financial Sector (FINSEC)² as an H2020 project. FINSEC aims to design and build a reference architecture for integrating physical and cyber security of financial institutions: in fact, in the financial services industry, cyber and physical security measures usually act in isolation, thus entailing inaccurate vulnerability assessment and risk analysis and, in general, poor-quality security guarantees.

Beyond architecture design, the definition of the semantic used to represent the information inside the FINSEC platform was one of the most demanding tasks. A proper data model is fundamental to:

- Enable the interactions between the FINSEC platform and third parties;
- Enable the interactions among the different services within the FINSEC platform;
- Provide the FINSEC services with a sufficiently fine-grained information granularity: in fact, too coarse-grained data may not be sufficient to support the activities of the services, while too fine-grained data may reveal to be unmanageable in actual cases;
- Enable the integration of cyber and physical security measures.

In this paper, the authors present the data model used by the FINSEC platform.

2 Background

2.1 FINSEC Reference Architecture

As shown in Fig. 1, the FINSEC core platform consists of different services divided into three distinct tiers:

- Edge Tier: it receives input data from the external probes and carries out activities on the probes, upon request from the upper layers. The probes are systems that collect and produce security information and events related to the organizations physical and cyber assets. Example of probes are logs, Security Information and Event Management (SIEM) systems, Closed Circuit Television (CCTV) systems and Simple Network Management Protocol (SNMP) agents;
- Data Tier: it stores information coming from the other tiers of the FINSEC platform or from external Cyber Threat Intelligence (CTI) sources;
- Service Tier: it contains the kernel applications and exposes the FINSEC functionalities to external parts. It consumes the information stored in the Data Tier.

² <https://www.finsec-project.eu>.

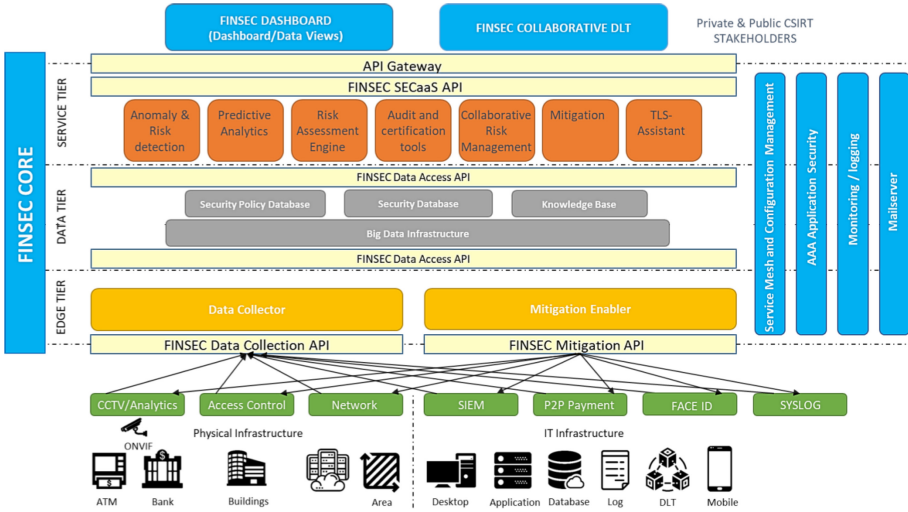


Fig. 1. The FINSEC platform Reference Architecture (logical view)

More details about the Reference Architecture are available in Deliverable 2.5 -“FINSEC Reference Architecture - II”³. In Sect. 3.1, the authors will present a practical example of the use of FINSTIX on a real use case scenario. There, the following architectural components will be considered:

- Probes: devices adopted in the financial field to collect cyber-physical information (e.g., CCTV probe, syslog probe, SIEM probe);
- Data Collector: collects data generated by the probes. Before storing them in the Data Layer, the Data Collector sanitizes data to remove eventual pieces of information violating the GDPR (personal references, etc.);
- Anomaly Detection: service that monitors events produced by the FINSEC probes in order to correlate them according to the models of attacks stored in the Data Layer as FINSTIX “x-attack”;
- Mitigation Service: service aimed to generate information that will be used to mitigate the attack detected by the Anomaly Detection;
- Mitigation Enabler: service acting on the field level to send notifications and to provide the probes with mitigation actions: some situations will require an active response of the platform onto the assets, such as locking a rack under attack, or shutting down a server that is threatened by a cyber attack.

³ <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ce3a941d&appId=PPGMS>.

2.2 STIX - Structured Threat Information eXpression

The Structured Threat Information eXpression (STIX) standard is a language and serialization format used to exchange CTI [4]. In particular, it was designed to support four use cases⁴:

- Analyzing Cyber Threats;
- Specifying Indicator Patterns for Cyber Threats;
- Managing Cyber Threat Response Activities;
- Sharing Cyber Threat Information.

STIX is a trademark of the MITRE Corporation, but it has been transitioned to OASIS, aiming to foster both the development of STIX and its adoption. STIX defines two kinds of objects: the STIX Domain Objects (SDOs) and the STIX Relationship Objects (SROs)⁵. The SDO corresponds to domain concepts commonly used in CTI, while the SROs are the relationships between the SDOs. From a graph standpoint, the SDOs are the nodes, while the SROs are the edges, as shown in Fig. 2.

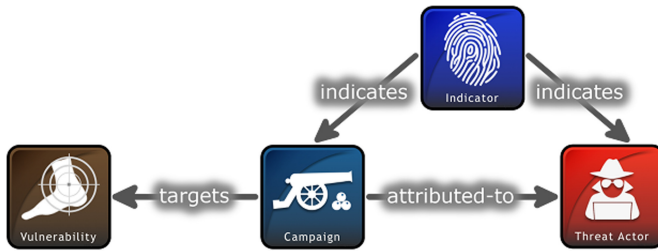


Fig. 2. SDOs and RDOs can be seen respectively as nodes and edges of a graph.

In STIX 2, SDOs and SROs are represented in JSON. More information on SDOs and SROs are available in the specification STIX™ Version 2.0. Part 2: STIX Objects⁶. The STIX standard enables customization by two different means: custom properties and custom objects. The first one consists in adding properties not defined by the specification to existing SDOs, while the second one consists in creating brand new objects. Independently from the customizations, STIX defines a set of requirements that must be enforced to preserve the conformity with the standard. For a description of such requirements, we refer the interested reader to the STIX™ Version 2.0. Part 1: STIX Core Concepts specification⁷.

⁴ STIX Use Cases web page, <http://stixproject.github.io/usecases/>.

⁵ STIX web page, <https://oasis-open.github.io/cti-documentation/stix/intro>.

⁶ STIX™ Version 2.0. Part 2: STIX Objects, <http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.html>.

⁷ STIX™ Version 2.0. Part 1: STIX Core Concepts, <http://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.html>.

2.3 State of the Art on the Use of STIX

CTI is a holistic approach to the automated sharing of threat intelligence [5]. Nowadays, it is considered one of the most promising strategies in the cybersecurity topic [10]. In the CTI context, [32] propose a classification and distinction among existing threat intelligence types. [16,23] instead summarize and compare the most prevalent information-sharing models adopted. Similarly, [24,27] propose a survey of the current platforms and formats available for threat information sharing. There are indeed different CTI formats available, like OpenIOC [15], Trusted Automated eXchange of Indicator Information (TAXII) [10,16], Real-Time Inter-network Defense (RID) [13,16], Incident Object Description Exchange Format (IODEF) [5,25]. Among the available CTI formats, Structured Threat Information Expression (STIX) [10] is considered the most commonly used CTI standard [29], although CyBOX and TAXII are considered good alternative solutions [7]. STIX provides a modular format that can also efficiently incorporate other standards [5].

STIX is adopted in different contexts of different nature. In this context, [18] adopts STIX as an input format for analyzing data for machine learning algorithms to increase new threat detection ability and responsiveness. In addition, [28] presents an innovative approach to automatically generate Cyber-Threat Intelligence data as STIX documents, starting from raw threat data. [19] adopts STIX to share threats and security information in IoT contexts, while [6,22] make use of a blockchain-based system to share CTI data using STIX format. [1] proposes an industrial adoption of STIX to exchange information between Integrated Management System (IMS) and Security Information and Event Management systems (SIEM). [20] presents an alternative use of STIX, to describe the actual state of the reference system, instead of exchange attack information. [34] presents a collaborative platform to share cyber threat information using STIX by focusing on anonymity exploitation. [26] makes use of STIX for threat information inputs, combining it with other similar information sources to develop a collaborative cognitive system, able to detect threats by combining different collaborative agents, covering both host and network information. Also, [11] combines STIX concepts with Markov chains ones, for cyber threats modeling. [9] proposes a cyber threat protection solution based on a Threat Intelligence Platform (TIP), based on both STIX and TAXII. Instead, [8] proposes MANTIS, a threat intelligence platform that makes use of different standards for threat data correlation, accomplished through a novel similarity algorithm. [17] proposes CyTIME, a framework that integrates CTI data like STIX under a global JSON format and automatically generates network security rules from the incorporated data seamlessly. Another innovative framework is proposed in [35], making use of STIX to exchange information about detected incidents, generated alerts, and applied mitigations. [14] introduces STIXGEN, a framework based on STIX able to generate meaningful, properly placed and error-free structured data. Although it is widely used, STIX presents different limitations: [3] analyses STIX by detailing the advantages and limitations of the format. STIX is indeed considered very complex to implement [16] and its

limits include the lack of support to reasoning [31]. In virtue of this, different extensions of STIX are proposed: UCO: A Unified Cybersecurity Ontology is a semantic-based alternative of STIX [31]. Also, [7] proposes some extensions of STIX, while [33] extends it to support the inclusion of relevant attack details on sophisticated attacks through the description of complex patterns. Similarly, while [30] extends STIX to support network and security events, [36] proposes a STIX extension to integrate and support additional cyber threats. Such extension is used in ChainSmith, a system able to extract Indicators of Compromise (IOC) by analyzing technical articles and industry reports.

The proposed work represents an extension of STIX in the fintech context. FINSTIX represents an innovative extension of STIX that considers both the cyber and the physical domain, thus modeling the entire financial infrastructure. Indeed, FINSTIX describes information like organization assets, how they are inter-connected, or monitoring probes and event types.

3 FINSTIX Cyber-Physical Data Model

Thanks to its expressiveness, flexibility, and extensibility, STIX is surely one of the most famous industrial standards used to represent and share Cyber Threat Intelligence. By the way, it has two weaknesses that prevent to use it as it is for the FINSEC purposes:

- It does not provide for an accurate representation of the financial institution infrastructure;
- It does not envisage physical systems; it is just limited to the cyber ones.

For those reasons, an extension to STIX was inevitable. The FINSEC extension to STIX 2 was driven by the FINSEC project use cases, which caused the introduction of some custom objects, among them:

- *Organization*: FDO used to represent an organization;
- *Area of Interest*: logical or physical area, such as a server room. An area of interest can be part of an organization or can be a sub-area of another area of interest;
- *Asset*: an organization valuable asset, such as a PC, an ATM, an application, or whatever is crucial for the organization. It can also be part of another asset;
- *Probe*: monitoring infrastructure that generates events and/or observed data. It can be related to one or more sensors, CCTVs, etc. A probe monitors one or more areas of interest;
- *Event*: event produced by a probe. Events, observed data, assets model and external CTI are used by the Analytics/Predictive modules to produce Cyber-Physical Threat Intelligence;
- *Attack*: cyber-physical attack. Differently from the Attack Pattern SDO, it also considers physical attacks, thus enabling an integration among cyber and physical scopes. An attack can be a sequence or a concurrency of events;

- *Cyber-Physical Threat Intelligence (CPTI)*: the result of the analytic tools. One or more CPTI objects are used to generate the output of the intelligence process, which is a report about ongoing or possible future attacks on one or more assets belonging to the infrastructure. This object is still a work-in-progress.

The resulting data model was named FINSTIX (from **FINSEC-STIX**). All the domain objects defined in FINSTIX, including those already introduced in STIX, take the name of FINSTIX Domain Objects (FDOs). The introduction of the FDOs *Organization*, *Area of Interest*, *Asset*, and *Probe* is due to the poor granularity of the *Identity* SDO. Instead, FINSTIX presents an ontology for a hierarchical representation of the organization infrastructure. As a consequence, any part of the infrastructure can be referenced during the Cyber-Physical Threat Intelligence process performed by the FINSEC platform. The introduction of *Event*, *Attack*, and *CPTI* is motivated by the need to cope with both the cyber and the physical domains. In fact, STIX is adequate for Cyber Threat Intelligence, but not for Cyber-Physical Threat Intelligence. The details of every custom object are not discussed here for reasons of space; however, the description of the FDOs is available in Deliverable 3.9 - “Security Knowledge Base”⁸.

Every custom object introduced in FINSTIX contains all the mandatory keys defined by STIX. In addition, FINSTIX defines the following mandatory keys for all the custom objects:

- *domain*: it can be “*Cyber*”, “*Physical*”, or “*Hybrid*”;
- *datatype*: it can be “*Model*” or “*Instance*”. When *datatype* is “*Model*”, then the object is used to define a concept, such as an event, an attack, a countermeasure, etc. Instead, if *datatype* is “*Instance*”, the object is used to represent something that actually happened/is happening;
- *x_organization*: contains the identifier of the organization that owns the information. This key was introduced for multi-tenant applications, in order to avoid data disclosure. *x_organization* is also a custom parameter for the objects already defined by STIX;
- *reference*: is the identifier of the parent object. It is used to generate a hierarchy of objects;
- *model_ref*: when *datatype* is “*Instance*”, *model_ref* contains the identifier of the object used to define the specific concept. The latter has the same *type*, but its *datatype* is “*Model*”.

Most of the custom objects introduced by FINSTIX are used to describe the financial organization infrastructure. Figure 3 shows a graph representing the infrastructure of an imaginary bank, SuperBank. The latter has an indoor area, which includes the entrance area and the ATM area. The indoor, entrance, and ATM areas are modeled through *Area of Interest* FDOs. In the ATM area,

⁸ <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c8e14437&appId=PPGMS>.

there is an ATM, which is composed by a vault and a computer. The ATM, the vault, and the computer are modeled through *Asset* FDOs. Finally, there is the monitoring infrastructure, which consists of four probes:

- Entrance CCTV: monitors the entrance area;
- ATM CCTV: monitors the ATM;
- Screen CCTV: monitors the person in front of the screen of the ATM;
- Network probe: monitors the ATM computer.

Each probe is also conceived as an asset. For this reason, it is modelled through both a *Probe* and an *Asset* FDOs. In particular, the *Probe* FDO refers to the *Asset* FDO, which in turns refers to an *Area of Interest* or to another *Asset*. For example, the *Probe* ATM CCTV refers to the *Asset* ATM CCTV, which refers to the *Asset* ATM. A concrete example of the use of FINSTIX inside the FINSEC Platform is presented in Sect. 3.1.

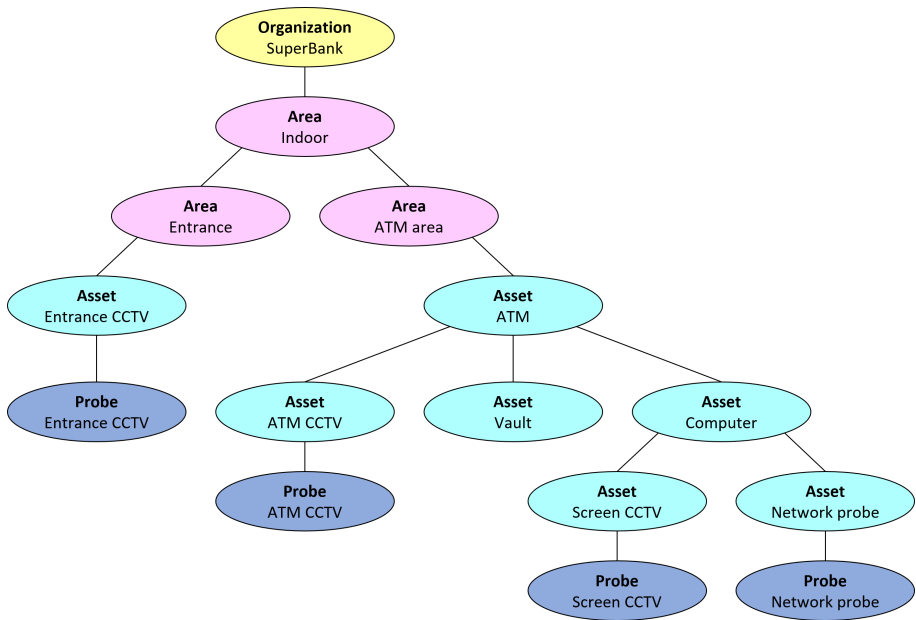


Fig. 3. Hierarchy of FINSTIX Domain Objects describing the infrastructure of a financial institution.

Events and observed data produced by the organization probes are pushed into the FINSEC platform, which correlates and aggregates information gathered from asset models and external CTI through the machine learning analytics and prediction algorithms. The result of this process is the Cyber-Physical Threat Intelligence, which integrates important information coming from both the cyber

and the physical world. The CPTI produced in the FinTech sector is the added-value information produced by the FINSEC platform that could be exchanged (in-out) between financial organizations and security organizations (CERT/C-SIRT like). The integration between cyber and physical security aspects introduced by FINSTIX is an innovation attributable to the FINSEC project.

3.1 Data Flow

This section shows the use of FINSTIX in a concrete use case scenario, in which the FINSEC platform detects a jackpotting attack to the organization SuperBank, whose infrastructure is described in Sect. 3. In particular, the use case describes how FINSTIX Domain Objects are used to exchange data between:

- The probes and the FINSEC Platform:
 - From the probes to the Data Collector;
 - From the Mitigation Enabler to the probes;
- The different services composing the FINSEC platform;
- The FINSEC platform and the Dashboard.

Step 1 and Step 2. In Step 1, the Network probe represented by the *Probe* FDO having id “*x-probe--network*” generates the instance of Event presented in Listing 1.1. The *Event* has “*x-event--cyber_instance*” as *id* and “*x-event--cyber_model*” as *model_ref*. The *Probe*, as well as all the FDOs used in this example, is owned by SuperBank, whose *id* is “*x-organization--superbank*”. The probe sends the *Event* to the Data Collector through the FINSEC Data Collection API. In Step 2, the Data Collector stores the *Event* in the Security Database through the FINSEC Data Access API.

Listing 1.1. Cyber *Event* generated by the Network probe

```

1 {
2   "type": "x-event",
3   "domain": "Cyber",
4   "id": "x-event--cyber_instance",
5   "created": "2020-06-22T20:02:45.028Z",
6   "modified": "2020-06-22T20:02:45.028Z",
7   "datatype": "Instance",
8   "name": "Cyber event",
9   "description": "Cyber event detected by network probe",
10  "x_organization": "x-organization--superbank",
11  "reference": "x-root--generic_root",
12  "model_ref": "x-event--cyber_model",
13  "coordinates": [41.899163, 12.473074],
14  "probe_ref": "x-probe--network",
15  "observed_refs": ["observed-data--netflow"],
16  "asset_refs": ["x-asset--ATM"]
17 }
```

Step 3 and Step 4. In Step 3, the CCTV probe generates an instance of *Event* having “*x-event--tampering_instance*” as *id* and “*x-event--tampering_model*” as *model_ref*, as shown in Listing 1.2. The probe sends the *Event* to the Data Collector through the FINSEC Data Collection API. In Step 4, the Data Collector stores the second *Event* in the Security Database through the FINSEC Data Access API.

Listing 1.2. Tampering *Event* generated by the ATM CCTV probe

```

1 {
2   "type": "x-event",
3   "id": "x-event--tampering_instance",
4   "datatype": "Instance",
5   "name": "Tampering",
6   "description": "CCTV camera tampering",
7   "model_ref": "x-event--tampering_model",
8   ...
9 }
```

Step 5 and Step 6. In Step 5, through the FINSEC Data Access API, the Anomaly Detection retrieves from the Security Database the instances of *Event* FDOs owned by “*x-organization--superbank*” and generated in a certain time window. As a result, in Step 6, the *Event* FDOs sent by the probes to the FINSEC platform in Step 1 and Step 3 are returned to the Anomaly Detection.

Step 7 and Step 8. In Step 7, the Anomaly Detection retrieves from the Security Database the models of attack owned by the organization identified by the *id* “*x-organization--superbank*”. As a result, in Step 8, the *Attack* FDOs matching the query are provided to the Anomaly Detection. The Anomaly Detection finds out that the *Attack* identified by the *id* “*x-attack--model*” has “*x-event--cyber_model*”, “*x-event--tampering_model*” as *event_refs*, as shown in Listing 1.3. These *Event* FDOs correspond to the models of the *Event* FDOs “*x-event--cyber_instance*” and “*x-event--tampering_instance*” respectively.

Listing 1.3. Model of *Attack* for the Jackpotting attack

```

1 {
2   "type": "x-attack",
3   "id": "x-attack--model",
4   "datatype": "Model",
5   "name": "Jackpotting",
6   "description": "Cyber-physical attack to send consecutive
7     money dispense command to ATM device to empty the
8     Cassettes",
9   "event_refs": ["x-event--cyber_model", "x-event--
10    tampering_model"],
11  "rules": [{...}, ...],
12  ...
13 }
```

Step 9 and Step 10. Consequently to Step 8, in Step 9, the Anomaly Detection analyses the *Observed Data* referenced by the *observed_refs* of the *Event* “*x-event--cyber_instance*” (namely “*observed-data--netflow*”). By using the rules described in the *Attack rules*, the Anomaly Detection detects something malicious. As a consequence, it generates the instance of *Attack* “*x-attack--instance*” presented in Listing 1.4, having “*x-attack--model*” as *model_ref* and “*x-event--cyber_instance*” and “*x-event--tampering_instance*” as *event_refs*. The *Attack* is then stored in the Security Database. The Mitigation Service listens to insertions of *Attack* FDOs and consequently receives the *Attack* “*x-attack--instance*” in Step 10.

Listing 1.4. Instance of *Attack* generated by the Anomaly Detection

```

1 {
2   "type": "x-attack",
3   "id": "x-attack--instance",
4   "datatype": "Instance",
5   "name": "Jackpotting",
6   "description": "Cyber-physical attack to send consecutive
7     money dispense command to ATM device to empty the
8     Cassettes",
9   "model_ref": "x-attack--model",
10  "event_refs":
11  ["x-event--cyber_instance", "x-event--tampering_instance"],
  ...
}

```

Step 11 and Step 12. The Mitigation Service needs to find a countermeasure for any attack modelled by “*x-attack--model*”. This information is contained in the *Cyber Physical Threat Intelligence (CPTI)* FDO. In Step 11, the Mitigation Service retrieves the *CPTI* characterized by “*x-attack--model*” as *attack_ref* from the Security Knowledge Base, through the FINSEC Data Access API. Then, in Step 12, the *CPTI* “*x-cpti--model*” presented in Listing 1.5 is returned to the Mitigation Service. From the *CPTI*, the Mitigation Service can find the *id* of the *Course of Action* necessary to mitigate the attack, namely “*course-of-action--model*”.

Listing 1.5. Model of *Cyber-Physical Threat Intelligence* for the Jackpotting attack

```

1 {
2   "type": "x-cpti",
3   "id": "x-cpti--model",
4   "datatype": "Model",
5   "name": "CPTI for jackpotting",
6   "description": "CPTI for jackpotting attacks",
7   "attack_ref": "x-attack--model",
8   "coa_refs": ["course-of-action--model"],
9   ...
10 }

```


Step 13 and Step 14. In Step 13, the Mitigation Service uses the *id* of the *Course of Action* learned in Step 12 (“*course-of-action--model*”) to request the FDO in Listing 1.6 to the Security Knowledge Base. In Step 14, the *Course of Action* “*course-of-action--model*” is returned to the Mitigation Service. The *Course of Action* is characterized by *x_subtype* set to *to_mail*: then the Mitigation Service knows that the mitigation consists in sending a mail notification. *x_actions* contains all the information regarding the mail to send, in particular the recipient address (*to*), the mail subject (*subject*) and the message (*body*). In the *body*, there are two markers, that must be replaced with information on the occurring attack, in particular the *id* of the asset involved in the attack and the timestamp of the attack. *markers* contains all the information needed by the Mitigation Service to perform this task.

Listing 1.6. Model of *Course of Action* used to mitigate the Jackpotting attack

```

1 {
2   "type": "course-of-action",
3   "x_subtype": "to_mail",
4   "id": "course-of-action--model",
5   "x_datatype": "Model",
6   "name": "CoA for jackpotting",
7   "description": "CoA to mitigate jackpotting attacks",
8   "x_actions": [{
9     "to": [{
10      "type": "fixed",
11      "address": "finsec@superbank.eu"
12    }],
13    "subject": "Jackpotting attack detected",
14    "body": "FINSEC system detected an activity attributable
15    to a possible jackpotting attack on the ATM identified by
16    %%asset_id%. The activity has been detected at %%time
17    %%".
18    "markers": [{
19      "marker": "%%asset_id%",
20      ...
21    }, {
22      "marker": "%%time%",
23      ...
24    }
25  ]
26 }

```

Step 15 and Step 16. The Mitigation Service has to generate an instance of *Course of Action* to mitigate the attack “*x-attack--instance*”. To do so, it needs to retrieve the instance of *Event* involved in the attack and modelled by “*x-event-cyber_model*”, to obtain the *id* of the asset involved. The Mitigation Service learned from the *Attack* “*x-attack--instance*” retrieved in Step 10 that the *Event* FDOs involved in the attack are identified by “*x-event--cyber_instance*” and

“*x-event--tampering_instance*”. As a consequence, in Step 15, the Mitigation Service retrieves from the Security Database the *Event* having “*x-event--cyber_instance*” or “*x-event--tampering_instance*” as *id* and “*x-event--cyber_model*” as *model_ref*. The Security Database returns the *Event* “*x-event--cyber_instance*” to the Mitigation Service in Step 16.

Step 17 and Step 18. In Step 17, the Mitigation Service generates the instance of *Course of Action* “*course-of-action--instance*”, whose *x_actions* contains the information needed to send the mail notification. The FDO is shown in Listing 1.7. Notice that the Mitigation Service replaced the markers contained in *body* with the *id* of the asset involved in the attack and the timestamp of the attack. The Mitigation Service inserts the FDO into the Security Database. In Step 18, the Mitigation Enabler, who listens to the insertion of instances of *Course of Action* FDOs into the Security Database, receives the FDO.

Listing 1.7. Instance of *Course of Action* used to mitigate the Jackpottting attack

```

1 {
2   "type": "course-of-action",
3   "x_subtype": "to_mail",
4   "id": "course-of-action--instance",
5   "x_datatype": "Instance",
6   "name": "CoA for jackpottting",
7   "description": "CoA to mitigate jackpottting attacks",
8   "x_model_ref": "course-of-action--model",
9   "x_organization": "x-organization--superbank",
10  "x_actions": [{
11    "to": ["finsec@superbank.eu"],
12    "subject": "Jackpottting attack detected",
13    "body": "FINSEC system detected an activity attributable
14            to a possible jackpottting attack on the ATM identified by
15            x-asset--ATM. The activity has been detected at
16            2020-06-22T20:03:05.142Z".
17  }],
18  ...
19 }
```

Step 19 and Step 20. Since in the *Course of Action* “*course-of-action--instance*” *x_subtype* is *to_mail*, the Mitigation Enabler knows that the mitigation consists in a mail notification. It then extracts *to*, *subject*, and *body* contained in *x_actions* and performs a request to the Mailserver. The latter finally sands the mail notification in Step 20.

4 Conclusions and Discussions

This paper presents the design of the FINSTIX data model as an extension of the Structured Threat Information eXpression standard version 2. In particular, FINSTIX introduces many custom objects to describe the financial organization’s infrastructure and integrates both cyber and physical security domains.

Moreover, the paper reports an example of the use of FINSTIX in a relevant use case scenario. The use case refers to the FINSEC platform. Albeit explicitly designed for the Finsec domain, the FINSTIX model could be easily extended to other critical infrastructures that require to model both cyber and physical security threats, e.g., cellular networks [21], industrial plants [2] or other financial services [12]. To the best of our knowledge, FINSTIX is the first data model coping with both cyber and physical domains and can thus be considered an innovation introduced by the FINSEC project.

Acknowledgements. This work has been supported by the following research project: Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures (FINSEC) project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant agreement no. 786727.

References

1. Abe, S., Uchida, Y., Hori, M., Hiraoka, Y., Horata, S.: Cyber threat information sharing system for industrial control system (ICS). In: 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp. 374–379. IEEE (2018)
2. Ackerman, P.: *Industrial Cybersecurity: Efficiently Secure Critical Infrastructure Systems*. Packt Publishing Ltd., Birmingham (2017)
3. Aviad, A., Wecel, K.: Cyber treat intelligence modeling. In: Abramowicz, W., Corchuelo, R. (eds.) BIS 2019. LNBIP, vol. 353, pp. 361–370. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-20485-3>
4. Barnum, S.: Standardizing cyber threat intelligence information with the structured threat information expression (STIX). MITRE Corporation
5. Burger, E.W., Goodman, M.D., Kampanakis, P., Zhu, K.A.: Taxonomy model for cyber threat intelligence information exchange technologies. In: Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security, pp. 51–60. ACM (2014)
6. Chia, V., et al.: Rethinking blockchain security: position paper. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1273–1280. IEEE (2018)
7. Fransen, F., Smulders, A., Kerkdijk, R.: Cyber security information exchange to gain insight into the effects of cyber threats and incidents. *e & i Elektrotechnik und Informationstechnik* **132**(2), 106–112 (2015)
8. Gascon, H., Grobauer, B., Schreck, T., Rist, L., Arp, D., Rieck, K.: Mining attributed graphs for threat intelligence. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, pp. 15–22. ACM (2017)
9. Ginn, R.J., Ionescu, I.: Cyber threat analysis (2017)
10. Gong, N.: Barriers to adopting interoperability standards for cyber threat intelligence sharing: an exploratory study. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) SAI 2018. AISC, vol. 857, pp. 666–684. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-01177-2_49
11. Gore, R., Padilla, J., Diallo, S.: Markov chain modeling of cyber threats. *J. Defense Model. Simul.* **14**(3), 233–244 (2017)

12. Guerar, M., Merlo, A., Migliardi, M., Palmieri, F., Verderame, L.: A fraud-resilient blockchain-based solution for invoice financing. *IEEE Trans. Eng. Manage.* **67**, 1086–1098 (2020)
13. Hazeyama, H., Kadobayashi, Y., Miyamoto, D., Oe, M.: An autonomous architecture for inter-domain traceback across the borders of network operation. In: 11th IEEE Symposium on Computers and Communications (ISCC 2006), pp. 378–385. IEEE (2006)
14. Iqbal, Z., Anwar, Z., Mumtaz, R.: STIXGEN-a novel framework for automatic generation of structured cyber threat information. In: 2018 International Conference on Frontiers of Information Technology (FIT), pp. 241–246. IEEE (2018)
15. Jaeger, D., Ussath, M., Cheng, F., Meinel, C.: Multi-step attack pattern detection on normalized event logs. In: 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, pp. 390–398. IEEE (2015)
16. Kampanakis, P.: Security automation and threat information-sharing options. *IEEE Secur. Priv.* **12**(5), 42–51 (2014)
17. Kim, E., Kim, K., Shin, D., Jin, B., Kim, H.: CyTIME: cyber threat intelligence management framework for automatically generating security rules. In: Proceedings of the 13th International Conference on Future Internet Technologies, p. 7. ACM (2018)
18. Kim, K., An, J.H., Yoo, J.: A design of IL-CyTIS for automated cyber threat detection. In: 2018 International Conference on Information Networking (ICOIN), pp. 689–693. IEEE (2018)
19. Ko, E., Kim, T., Kim, H.: Management platform of threats information in IoT environment. *J. Ambient Intell. Humaniz. Comput.* **9**(4), 1167–1176 (2018)
20. Leichtnam, L., Totel, E., Prigent, N., Mé, L.: STARLORD: linked security data exploration in a 3D graph. In: 2017 IEEE Symposium on Visualization for Cyber Security (VizSec), pp. 1–4. IEEE (2017)
21. Lewis, T.G.: *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*. Wiley, Hoboken (2019)
22. Li, J., Xue, Z.: Distributed threat intelligence sharing system: a new sight of P2P botnet detection. In: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–6. IEEE (2019)
23. Liu, M., Xue, Z., He, X., Chen, J.: Cyberthreat-intelligence information sharing: enhancing collaborative security. *IEEE Consum. Electron. Mag.* **8**(3), 17–22 (2019)
24. Lutf, M.: Threat intelligence sharing: a survey. *J. Appl. Sci. Comput.* **8**(11), 1811–1815 (2018)
25. Martinelli, F., Oslia, O., Saracino, A.: Towards general scheme for data sharing agreements empowering privacy-preserving data analysis of structured CTI. In: Katsikas, S.K. (ed.) *SECPRE/CyberICPS -2018*. LNCS, vol. 11387, pp. 192–212. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12786-2_12
26. Narayanan, S.N., Ganesan, A., Joshi, K., Oates, T., Joshi, A., Finin, T.: Early detection of cybersecurity threats using collaborative cognition. In: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 354–363. IEEE (2018)
27. Rattan, A., Kaur, N., Chamotra, S., Bhushan, S.: Attack data usability and challenges in its capturing and sharing
28. Sadique, F., Cheung, S., Vakulinia, I., Badsha, S., Sengupta, S.: Automated structured threat information expression (STIX) document generation with privacy preservation. In: 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (IEEE UEMCON 2018) (2018)

29. Shackleford, D.: Who's using cyberthreat intelligence and how? SANS Institute (2015)
30. Steinke, M., Hommel, W.: A data model for federated network and security management information exchange in inter-organizational IT service infrastructures. In: NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–2. IEEE (2018)
31. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A.: UCO: a unified cybersecurity ontology. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
32. Tounsi, W., Rais, H.: A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur.* **72**, 212–233 (2018)
33. Ussath, M., Jaeger, D., Cheng, F., Meinel, C.: Pushing the limits of cyber threat intelligence: extending STIX to support complex patterns. *Information Technology: New Generations. AISC*, vol. 448, pp. 213–225. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32467-8_20
34. Wagner, T.D., Palomar, E., Mahbub, K., Abdallah, A.E.: Towards an anonymity supported platform for shared cyber threat intelligence. In: Cuppens, N., Cuppens, F., Lanet, J.-L., Legay, A., Garcia-Alfaro, J. (eds.) *CRiSIS 2017. LNCS*, vol. 10694, pp. 175–183. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76687-4_12
35. Zarca, A.M., et al.: Security management architecture for NFV/SDN-aware IoT systems. *IEEE Internet Things J.* **6**, 8005–8020 (2019)
36. Zhu, Z., Dumitras, T.: ChainSmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 458–472. IEEE (2018)

Data Anomaly Detection: Predict and Prevent



Inferring Anomaly Situation from Multiple Data Sources in Cyber Physical Systems

Sara Baldoni¹✉, Giuseppe Celozzi², Alessandro Neri¹, Marco Carli¹,
and Federica Battisti¹

¹ Roma Tre University, Rome, Italy

{sara.baldoni,alessandro.neri,marco.carli,federica.battisti}@uniroma3.it

² Ericsson S.p.A., Rome, Italy
giuseppe.celozzi@ericsson.com

Abstract. Cyber physical systems are becoming ubiquitous devices in many fields thus creating the need for effective security measures. We propose to exploit their intrinsic dependency on the environment in which they are deployed to detect and mitigate anomalies. To do so, sensor measurements, network metrics, and contextual information are fused in a unified security architecture. In this paper, the model of the proposed framework is presented and a first proof of concept involving a telecommunication infrastructure case study is provided.

Keywords: Anomaly detection · Cyber physical system · Critical infrastructure protection

1 Introduction

Cyber Physical Systems (CPSs) can be defined as the result of the integration of computing, communication, and control capabilities for monitoring and managing physical world objects [5]. The “Industry 4.0” paradigm is pushing the spread of CPSs in many fields: smart manufacturing, e-health, smart city, smart vehicles, wearable devices, telecommunication systems, defense systems, etc. As can be easily understood, the security of CPSs is an open and critical issue [5].

A CPS can be described with a three-layer architecture: perception, transmission, and application. The first layer is responsible for data collection in real-time, the second deals with data exchange, and the third layer provides data processing and control functionalities. Even if the security can be addressed for each layer separately, to counteract the attacks in an effective manner, a multi-layer approach can be beneficial. The intrinsic dependency between CPSs and the sensing environment can further expand the attack surface. However, this connection can also be exploited to design effective security measures. The

The current work has in parts been supported by the EU projects RESISTO (Grant No. 786409) on cyber-physical security of telecommunication critical infrastructure.

© The Author(s) 2021

H. Abie et al. (Eds.): CPS4CIP 2020, LNCS 12618, pp. 67–76, 2021.

https://doi.org/10.1007/978-3-030-69781-5_5

idea of context aware security has been introduced in [14], where the context is defined as “the set of environmental states and settings that either determine an application’s behaviour or in which an application event occurs”. Moreover, four context classes are introduced: system, user, physical environment, and time. This concept has been applied to CPSs in [8], where context is considered as a new class of information to be exploited for improving the safety and security of CPSs. The authors suggest that contextual data can be used both for inferring information about the system state, and for preventing wrong detection decisions due to bad-data scenarios. In [7], moreover, a general introduction about CPS security issues is provided together with the presentation of a context aware biometric security framework. The proposed approach fuses real-time mechanisms with contextual information such as the client setting area, lighting, temperature, climate and time. Context is considered also in the security infrastructure for Internet of Things (IoT) systems proposed in [11]. The presented architecture includes some contextual information, such as the amount and rate of collected data, which may be correlated to security indicators. Furthermore, the environmental impact has been exploited in [12], where device fingerprinting for IoT authentication is analyzed. More specifically, the authors propose to exploit the environmental effects on IoT fingerprints to detect emulation attacks. The idea is that an attacker will not be able to imitate the true environmental changes experienced by the legitimate device thus failing in reproducing an environment-based fingerprint. At last, in [10], the concept of context aware intrusion detection systems is realized by including the operating environment information in the collected data. This category involves for instance networking conditions (e.g. start and goal address/port, access frequency, and data traffic), and systematic operation conditions (e.g. the presence of idle CPU or memory occupation conditions). In this paper, we focus on the impact of the physical environment context to design a CPS anomaly detection framework, and we apply the introduced model to a telecommunication infrastructure case study. More specifically, according to [4], here we define an anomalous situation as a malicious or a genuine but unusual behaviour of the system. We propose the CPS Context Aware Security Protection for Enhanced Robustness (CCASPER) model, to identify anomalies in the system behaviour exploiting both the physical aspect of the CPS and the information that can be collected through their networking capabilities. In more details, sensor outputs, network performance indicators, and the surrounding context conditions are fused together for selecting the mitigation strategy. A multi-metric approach which treats each metric independently, in fact, would ignore the possible correlations or cause-effect relationships between them, thus resulting less effective [2].

The rest of the paper is organized as follows. In Sect. 2 the security framework is presented, in Sect. 3 the model is applied to a communication infrastructure scenario, and in Sect. 4 the conclusions are drawn.

2 Proposed Method

The proposed CCASPER model is based on the fusion of the information gathered from two sources: the system (i.e. the deployed CPSs) and the context. Concerning the first source, information may be collected both in the perception layer (which gathers data about a physical process) and in the communication layer (which allows to share information). Nominal value ranges for sensor outputs and network performance indicators may be defined, so that when the measurements deviate from their nominal values, an anomaly is detected and an alarm is triggered. However, using only system indicators could give a partial representation of the real scenario leading to a misclassification of anomalies. The physical process and the communication link, in fact, are considerably influenced by the surrounding context. If sensor outputs and network performance indicators do not match the expected values, then, at least four causes should be investigated: a fault, a cyber-attack, or a physical attack may be occurring, or the surrounding ecosystem may be affecting network performances and/or sensor measurement. Therefore, before triggering an attack alarm, it is worth to collect contextual information to verify if a natural event (a storm, an earthquake, or strong wind) may cause such anomaly. To this aim, the proposed model relies on a context monitoring algorithm localized around the CPS position. In fact, adverse environment conditions may cause a temporary decrease of network performance or, even, damages to the physical system (e.g., antenna collapsing or broken sensor). In the first case, once the environmental emergency is over, the normal operating level of the network has to be quickly restored. On the contrary, a physically damaged system must be repaired. The proposed security model can be described through the state machine architecture shown in Fig. 1.

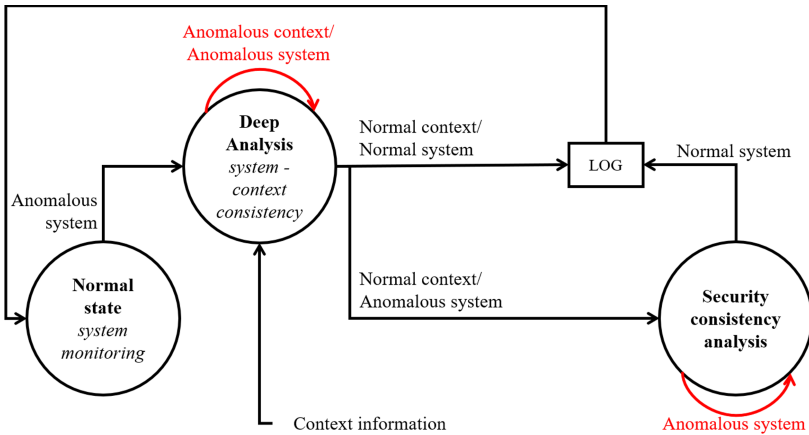


Fig. 1. Security framework: state machine model.

The state machine model is detailed as follows. In the *normal state* the system monitoring is performed through the analysis of application-dependent

quantitative attributes. For example, in a point to point connection, network performance metrics could be delay or throughput values while the sensor measurements could be temperature, humidity, or antenna tilt. If a relevant deviation from the nominal behavior occurs, the model moves to the *deep analysis state*. In this state, a first consistency analysis is performed to verify if the system behaviour can be justified through the current context conditions. To do so, several additional information sources can be analysed: weather news, local news, exceptional events such as flooding or fire, etc. Moreover, due to the distributed nature of CPSs, information gathered from neighbouring devices can be included in the consistency analysis. The context, in fact, should cause a similar deviation from the nominal behaviour for CPSs deployed on the same area. The gathered data are exploited to perform the consistency check between the current context and the quantitative attributes, leading to one of the following outcomes:

- anomalous context/anomalous system: the environmental context may have caused a temporary network performance decrease or a change in the monitored physical process so that the measured values deviate from the nominal range. In this case a joint monitoring is performed until when the context anomalous behaviour is over. Two outcomes are possible:
 - the system returns in the nominal range: the event is saved in a log file, and the system goes back to the *normal state*.
 - the system anomaly persists thus requiring a further analysis. The model moves to the *security consistency analysis state*.
- normal context/anomalous system: a further analysis is needed, the model goes in the *security consistency analysis state*.

If the model goes in the *security consistency analysis state* an additional monitoring is performed in order to identify the possible causes for system anomaly. This state can be further expanded as shown in Fig. 2.

Depending on the use case, different combinations of the monitored parameters may lead to four states which, in turn, represent the anomaly cause:

- **cyber attack** a cyber attack to the sensor may be on going;
- **cyber-physical attack** an attack involving both physical and cyber aspects of the system may be on going.
- **physical attack** a physical damage of the system may have occurred;
- **fault** a system fault (e.g. a component damage) may have occurred.

Once the cause has been identified, the associated recovery procedure is deployed. If such procedure is effective, the event occurrence is saved and the model returns to the *normal state*, otherwise a new security consistency check is performed. Let us note that in case of cyber-physical attacks, a further study is needed to select the most suitable recovery procedure. As first proof of concept, here we present the case in which the choice between cyber and physical recovery plans is performed based on cost. However, depending on the application, other parameters could be considered such as the time needed for the solution deployment. In some scenarios, in fact, restoring the system availability is the most crucial issue.

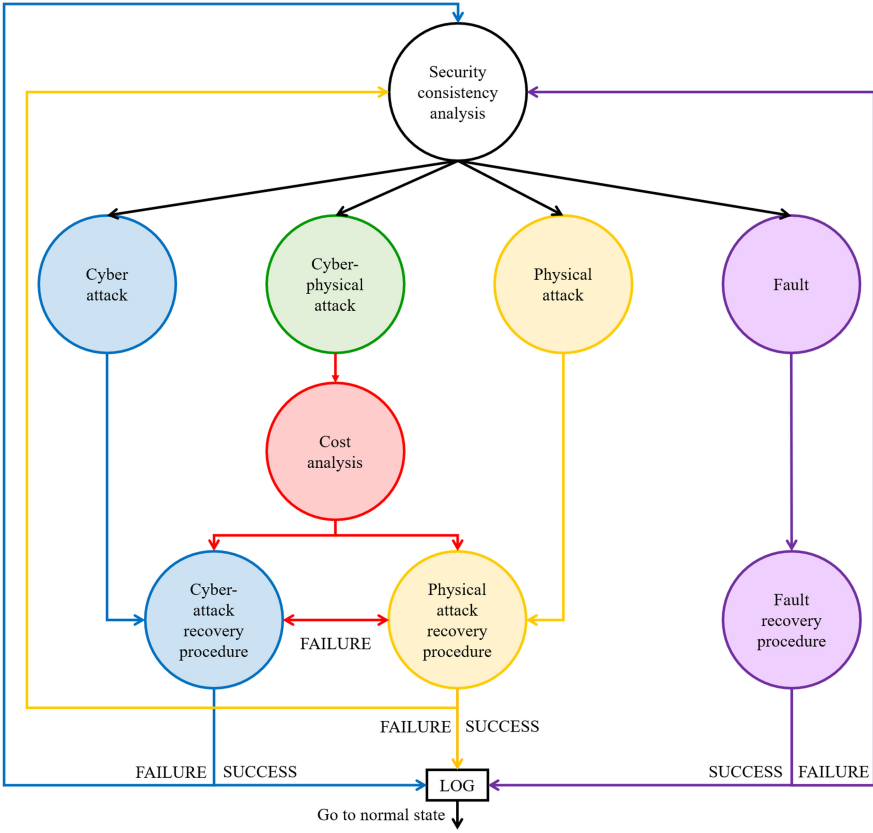


Fig. 2. Security framework: consistency analysis model.

3 Case Study

In this paper we deal with a use case scenario involving a mobile network operator radio infrastructure. The strict 5G requirements lead to the need of relevant improvements in the Radio Access Network (RAN). To this aim, the classic rule-based network functionalities can be replaced by their Artificial Intelligence (AI) based counterparts. However, a key enabler for the application of AI in this context is a deep insight into the nature and role of the different network performance contributors [1]. Moreover, the satellite segment of 5G networks is one of the main topics in the 5G development in 2020–2025 [13] and, as highlighted in [9], for such applications the antenna pointing and the mobile tracking are crucial. More specifically, for pointing purposes, the AI-based method proposed in [9] exploits data acquired by the inclinometer and an electronic compass. In addition, antenna tilting is considered as a key enabler for RAN optimization also in [1]. Let us note that tilt monitoring is of utmost importance both for mechanical and electronic tilting systems. In the former case it is needed to

correctly set the beam pointing, whereas in the latter the physical orientation of the antenna has to be set and kept with sufficient accuracy. For this reason, we included a tilt sensor installed on the base station antennas in our case study.

Concerning the network performance metrics, we referred to the security studies performed on the existing LTE infrastructure. In [6], for instance, the authors analyzed three metrics: throughput, end to end delay, and packet delivery rate. Their simulations show a significant impact on these parameters due to malicious user equipment devices, malicious base stations, malicious connections, and malicious femtocells. To show the operating principle of the proposed framework, in this work we considered only the throughput, but the analysis can be extended to the other network parameters as well. In this case, the general framework presented in Fig. 1 can be modified as shown in Fig. 3.

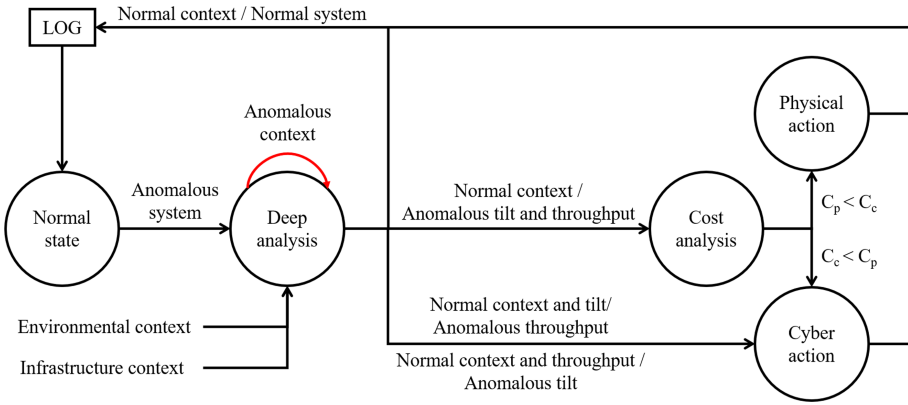


Fig. 3. Security framework: case study model. In the figure, C_c and C_p represent the cyber and physical action cost, respectively.

As in the general model, once an anomalous parameter is detected, the system moves to the *deep analysis state*. More in details, we included in the context monitoring the analysis of: weather, terrorism, civil protection department alerts, earthquakes, floods, severe storm, and fire. Moreover, in this scenario, the measurements collected by nearby base stations can be considered as an additional information source. As a consequence, the context information includes both the surrounding environment and the neighbouring infrastructure. The deep analysis correlates these data with the current quantitative attributes. For instance, a heavy rain may cause a severe attenuation phenomenon which, in turn, causes a throughput reduction without any impact on the antenna tilting. A strong wind, on the contrary, may cause an antenna movement which will produce a pointing inaccuracy. In this case, the first phenomenon will be detected by the tilt sensor, whereas the latter will have an impact on the measured throughput. Under these circumstances, the system stays in the *deep analysis state* until when the environmental emergency is over.

If the contextual cause can be excluded, or if the system parameters are still anomalous when the context emergency is over, a consistency check of the remaining inputs is performed. The analysis of the throughput indicator and the antenna tilt sensor, in fact, can lead to different scenarios. A normal tilt value in conjunction with an anomalous throughput can suggest a channel attack. The opposite situation can be due to a cyber-attack to the tilt sensor. If both indicators fail to satisfy their requirements, a physical attack to the antenna, or the combination of a channel and sensor attacks, should be considered. Moreover, the anomalous context may have caused a network damage which has to be physically repaired. If both cyber and physical recovery hypothesis need to be checked, the mitigation scheduling is chosen according to the cost of the required operations (C_c and C_p respectively). As a consequence, the cheaper recovery procedure is deployed first and, if it is not effective, the other one is performed. Let us note that when a “Physical action” is required, a technical team will be sent to the antenna location. This solution will thus be required in case of physical antenna damage.

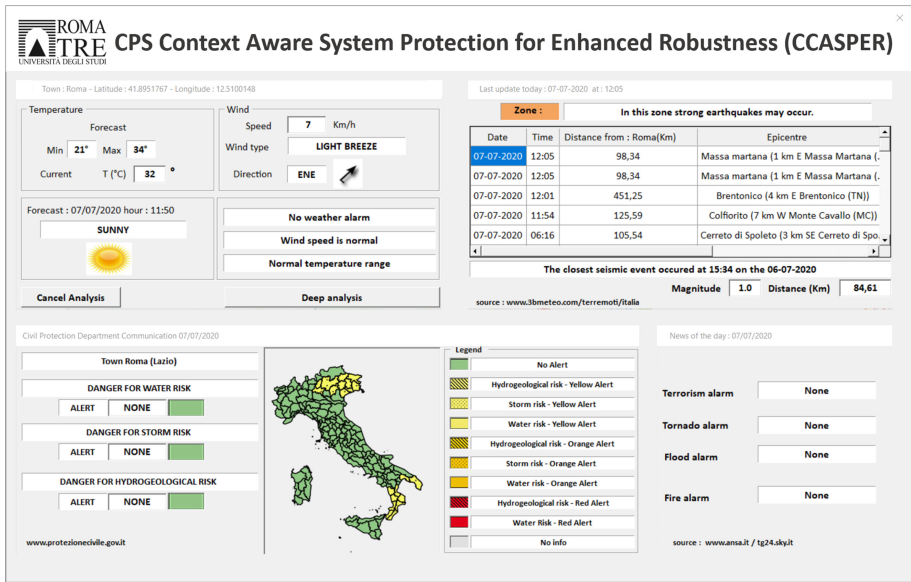


Fig. 4. Context data collection: the CCASPER GUI

As first proof of concept of the proposed security framework we implemented the CCASPER model and a user interface. In order to gather up-to-date environmental context information, the antenna position has to be retrieved. To this aim, we used the antenna map provided in [3]. Moreover, in the implemented system, antennas can be deleted or added directly from the Graphical User Interface (GUI). From the tool GUI, it is possible to select a specific antenna and run

the simulation. In the performed tests, as a proof of concept, tilt and throughput data have been entered by the user. The implementation of the complete framework which collects the tilt information from the sensor and uses the measured throughput will be the object of future research. Once the tilt and throughput data have been entered, context information about the location where the antenna is placed is collected and shown as presented in Fig. 4. The button “Deep analysis” allows to move to the corresponding state in which, as first step, the context analysis is performed. In case of normal environmental conditions, according to the tilt and throughput values, a cost analysis or a cyber action can be suggested as shown in Fig. 3.

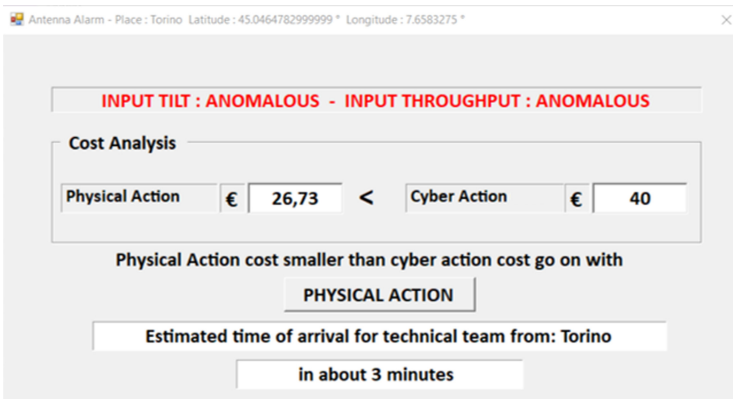


Fig. 5. Cost analysis output.

Concerning the cost analysis, the physical action cost may be computed depending on the distance of the antenna from the actual position of the technical team. More specifically, we assumed to have an office for each town and we considered a total cost made of a fixed component plus a variable cost linked to the distance from the antenna. However, it is possible to configure the costs directly from the GUI main page. An example of the cost analysis output is provided in Fig. 5.

The proposed model allows to process the information coming from several data sources to figure out the reason behind an anomalous behaviour of the system, thus suggesting a suitable mitigation strategy. Although limited to the analysis of two quantitative attributes, the presented approach can be easily extended for including other performance indicators (e.g. delay) and/or other sensor measurements (e.g. temperature). According to the analyzed case study, we argue that the fusion of environmental, technical, and financial information sources is a key enabler to provide prompt reactions to communication infrastructure anomalies.

4 Conclusions

In this contribution, a model for inferring anomaly situations is proposed. Technical and environmental inputs are considered for defining the best mitigation technique in terms of effectiveness and financial feasibility. We implemented the proposed security framework through a state machine model which provides the general architecture of the proposed approach. As first proof of concept we analyzed a telecommunication infrastructure case study. To do so, we realized a tool for collecting and fusing the information coming from the different sources, and exploited a user friendly graphical interface for entering input data.

The proposed framework can be applied to every scenario involving CPSs, after the definition of application-dependent quantitative attributes. We believe that the fusion of several information sources is a crucial facilitator for inferring anomalous situations in CPS-based critical infrastructures.

Acknowledgement. The authors would like to acknowledge the support provided by Vanessa Gaglione that carried out part of this work in her M.Sc. thesis project.

References

1. Enhancing RAN performance with AI. <https://www.ericsson.com/493ce3/assets/local/reports-papers/ericsson-technology-review/docs/2020/enhancing-ran-performance-with-ai.pdf>. Accessed 08 July 2020
2. How to build robust anomaly detectors with machine learning. <https://www.ericsson.com/en/blog/2020/4/anomaly-detection-with-machine-learning>. Accessed 08 July 2020
3. Italian antenna map. http://www.datiopen.it/it/opendata/Mappa_delle_antenne_in_Italia. Accessed 17 June 2020
4. Amaral, A.A., de Souza Mendes, L., Zarpelão, B.B., Proença Junior, M.L.: Deep IP flow inspection to detect beyond network anomalies. *Comput. Commun.* **98**, 80–96 (2017). <https://doi.org/10.1016/j.comcom.2016.12.007>, <http://www.sciencedirect.com/science/article/pii/S0140366416306612>
5. Ashibani, Y., Mahmoud, Q.H.: Cyber physical systems security: analysis, challenges and solutions. *Comput. Secur.* **68**, 81–97 (2017). <https://doi.org/10.1016/j.cose.2017.04.005>, <http://www.sciencedirect.com/science/article/pii/S0167404817300809>
6. Bhattarai, S., Rook, S., Ge, L., Wei, S., Yu, W., Fu, X.: On Simulation studies of cyber attacks against LTE networks. In: 2014 23rd International Conference on Computer Communication and Networks (ICCCN), pp. 1–8 (2014)
7. Dsouza, J., Elezabeth, L., Mishra, V.P., Jain, R.: Security in cyber-physical systems. In: 2019 Amity International Conference on Artificial Intelligence (AICAI), pp. 840–844 (2019)
8. Ivanov, R., Weimer, J., Lee, I.: Towards context-aware cyber-physical systems. In: 2018 IEEE Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS), pp. 10–11 (2018)
9. Liu, Q., Yang, J., Zhuang, C., Barnawi, A., Alzahrani, B.A.: Artificial intelligence based mobile tracking and antenna pointing in satellite-terrestrial network. *IEEE Access* **7**, 177497–177503 (2019)

10. Park, S.-T., Li, G., Hong, J.-C.: A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning. *J. Ambient Intell. Human. Comput.* **11**(4), 1405–1412 (2018). <https://doi.org/10.1007/s12652-018-0998-6>
11. Roukounaki, A., Efremidis, S., Soldatos, J., Neises, J., Walloschke, T., Kefalakis, N.: Scalable and configurable end-to-end collection and analysis of iot security data : towards end-to-end security in IoT systems. In: 2019 Global IoT Summit (GIoTS), pp. 1–6 (2019)
12. Sharaf Dabbagh, Y., Saad, W.: Authentication of wireless devices in the internet of things: learning and environmental effects. *IEEE Internet Things J.* **6**(4), 6692–6705 (2019)
13. Tikhvinskiy, V., Koval, V.: Prospects of 5G Satellite Networks Development. In: Moving Broadband Mobile Communications Forward-Intelligent Technologies for 5G and Beyond. IntechOpen (2020). <https://doi.org/10.5772/intechopen.90943>, <https://www.intechopen.com/online-first/prospects-of-5g-satellite-networks-development>
14. Wang, E.K., Ye, Y., Xu, X., Yiu, S.M., Hui, L.C.K., Chow, K.P.: Security issues and challenges for cyber physical system. In: 2010 IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing, pp. 733–738 (2010)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Fusing RGB and Thermal Imagery with Channel State Information for Abnormal Activity Detection Using Multimodal Bidirectional LSTM

Nikolaos Bakalos¹(✉), Athanasios Voulodimos², Nikolaos Doulamis¹,
Anastasios Doulamis¹, Kassiani Papatotiriou³, and Matthaïos Bimpas¹

¹ School of Rural and Surveying Engineering, National Technical University of Athens,
15773 Athens, Greece

bakalosnik@mail.ntua.gr

² Department of Informatics and Computer Engineering, University of West Attica,
12243 Athens, Greece

³ Columbia University, New York, NY 10027, USA

Abstract. In this paper, we present a multimodal deep model for detection of abnormal activity, based on bidirectional Long Short-Term Memory neural networks (LSTM). The proposed model exploits three different input modalities: RGB imagery, thermographic imagery and Channel State Information from Wi-Fi signal reflectance to estimate human intrusion and suspicious activity. The fused multimodal information is used as input in a Bidirectional LSTM, which has the benefit of being able to capture temporal interdependencies in both past and future time instances, a significant aspect in the discussed unusual activity detection scenario. We also present a Bayesian optimization framework that fine-tunes the Bidirectional LSTM parameters in an optimal manner. The proposed framework is evaluated on real-world data from a critical water infrastructure protection and monitoring scenario and the results indicate a superior performance compared to other unimodal and multimodal approaches and classification models.

Keywords: Abnormal activity detection · Human intrusion · Multimodal data fusion · Bidirectional LSTM · Critical infrastructure monitoring

1 Introduction

Abnormal activity detection is a research problem that attracts significant interest in the image and video analysis research community (e.g. [9, 10]). Many different techniques have been proposed in the field of computer vision and video analysis, including methods based on trajectory analysis [12], pixel-level processing [11], combined trajectory and low-level analysis [1], background modelling [14], object detection [13] and tracking [15], activity recognition [16], and crowd behavior analysis [17]. Despite the efficacy of such techniques, their dependence on strictly visual information makes them susceptible

to occlusions, difficult fields of view and poor illumination circumstances. This limitation has motivated the exploration of vision techniques beyond the visible spectrum. Thermographic data can provide a useful alternative stream of information. Thermal camera sensors are not sensitive to illumination changes [4]; on the other hand, thermal information does not entail texture or color information. Since both RGB and thermal sensing are actually based on visual cues, an interesting idea is to supplement them by additional data that are not limited by the restrictions of visual information (such as occlusions).

Recent studies have indicated that wireless signal reflection can be effectively leveraged to sense human presence. Different kinds of techniques have been described in the literature, including device-free Software Defined Radio (SDR) methods, which process the Received Signal Strength of a transmitted signal. However, the accuracy of such techniques is often not sufficiently high [18]. In contrast, it has been shown that techniques based on commercial off the shelf (COTS) equipment [5] can yield good performance rates in human presence detection, by making use of Channel State Information (CSI) [7].

Moving on from the input modalities to the machine learning models used for abnormal activity detection, it is clear that deep learning techniques, and especially Convolutional Neural Networks (CNN), have been shown to outperform traditional classifiers [1, 6, 16], which is explained by their high representational capabilities. However, one limitation of CNNs is that they cannot inherently capture temporal interdependencies in a bidirectional manner, i.e. from both past and future time instances, which is an important aspect in time series modeling problems.

In this work, we propose a model based on a Bayesian optimized multimodal bidirectional LSTM neural network for abnormal activity detection. Our model harnesses the power of LSTM networks to capture long and short term dependencies, while the backward and forward pass of the bidirectional version of LSTM ensure the consideration of both past and future time instances. Our proposal also includes a Bayesian optimization framework that optimally tunes the parameters of the utilized bidirectional LSTM. Finally, the combination of heterogeneous input modalities, such as RGB and thermal imagery with Channel State Information (CSI) from wireless signal reflection leads to a significantly improved detection performance compared to cases that are solely based on a single information modality.

2 Fusion of RGB and Thermal Imagery with Channel State Information

2.1 RGB Imagery

Contrary to traditional abnormal activity detection systems which are usually based on RGB video sequence input, in the work at hand an additional modality is considered, that of thermographic imagery. Visual streams from RGB cameras are initially processed using the object detection module YOLO (You only look once) [13]. YOLO locates spatial bounding areas on the frame and allocates each area a probability for an object class. A Convolutional Neural Network is used for object detection, comprising 24

convolutional layers and 2 fully connected layers. Each image frame is described as a class image CL_{RGB} , having the same size as the initial RGB image, where the (x,y) pixel of the RGB image $I(x,y)$ is denoted as $o_{k,RGB}(x,y)$, in the class in the following way:

$$CL_{RGB}(x,y) = o_{k,RGB}(x,y) \quad (1)$$

where k denotes the object with identity k in the object detection module employed.

2.2 Thermal Imagery

Data acquired by thermographic sensors undergo background subtraction [14]. A class label image CLT is extracted, having the same size as the input thermal frame T, where the (x,y) pixel of T is denoted in the class label image as:

$$CL_T(x,y) = o_{b,T}(x,y), b = \{Background, Foreground\} \quad (2)$$

In order to facilitate the subsequent processing steps, the RGB and thermal image frames are resized so as to become of identical sizes, $N \times M$. In other words, $x_{RGB}(n) \in R^{N \times M}$ stands for an image, whereby each pixel indicates the object ID that pixel belongs to. In a similar manner, tensor $x_{thermal}(n) \in R^{N \times M}$ denotes the class label image of the thermographic modality.

2.3 Channel State Information

Channel State Information (CSI) can be leveraged for human movement detection using WiFi devices, based on propagation modeling of a signal from the transmitter to the receiver, supporting many subcarriers due to the Orthogonal Frequency Division Multiplexing (OFDM) principle. CSI includes physical attributes of the wireless channel, such as scattering, power decay per distance, fading, shadowing and effects of interference [7], which are measured by the amplitude/phase over all K available subcarriers:

$$H(n) = [H(n, f_1), H(n, f_2), \dots, H(n, f_k)]^T \quad (3)$$

where $H(n, f_i)$ refers to the amplitude and the phase of the i -th subcarrier with central frequency f_i . Therefore, we have that: $H(n, f_i) = |H(n, f_i)|e^{j\angle H(n, f_i)}$.

Usually, $H(n)$ input data contain noise and are distorted by outliers. For this reason, CSI signals $H(n)$ need to undergo a pre-processing stage. First, outliers are removed using a Hampel identifier [8] or density-based clustering algorithms such as DBSCAN [23]. In the sequel, noise is removed with wavelet denoising, followed by normalization, correlation of subcarriers and, finally, eigenvector processing of the signals. After pre-processing, CSI data are used as input to a linear SVM for human intrusion detection. The SVM's output classification IDs, say $C_{CSI}(n)$, will be used as input to our proposed multimodal bidirectional LSTM framework. The CSI related input $x_{CSI}(n)$ is given by:

$$x_{CSI}(n) = [H(n)C_{CSI}(n)]^T \quad (4)$$

For spatial coherency with the visual input data, tensor $x_{CSI}(n)$ is expanded over the $R^{N \times M}$ grid, forming an additional input channel.

2.4 Fusion of RGB, Thermal and CSI Modalities

Approaches based on solely one of the above types of information are unavoidably plagued by the limitations of each information modality (e.g. occlusions, noise, etc.). We hereby propose the fusion of the above described information channels to create a multimodal input tensor $x(n)$:

$$x(n) = [x_{RGB}(n), x_{thermal}(n), x_{CSI}(n)]^T \tag{5}$$

where $x_{RGB}(n)$ is the data tensor pertaining to RGB visual signals, $x_{thermal}(n)$ the respective data tensor of the thermal component, and $x_{CSI}(n)$ the data tensor pertaining to the WiFi reflection signal.

3 Bayesian Optimized Multimodal Bidirectional LSTM

3.1 Bidirectional LSTM

LSTMs is a type of Recurrent Neural Network (RNN) which was designed to address the problem of exploding and vanishing gradient that can arise when training traditional RNNs. LSTM networks are a good fit to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series [25–27]. In LSTMs, each node in the hidden layer is replaced by a memory cell, instead of a single neuron [25]. The structure of a memory cell is illustrated in Fig. 1.

The LSTM memory cell is composed of the following: the forget gate, the input node, the input gate, and the output gate. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The activation function of the LSTM gates is often the logistic sigmoid function.

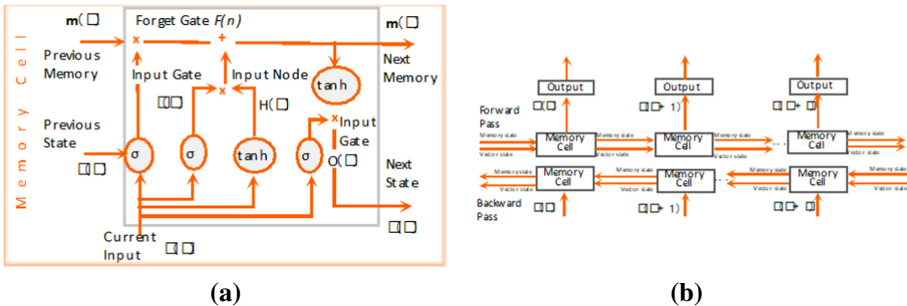


Fig. 1. (a) The memory cell of a LSTM network. (b) Bidirectional LSTM unfolded in time

The goal of the forget gate is to decide what information should be discarded out of the memory cell [24]. The output, denoted as $f(n)$ ranges between 0 and 1, according

to the sigmoid activation function. The forget gate learns whether a previous or future vector state is necessary for the estimation of the current value state. The input node performs the same operation with that of a hidden neuron of a typical recurrent regression model. We denote the output of this node as $I_n(n)$. The goal of this node is to estimate the way in which each latent state variable contributes to the final model.

As far as the input gate is concerned, its role is to regulate whether the respective hidden state is sufficiently important. The output of this gate is denoted as $I_g(n)$. It has the sigmoid function, therefore its response ranges between 0 and 1. This gate addresses problems related to the vanishing of the gradient slope of a $\tan H(\cdot)$ operator. Finally, the output gate regulates whether the response of the current memory cell is sufficiently significant to contribute to the next cell. Therefore, this gate actually models the long range dependency together with the forget gate. The output of this gate is denoted as $O(n)$.

One of the disadvantages of the memory cell of Fig. 1, is that it considers only past state information. On the contrary, bi-directional forms of LSTM can process data in both directions, and include, therefore, apart from the forward pass an additional backward operation. The structure of a bi-directional LSTM, unfolded in time is presented in Fig. 1(b). Detection of abnormalities in video and CSI time series is an application which can inherently benefit from this additional backward operation of the bidirectional LSTM, which is the base model adopted in this work.

3.2 Bayesian Optimization

We hereby present a Bayesian optimization method for the selection of the bidirectional LSTM model parameters. In lieu of employing manual tuning of model parameters, we hereby present and use a probabilistic Bayesian approach through which model parameters are optimally tuned.

As in all models, let us denote by π_i the set of configurable parameters, e.g. in our case the number of memory cells, the learning rates, etc. Supposing a set Q of different configurations, i.e., $D_{1:Q} = \{\pi_1 \dots \pi_Q\}$, we can then evaluate the error $E(x, d, \pi)$ yielded when (i) the model receives input data x , (ii) its output is compared against the target outputs d and (iii) we consider a specific model configuration π . Let E_{min} be the minimum Mean Square Error across all Q configurations. The following can then be an improvement function:

$$I(x, d, \pi) = \max\{0, E_{min} - E(x, d, \pi)\} \quad (6)$$

In the sequel, the expectations of Eq. (6) can be computed in a probabilistic context. The probability distribution of the error function for a given set of configurations, $P(E|D_{1:Q})$, is written in a Bayesian context as:

$$P(E|D_{1:Q}) \propto P(D_{1:Q}|E)P(E) \quad (7)$$

Usually $P(E)$ follows a Gaussian distribution and $P(D_{1:Q}|E)$ is then expressed as a Gaussian process of mean $\mu(\pi)$ and standard deviation Σ [28]:

$$\Sigma = \begin{bmatrix} k(\pi_1, \pi_1) & \dots & k(\pi_1, \pi_Q) \\ \vdots & \ddots & \vdots \\ k(\pi_Q, \pi_1) & \dots & k(\pi_Q, \pi_Q) \end{bmatrix} \quad (8)$$

where $k(\bullet)$ is a kernel function. The target of our optimization is to find out a new configuration $\pi^* \equiv \pi_{Q+1}$, which will further reduce the MSE or equivalently increase the improvement $I(x, d, \pi^*)$. Then, for the new augmented set $D_{1:Q+1}$, that includes $\pi^* \equiv \pi_{Q+1}$, $P(D_{1:Q+1}|E)$ will again be a Gaussian process of standard deviation

$$\begin{bmatrix} \Sigma & b \\ b^T & k(\pi_{Q+1}, \pi_{Q+1}) \end{bmatrix} \quad (9)$$

Where $b = [k(\pi_{Q+1}, \pi_1) \dots k(\pi_{Q+1}, \pi_Q)]$. Then, according to [28], it can be proven that the $P(E_{Q+1}|D_{1:Q}, \pi_{Q+1})$ is also a Gaussian with mean value and standard deviation related with previous variables. Therefore, the new configuration π^* is estimated, which is actually the integral of $I(\bullet)$ and $P(E_{Q+1}|D_{1:Q}, \pi_{Q+1})$, that is the probability that $I(\bullet)$ follows.

4 Experimental Evaluation

4.1 Experimental Setup

To scrutinize the effectiveness of the proposed model, we have used a dataset that has been created in the context of the European Horizon 2020 STOP-IT Project (<https://stop-it-project.eu/>). STOP-IT aims at tackling the protection of critical water infrastructure using novel methods. The dataset includes RGB and thermal video sequences as well as Channel State Information. The RGB data were captured using an OB-500Ae camera with 1280×720 pixel resolution at 30 fps. The thermal data were obtained by means of a Workswell InfraRed Camera 640 (WIC) with a 640×512 pixel resolution at 30 fps. WiFi data were acquired using a transmitter-receiver couple that comprised a WiFi router (TP-Link N300 TL-WR841N) and an Intel 5300 NIC receiver, with a 0.1 Hz capturing frequency. Data annotation was performed on the basis of pre-determined scenarios by end users that prescribed whether the captured activity over all data modalities should be considered as irregular/abnormal.

The entirety of data across all modalities were normalized so as to be in the same range (0–1). The computer used for all training and testing was an Intel® Core™ i7-6700 CPU@ 4000 GHz CPU with 16GB RAM and an NVIDIA GeForce GTX 1070 with 8GB DDR5 memory. CUDA 9.2 Toolkit was also used for deep learning classifiers.

4.2 Experimental Results

The first round of experiments focuses on the impact of using fused multimodal data as input, instead of solely considering a single modality. We have initially experimented

with the following popular machine learning models: (i) a linear kernel SVM, (ii) a non-linear Radial Basis Function SVM (RBF-SVM), two different architectures of a traditional feedforward neural network: (iii) with 1 hidden layer of 10 neurons/layer and (iv) 2 hidden layers of 10 neurons/layer respectively, (v) a CNN and (vi) a plain LSTM (without bidirectionality or optimization). Fig. 2 depicts the accuracy rates attained by the above classifiers in cases with (a) only RGB and thermal input, (b) CSI (WiFi) and (c) multimodal input. From the results, it is evident that the proposed data fusion scheme of significantly increases the achieved performance detection performance regardless of classification scheme.

In the second round of experiments, we conduct experiments to validate the effectiveness of the proposed multimodal Bayesian optimized bidirectional LSTM. Focusing on the multimodal case, we compare the performance of the proposed model with the six models mentioned above (SVM-linear, SVM-RBF, FNNs, CNN, LSTM). The results of the experiments in terms of precision, recall, F1-score and accuracy are depicted in Table 1. We observe that all deep learning models (CNN, LSTM) clearly outperform shallow classifiers, which is explained by the greater representational and understanding power of the deep models in complex scenarios such as the discussed abnormal activity detection application. Moreover, the proposed approach based on optimized bidirectional LSTM attains higher performance rates compared to the other examined deep learning models, revealing the contribution of both the bidirectionality and the proposed framework for Bayesian optimization of the network parameters.

Table 1. Performance metrics on multimodal experiments

Method	Precision	Recall	Accuracy	F1 score
SVM-Linear	68.51%	61.71%	77.36%	64.93%
SVM-RBF	66.99%	60.06%	76.11%	63.34%
FNN1	69.95%	63.30%	78.52%	66.46%
FNN2	70.13%	63.50%	78.66%	66.65%
CNN	80.62%	76.09%	86.56%	78.29%
LSTM	81.14%	76.12%	87.11%	78.55%
Proposed Optimized Bidirectional LSTM	90.01%	87.42%	88.70%	88.77%

Finally, we have experimented with providing as input to the classifiers a “window” of past frames of different sizes, in other words feeding the model with “memory”. We have explored three cases for window length: no window, brief window (50 frames) and longer window (100 frames). The results for the multimodal case are depicted in Fig. 3. We can see that the presence of a time window in the input increases the performance in the examined cases of CNN, LSTM and the proposed optimized bidirectional LSTM, but the improvement ratio decreases as the window length increases. Furthermore, the improvement attained by the window is less significant in the proposed model compared to CNN and plain LSTM, where there is more room for improvement. In any case, though,

the performance attained by the proposed model steadily outperforms the remaining examined approaches by a considerable difference.

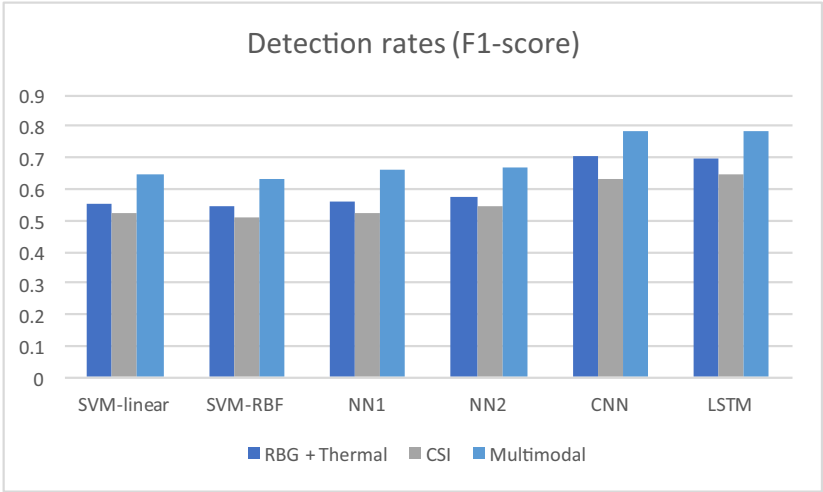


Fig. 2. Attained F1-score of shallow and deep learning models for: (i) visual (RGB + thermal), (ii) WiFi-CSI, and (iii) multimodal input.

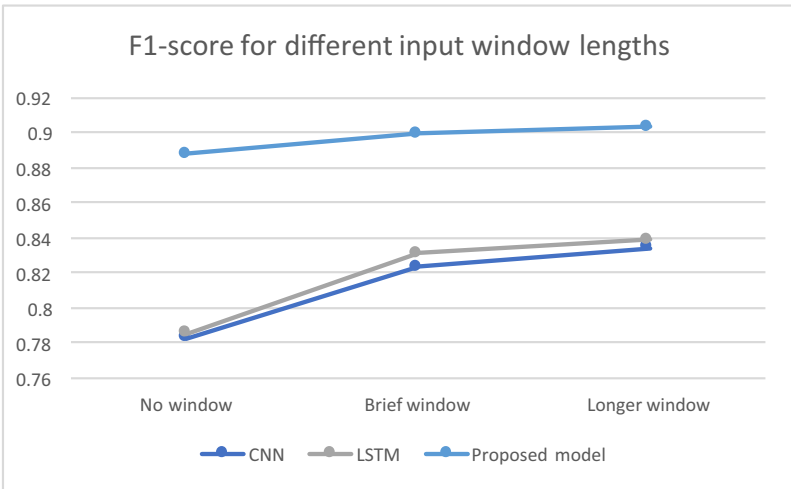


Fig. 3. Attained F1-score for different input window lengths (number of frames) in the multimodal case for: (i) CNN, (ii) plain LSTM, and (iii) the proposed optimized bidirectional LSTM.

5 Conclusion

In this paper, we proposed a multimodal bidirectional Long Short-Term Memory neural network (LSTM) model for detection of abnormal activity in critical infrastructures.

Three input modalities are considered: RGB, thermal and Channel State Information, the fusion of which is proved to provide significant added value in the unusual activity detection scenario. The multimodal input is fed into a bidirectional LSTM, which allows for an effective capturing of both forward and backward temporal dependencies. Moreover, a Bayesian optimization method is used to optimally select the parameters of the employed model. The presented methods have been experimentally evaluated with a real-world critical water infrastructure monitoring and protection dataset, and have been shown to achieve very promising detection rates.

Funding. The research leading to these results has received funding from the EU H2020 research and innovation programme under grant agreement No. 740610, STOP-IT project.

References

1. Coşar, S., Donatiello, G., Bogorny, V., Garate, C., Alvares, L.O., Brémond, F.: Toward abnormal trajectory and event detection in video surveillance. *IEEE Trans. Circuits Syst. Video Technol.* (2017).
2. Kosmopoulos, D.I., Doulamis, N.D., Voulodimos, A.S.: Bayesian filter based behavior recognition in workflows allowing for user feedback. *Comput. Vis. Image Underst.* **116**(3), 422–434 (2002)
3. Sze, V., Chen, Y.H., Emer, J., Suleiman, A., Zhang, Z.: Hardware for machine learning: challenges and opportunities. In: *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–8 (2017)
4. Makantasis, K., Nikitakis, A., Doulamis, A., Doulamis, N., Papaefstathiou, Y.: Data-driven background subtraction algorithm for in-camera acceleration in thermal imagery. *IEEE Trans. Circuits Syst. Video Technol.* (2017)
5. Halperin, D., Hu, W., Sheth, A., Wetherall, D.: Tool release: gathering 802.11n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.* **41**(1), 53 (2011)
6. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
7. Zhu, H., Xiao, F., Sun, L., Wang, R., Yang, P.: R-TTWD: robust device-free through-the-wall detection of moving human with WiFi. *IEEE J. Selected Areas Commun.* **35**(5) (2017).
8. Davies, L., Gather, U.: The identification of multiple outliers. *J. Amer. Statist. Assoc.* **88**(423), 782–792 (1993)
9. Popoola, O., Wang, K.: Video-based abnormal human behavior recognition -a review. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **42**(6), 865–878 (2012)
10. Li, T., Chang, H., Wang, M., Ni, B., Hong, R., Yan, S.: Crowded scene analysis: A survey. *IEEE Trans. Circuits Syst. Video Technol.* **25**(3), 367–386 (2015)
11. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1975–1981, June 2010
12. Ouivirach, K., Gharti, S., Dailey, M.N.: Incremental behavior modeling and suspicious activity detection. *Pattern Recogn.* **46**(3), 671–680 (2013). <https://www.sciencedirect.com/science/article/pii/S0031320312004426>
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *2016 IEEE CVPR, Las Vegas, NV*, pp. 779–788 (2016)
14. Herrero, S., Bescs, J.: Background subtraction techniques: Systematic evaluation and comparative analysis. In: *11th International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS 2009. Springer, Heidelberg* (2009). https://doi.org/10.1007/978-3-642-04697-1_4

15. Yeo, D.S.: Superpixel-based tracking-by-segmentation using markov chains. In: IEEE Conference in Computer Vision and Pattern Recognition (CVPR) (2017).
16. Kosmopoulos, D., Voulodimos, A., Doulamis, A.: A system for multicamera task recognition and summarization for structured environments. *IEEE Trans. Industr. Inf.* **9**(1), 161–171 (2013)
17. Mousavi, H.M.: Analyzing tracklets for the detection of abnormal crowd behavior. In: IEEE Winter Conference on In Applications of Computer Vision (WACV) (2015)
18. Wu, K., Xiao, J., Yi, Y., Gao, M., Ni, L.M.: FILA: fine-grained indoor localization. In: Proc. IEEE INFOCOM, pp. 2210–2218, March 2012
19. Jiang, D., Zhuang, D., Huang, Y., Fu, J.: “Survey of multispectral image fusion techniques in remote sensing applications”, *Image Fusion and its applications*, Y. Zheng, INTECH Open Access Publisher **1**, 1–22 (2011)
20. Pal, A.R., Singha, A.: A comparative analysis of visual and thermal face image fusion based on different wavelet family. In: 2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC), Shillong, pp. 213–218 (2017)
21. Connor, J., Martin, D., Altas, L.: Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Networks* **5**, 240–254 (1994)
22. Doulamis, A.D., Doulamis, N.D., Kollias, S.D.: An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources. *IEEE Trans. Neural Networks* **14**(1), 150–166 (2003)
23. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD* **96**(34), 226–231 (1996)
24. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. In: 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), vol. 2, pp. 850–855 (1999)
25. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
26. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: ICML, pp. 1310–1318 (2013)
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS, pp. 3104–3112 (2014).
28. Bardenet, R., Balázs, K.: Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In: 27th International Conference on Machine Learning (ICML 2010), Omnipress (2010)


Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Cloud-Based Anomaly Detection for IoT Big Data

Omri Soceanu¹(✉), Allon Adir¹(✉), Ehud Aharoni¹(✉), Lev Greenberg¹(✉),
and Habtamu Abie²(✉) 

¹ IBM Research - Haifa, Haifa University Campus,
3490002 Mount Carmel, Haifa, Israel

{Omri.Soceanu,adir,aehud,levg}@il.ibm.com

² Norwegian Computing Center, adir, P.O. Box 114, 0314 Blindern, Oslo, Norway
habtamu.abie@nr.no

Abstract. Security of IoT systems is a growing concern with rising risks and damages due to successful attacks. Breaches are inevitable, attacks have become more sophisticated, and securing critical infrastructure has become a greater challenge. Anomaly detection is an established approach for detecting security attacks, without relying on predefined rules or signatures of potential attacks. However, existing outlier detection techniques require adaptation if they are to be applied in a Big Data cloud context. We describe a novel outlier detection solution, which is currently being used by hundreds of customers with highly variable data scales. We describe our work in adapting this technology to handle IoT on a Big Data cloud setting. Specifically, we focus on efficient outlier analysis and managing large numbers of alerts using automatically controlled alert budgets.

Keywords: Cyber-security · Anomaly detection · Outliers · IoT big data · Cloud · Critical infrastructure

1 Introduction

IoT security is of the utmost importance. The impact of a security breach can go far beyond the attacked device. Attacks can range from a privacy breach and the exposure of sensitive and personal information, to a life threatening attack on an automotive system, or a distributed bot-nets attack with major financial ramifications. A breach can also lead to diminished customer trust or even to customer turnover.

Attacks have also become wider and more sophisticated. A survey of IoT-enabled cyber attacks [15] assessed the increasing attack paths to critical infrastructures and services for all application domains. DDoS attacks are on the rise, getting more complex and increasing in power with the average attack in 2020 using more than 1 Gbps of data with most attacks now lasting 30 min to an hour [8].

Breaches are inevitable and securing critical infrastructure has become a greater challenge. Moreover, many attacks cannot be stopped by standard policy-based or rule-based security systems. In cases of accounts stolen from privileged users or internal attacks, firewalls, access control levels, and rule-based management are not sufficient. There is a real need for a different approach that detects malicious activities beyond the abilities of rule-based systems.

Anomaly detection is one such established approach, and it has already been used in a wide range of security settings. This approach unites a rich family of techniques [10] and doesn't rely on predefined rules or signatures of potential attacks. Instead it learns the typical behavior of the monitored system and triggers an alert when abnormal behavior is detected.

1.1 Challenges and Our Contribution

Compared to rule-based and signature-based approaches, anomaly detection is a more complex process that includes advanced analytics for training and analysis. This in turn imposes higher demands on disk space, memory, and CPU resources. Even in organizations that recognize the importance of data security, typically only a small portion of the overall computational resources can be allocated for security analytics.

The growing data volumes of industrial use cases make the performance and scalability requirements for this kind of analytics increasingly difficult to meet. This has become a critical factor in the acceptance of anomaly detection solutions. The need to adapt anomaly detection techniques to large scalable systems is especially vital for IoT platforms. Here, large numbers of highly diverse devices, from different domains and behavioral aspects, need to be analyzed to detect anomalies. IoT platforms often include many devices with high communication frequencies. The size of the IoT platform, and thus the amount of data to be analyzed, can also grow or change over time. This raises the requirements of the anomaly detection platform to automatically scale, while preserving performance and resilience, in the face of a dynamically growing IoT system.

Another challenge in anomaly detection is the number of false positives, i.e., cases when an alert is triggered, but no attack is actually taking place. Even a false positive rate of one per mille (0.1%) would make an anomaly detection system impractical in a large scale setup. It would produce too many alerts to be handled by the limited resources of security operators. Such a high rate of alerts would cause the security operators to become less sensitive and might result in their missing an alert triggered for a real attack.

In this work, we present a novel scalable platform for anomaly detection. Our main contributions are:

- An industrially proven generic and extendable anomaly detection architecture that can be easily customized for different domains.
- A number of techniques that enable the significant reduction of computational resources and disk space requirements, geared for scalable anomaly detection.

- A novel approach to control a number of generated alerts within a predefined budget, while preserving the most significant outliers by learning historical outlier distribution.
- A scalable cloud-based architecture that can analyze the Big Data produced by large IoT systems.

1.2 Related Work

The main approaches of anomaly detection use statistical methods, machine learning, and data mining techniques. A wide range of anomaly detection techniques is reviewed in [10]. One of the most well-studied applications of anomaly detection in the security domain is network intrusion detection. Bhuyan et al. [6] provides a comprehensive survey on anomaly-based network intrusion detection techniques. However their usage for data protection is limited. Deorankar et al. [9] survey anomaly detection techniques based on Machine Learning for IOT cyber-security. A number of intrusion detection techniques, datasets and challenges are surveyed in [11]. IoT anomaly detection has been suggested at the device, router, and cloud levels. Butun et al. [7] survey different anomaly methods and their applicability to IoT and cloud applications. Yu et al. [17] proposed unsupervised contextual anomaly detection method in IoT through wireless sensor networks. Yan et al. [16] use a classification algorithm based on mini-batch gradient descent with an adaptive learning rate and momentum to detect network anomalies in IIoT. Liu et al. [13] and Arrington et al. [4] use artificial immunity to detect anomalous behavior of IoT devices at the router level. Mehnaz and Bertino [14] designed a privacy-preserving real-time anomaly detection using edge computing that detects point, contextual, and collective anomalies in streaming large scale sensor data while preserving the privacy of the data. A scalable, cloud-based model to provide a privacy preserving anomaly detection service for quality assured decision-making in smart cities described in [2].

Scalability is a well known challenge for anomaly detection. Koufakou [12] proposes the Attribute Value Frequency (AVF) algorithm as an efficient approach for anomaly detection of categorical data. AVF is compared to a number of Frequent Itemset Mining algorithms based on the Apriory algorithm [1] and shows much better run-time without losing accuracy. Another efficient technique is described by Bertino et al. [5]. Here the Naive Bayes Classifier method is used to obtain linear scalability for the training phase and very fast analysis. Moreover, Bertino et al. compare the use of three different levels of information granularity: coarse, medium-grained, and fine. They show that on real data the achieved precision and accuracy of outlier detection is very similar for all the levels, but the execution times are a few orders of magnitude faster for coarse information granularity. We chose a multi-level approach to improve scalability. At the algorithmic level, we pervasively use exponential smoothing. At the architecture level, deploying the system in IBM's cloud environment Apache SparkTM for a quick and easy storage and performance scale-up.

2 Cloud-Based Architecture

The architecture we use to deploy the outlier detection system on the cloud is depicted in Fig. 1. The main input to the system is an Apache KafkaTM stream of event descriptions, where each event is described in terms of specific features. For example, Table 1 shows part of a stream of events, each corresponding to some communication between a particular device and the IoT management system being used. The features include some meta data describing the communication, such as the time, device owner (*Client*), device ID, action type (e.g., send, connect, or disconnect), and whether the action failed or not. In the case of a *send* action - the sent payload can also be analyzed. Thus, the system can detect anomalies related to meta-data of the message, such as the software used to send the message (*SW*), and the message size and format (which can be automatically computed). We can also analyze semantic features of the message, such as temperature readings (*T*).

Table 1. An event stream of a camera IoT system

Time	Client	Device	Action	Fail	Payload
07:06	C1	Cam1	Send	0	{ T=20, SW=S1...}
07:07	C2	Cam2	Send	0	{ T=19, SW=S2...}
07:07	C2	Cam5	Connect	1	
07:10	C1	Cam1	Send	0	{ T=56, SW=S1...}
07:13	C1	Cam4	Connect	1	
07:14	C1	Cam4	Connect	1	

Our outlier detection system defines a specific interface for the event stream. The event stream arrives on a dedicated Kafka stream and in a particular format, which includes an identification of the device involved in the event and the set of event features in JSON form. The outlier detection system is thus general purpose in the sense that it can analyze any event stream that conforms with this interface. The IoT devices communicate using messages of different forms and purposes arriving on multiple MQTT *topics*. Messages are then forward to a central *Message Hub* cloud service from which they arrive at the *input bridge* shown in Fig. 1. The *input bridge* converts all these variously formed messages into the specific form defined by the interface. Beyond simple formatting, the *input bridge* can also compute features from the incoming messages. For example, it might compute the payload size, determine the message format (as features to analyze), or extract other feature from the payload data.

The outlier analysis takes place on a cloud environment, the same cloud environment used. The outlier detection runs on a Spark service. Using Spark makes the solution very scalable because the resources needed for the analysis (memory, disk, CPU, communication) can be adjusted if the analyzed IoT system

grows (or shrinks) without any manual changes required in the code or in other parts of the system.

The outlier detection includes two main units running in a pipeline: aggregation and scoring (see Fig. 1). The aggregation unit takes the input event stream described above and converts it into a stream of features aggregated for pre-defined time windows (termed *analysis periods*). For example, it could compute the total payload size sent by a device during the last 15 min. When the analysis period ends, the aggregator streams all the aggregations to the scoring unit. In the example just given, the aggregated input feature is the payload size, the aggregation type is *summation*, and the analysis period is 15 min. This specific setting is defined in a configuration kept on a Cloud storage service (*Object Storage* shown in Fig. 1). The configuration typically defines several aggregations; some are general purpose (e.g., the number of failures, or communication frequency) and some are specific to the IoT domain.

The scoring unit receives the stream of aggregations and uses these to update *behavioral models* of the analyzed entities. The entities being modeled are defined for every specific domain; they typically include all the devices, device types, and clients in the system. We model the devices so that we can detect anomalous device behavior. We model the device types so that we can detect an attack on an entire class of devices, for example via malware that attacks some particular firmware. Modeling clients can help detect a broad attack where many devices exhibit minor changes in behavior that only appear abnormal when examined together in aggregate.

The scoring unit can now compare the incoming aggregations with the behavioral models of the entities involved. If the difference is large enough, an alert is sent on the output *alerts* Kafka stream. The alert is sent along with enough information to justify and explain the reason for the alert. This includes the observed anomalous feature along with the corresponding statistical information from the model (e.g., the historic average and standard deviation). An alerts-consumer now reads this stream of alerts and can store or display these in a dedicated GUI.

The data continuously streams along the system units, from the devices sending their events to Watson IoT, to the output alerts stream. Thus, data is almost never at rest, except for the short terms during the windowed aggregation and for the summaries kept in the models for longer periods. This means we do not have to keep the very large quantities of data that can be expected in IoT system logs. That said, the system only knows the summary information in the model and does not have access to the entire precise historical data. If such historical data is needed (e.g. for detailed post-mortem attack analysis), then the history should be backed up via a separate facility.

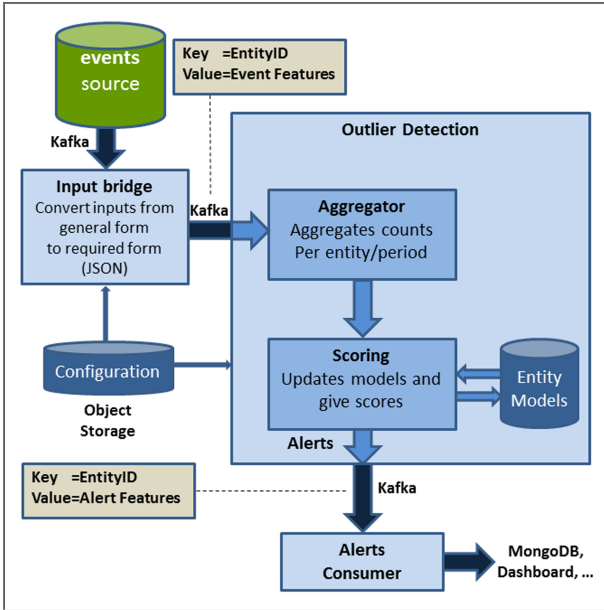


Fig. 1. Cloud architecture

2.1 Anomaly Detection Unit

As described above, the system models the typical behavior of certain **modeled entities** such as devices, device-types, clients, end-users, and geographic locations. This allows the system to later detect deviations from the normal modeled behavior. We model these behaviors using various features that are aggregated during **analysis periods** of fixed length. For example, the model may note the average and variance of the number of failures the entity has per analysis period.

The events streaming into the system are described in terms of **features**, some of which are used to identify the modeled entity involved in the event (e.g., the Device and Client features in Table 1). Other features are aggregated over time periods for analysis purposes. These are termed **aggregated features** (e.g., the payload size, and failure counts). An **event meta-type** is a tuple of event features names. Consider the example features SW , $Action$ of the event stream of Table 1, and the computed feature of the message format - $Format$. The defined **event meta-types** may include, for example, $\langle SW, Format \rangle$, $\langle SW, Action \rangle$, and $\langle Action \rangle$. An **event type** is an instantiation of an event meta-type that assigns specific values to the features named in the event meta-type. For example, $\langle SW = S1, Format = JSON \rangle$ is an event-type of the event meta-type $\langle SW, Format \rangle$. Every specific **event** in the event stream may correspond to a number of event types, each relating to different event features. For example, the first record in Table 1 corresponds to (at least) the event types $\langle SW = SW1, Format = JSON \rangle$, $\langle SW = SW1, Action = Send \rangle$, and

$\langle Action = Send \rangle$. The system uses a fixed collection of event meta-types for every modeled entity type. So, for example, the system may define the event meta-types $\langle SW, Format \rangle$, $\langle SW, Action \rangle$, and $\langle Action \rangle$ for device entities, and the event meta-types $\langle device \rangle$, $\langle SW, Action \rangle$ for client entities. For every event, the system creates a list of modeled entities and corresponding event types that match the event. For example, for the first record in Table 1, the system identifies the modeled entity $\langle Device = Cam1 \rangle$ and corresponding event types $\langle SW = SW1, Format = JSON \rangle$, $\langle SW = SW1, Action = Send \rangle$, and $\langle Action = Send \rangle$. It also identifies the modeled entity $\langle Client = C1 \rangle$ and corresponding event types $\langle Device = Cam1 \rangle$, $\langle SW = SW1, Action = Send \rangle$. Note that the feature *Device* sometimes serves to identify a modeled entity and sometimes serves as part of an event type. Every analysis period (e.g., 15 min), the aggregation unit (see Fig. 1) creates an **event type summary** for every modeled entity. The summary specifies the number of events of each event-type that occurred with the entity. The summary also includes aggregations of features for every event type used by the entity - such as the total number of failures, or maximum payload size. In our example, the system would calculate that during the period 7:00–7:15, the modeled entity $\langle Client = C1 \rangle$ had 2 events of the event type $\langle Device = Cam4 \rangle$, aggregated with 2 fails. It also had 2 events of the event type $\langle SW = SW1, Action = Send \rangle$ aggregated with no fails. Thus, the aggregation unit aggregates the aggregated features for every event type per modeled entity in the event type summary. It can also compute the total aggregation of the aggregated features for every entity, regardless of the event types involved. Thus, we can configure the aggregator to compute the total number of fails, and average or maximum payload size for every entity.

The aggregation unit runs on a Spark service as a collection of distributed processes that perform the aggregations jointly in a map-reduce approach. Even in very large IoT systems, where Big Data scales of traffic information needs to be analyzed, the system can still perform the aggregation with reasonable performance and can be simply scaled by adding more resources to the Spark cluster. The system was designed to keep a minimal amount of data during its operation. Once any piece of the data is streamed into the system and analyzed, it can be dropped. The temporary event summaries need only be stored, in the form of Spark RDDs (Resilient Distributed Dataset), for the duration of the analysis periods. Even the entity models that need to be kept for longer time spans while the entity remains active, keep relatively small amounts of summary data per entity. This is possible because the averages in the models are all smoothing averages; these can be updated incrementally after every period of activity. See Sect. 3.

2.2 Anomaly Scoring

The anomaly scoring is carried out concurrently by multiple Spark processes. Every **analysis process** is responsible for analyzing the activities of several modeled entities during the last analysis period. An analysis process considers all the event types and corresponding aggregated features from the event type

summary of the analyzed modeled entity. It compares the behavior observed in the last analysis period to the entity's normal behavior as modeled. It then updates the behavioral model of the entity. This is carried out by a collection of anomaly **scorers**, each scorer responsible for analyzing, modeling, and checking a particular behavioral aspect and producing a score. The scores represent the risk or probability of an attack and are therefore given as numeric values between 0 and 1. A higher value corresponds to a higher probability of an attack.

For example, the **volume** scorer looks for an unusually high number of events of some event type during the analyzed period. For this purpose, for every modeled entity, the scorer maintains some volume related statistics for *every* event type (e.g., $\langle SW = SW1, Action = Connect \rangle$). This includes the average volume observed so far, the average of the volume squares (to compute the variance), the top volume scores with representatives from recent weeks, and the number of samples that were used for computing the statistics i.e., the number of events the modeled entity had of this event type in the past. All the averages and counts here are computed using smoothing average techniques (see Sect. 3) and are computed in a logarithmic scale. Using logarithmic scales is a conventional approach when analyzing volumes, since their order of magnitude is typically more interesting than the precise volume.

These statistics are, in effect, the volume-related part of the model for each entity. The volume scorer is responsible for updating this part of the model by referring to the aggregated volume in the event type summary of the entity. But before doing so, the scorer would give a volume anomaly score to the event type by comparing the count in the event type summary to the related statistics in the model. For example, it could compute a risk based on the Z-score from the newly observed volume and from the modeled average and variance; it would then “moderate” the score by considering the number of samples and the maximal values observed so far. A combined volume score is then computed for the modeled entity by taking the top scoring event types and computing some weighted average of their scores.

The volume-related part of the model includes statistics for individual event types alongside general statistics of the modeled entities. Most other scorers require more compact models that only include general statistics of the modeled entities. For example, a **new** activity types scorer models the typical number of new behaviors that a modeled entity (client or device) exhibits every period. A **variability** scorer tracks the general variability of a device's activity. If a device normally exhibits small variability (i.e., repeats having more or less the same diversity of activities) and suddenly behaves with much greater variability, this could indicate an account hijacking.

Alert generation is the final step in the analysis process for a period. Here, the multiple scores coming from the various scorers are weighted and combined for every modeled entity. If the resulting score is above some threshold (automatically set by the budgeting scheme - see Sect. 4), an alert is produced and displayed in the system GUI. The GUI also gives the reasons for the alert in a user friendly display of the related statistics, and allows the end user to further

investigate the root causes for the alert (e.g., by displaying all the related device activities).

2.3 Generic Architecture

The system is generic in the sense that it can easily be configured to find anomalies in completely different domains, in logs of different types of systems, and for different purposes. As mentioned above, beyond the IoT use case, the tool is currently part of an IBM product where it reports possible insider attacks by analyzing the logs of user database activity. The tool was also configured to serve other use cases, including detecting anomalies in logs of storage devices to predict pending device failures, and to detect anomalies in the logs of a cloud-based object store.

Configuring the tool for a specific domain includes the following steps:

Model Design: The model consists of entities and corresponding event meta-types. These are defined in terms of event features that are available or that can be computed. This step is carried out by experts in the domain and depends on the particular use case for the anomaly detection. In our example above, the experts decided to model the behavior of devices, device types, and clients. When selecting meta-types, the choice of the meta-type $\langle SW, Action \rangle$ was made so that abnormal volumes of different types of actions by some software can be detected. The choice of the meta-type $\langle Action \rangle$ was made so that abnormal usage volumes for specific action can be detected, regardless of the software involved.

Preparing the Event Stream: The IoT system should forward the events to be analyzed to the dedicated Kafka topic used by the input-bridge. The input-bridge must be set up to collect or compute the chosen event features from the input event stream before forwarding them to the outlier detection units (see Fig. 1).

Domain Specific Scorers: During system configuration, one can select relevant scorers from a library of available generic scorers. The overall volume and failure scorers described above are generic in the sense that they are relevant to many IoT domains. However, one can also add new scorers that are relevant *only* to the specific domain being configured. In a camera IoT for example, a scorer can be added to detect anomalies in the number of automatic movement operations that the camera reports in its message payloads.

3 Exponential Smoothing Averages

The behavioral models we use for anomaly detection rely on computing the average and standard deviation of various properties, e.g., the number of operations of a particular type that a particular device performs per hour. These statistics can then be used to detect anomalies. If in the current hour, the number of operations of this type is far from the average by a large enough number of standard deviations, an alert is issued.

Say V_1, V_2, V_3, \dots is a stream of observed values, where V_i is observed at the i 'th period. One way to compute the average at time i is simply as the average of values observed until this point in time, $(V_1 + V_2 + \dots + V_i)/i$. This has the advantage of being easy to update incrementally, without the need to store the values observed prior to period i . However, it has the disadvantage that it gives equal weight to all observed values, old and new. In reality, a device's behavior is gradually changing, so older values should have a decreased impact on the current estimate of the average.

Another approach is to store a sliding window of the last k observed values, $V_{i-k+1}, V_{i-k+1}, \dots, V_i$, and compute the average over this window. This has several disadvantages. One disadvantage relates to the way old values are treated. Older values within the window are weighted the same as the newer values in the window. When a value drops off the window, it abruptly gets completely forgotten and no longer has any effect on the current estimate of the average. Another disadvantage is that performance issues may arise. Assuming we are tracking a large number of properties and use large windows, storing this data may require a large space, and maintaining it as new data arrives may be slow.

Since our system must work on Big Data, efficiency considerations are crucial. Therefore, we chose a third approach: exponential smoothing averages, referred to as smoothing averages below. Denoting the smoothing average at time i by S_i , the basic formula is as follows.

$$S_i = V_i\alpha + S_{i-1}(1 - \alpha), \quad (1)$$

where α is some predefined constant. A possible method for choosing α is to select the window size that will have a total influence weight of 0.5 on the smoothing average result. For a window of n periods, this is $\alpha = 1 - 0.5^{1/n}$.

A model composed of a set of smoothing averages has several advantages: much smaller model size, efficient model updates and accounting for both recent and historical data.

3.1 Smoothing Average Behavior

We demonstrate the behavior of the smoothing average scheme over specific test cases. We created the test cases by generating independent and identically distributed samples of normal distributions with a standard deviation of 1 and with means that were chosen to simulate various changes in behavior, as detailed below. One additional test case was taken from real data gathered on a running instance of our system at a customer site. In all test cases, we show the smoothing average approach with initialization and gap handling. The α is configured such that the weight of 25 time periods will be equal to 0.5 ("smoothing average" in the figures). We compare this to a simple sliding window scheme of 50 periods ("sliding window" in the figures).

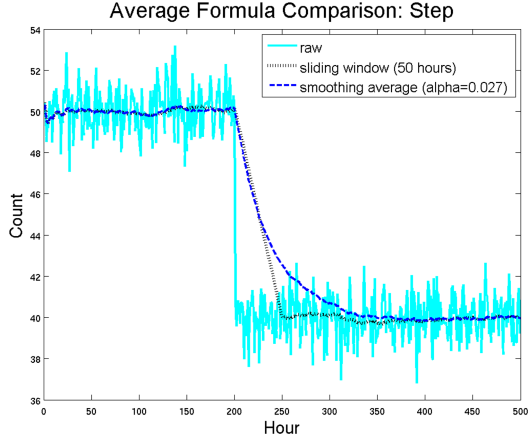


Fig. 2. Comparison of sliding window and our scheme of smoothing average: Behavior during and after an abrupt change in input

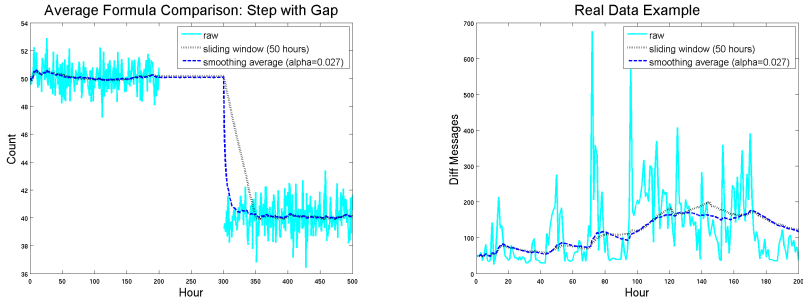
Figure 2 shows the reaction to an abrupt change in the input. Both approaches are similar, except the smoothing average takes a bit longer to fully adapt to the change. In Fig. 3a, we show the same thing, except there’s also a gap of inactivity between the old (high) values and the new (low) ones. Since it gives lower weights to older values, here the smoothing average adapts to the change more quickly.

Figure 3b shows real data giving the number of different unique operations per hour performed by a user. During most of the sampled period there are no gaps and only moderate changes. As a result, the two measures behave similarly, but employing the smoothing average is more efficient computationally. Some differences do exist and can be explained. During hours 130–150 the sliding window average increases with no apparent justification. This happens because the very low measures of hours 80–90 are being removed from the window. The smoothing average more intuitively tracks the data and actually decreases during hours 130–150, by giving higher weights to the new measures.

4 Alert Budgeting

Outlier detection systems are characterized by the dominance of legitimate behaviors in the training data. Malicious examples are typically missing from the training data or are too few to be representative. The commonly used approach in such a setup is to train models that describe the typical behavior of entities, including typical deviations from the average behavior. The trained models are then used to score the entities’ current behaviors according to their deviations from the typical historical behavior.

In an ideal world, the malicious behaviors’ deviations would be much larger than the deviations of legitimate behaviors. Thus, one could just look for large



(a) Behavior during and after an abrupt change in input that occurs after a gap of inactivity (b) Behavior over real-world counts of unique number of operations per hour performed by a user

Fig. 3. Comparison of sliding window and our scheme of smoothing average

enough deviations from the historical data and expect to detect all malicious events, without producing any false positives for legitimate behaviors. In practice, it is not exceptional that some legitimate behaviors exhibit a high level of abnormality. Meanwhile, sophisticated attacks try to mimic legitimate behaviors to escape detection by outlier detection systems. One of the important questions is how to choose the threshold above which a behavioral deviation should trigger an alert to the security operator. If the threshold is set too low, the rate of false alerts is too high and exceeds the capacity of the operators. On the other hand, if it is set too high, the chance of missing malicious behaviors significantly increases.

There are a number of existing approaches for the selection of a threshold. The most basic approach requires security operators to manually select the threshold. The problem with this approach is that it is not clear how to come up with a good threshold. A good threshold can differ in different domains and may also change over time. It can lead to too many alerts emitted at some periods, and to no alerts at all even when high risk events occur. Another approach allows you to define a limited budget for the number of alerts emitted per a predefined interval, such as 48 alerts per day. In this case, waiting 24 h and reporting 48 top alerts introduces a reporting delay that is impractical. On the other hand, reducing the delay by reporting alerts each hour (in this case 2 alerts per hour) is also not optimal. The top scoring events typically have low scores for most of the hours, but the system would still trigger 2 alerts per hour. This makes operators less sensitive to the alerts. In addition, hours with many events with high scores will be limited to 2 alerts, so some high risk events might be missed. A number of works [3] propose various optimization methods for thresholds selection (see Sect. 1.2).

In this work, we developed a novel approach that defines an **alert budget** B as the rate of alerts that operators decide they can handle. Given the alert budget, the system periodically re-adjusts the threshold automatically based on historical scores, so the average rate of emitted alerts matches the predefined alert budget.

The alert budget is designed to be satisfied on average and not for every period, so one can expect some fluctuations in the number of alerts per period. To constrain the variation of the number of triggered alerts, we introduce a **max alert number** parameter K . The max alert number is a “hard” constraint, so if for a specific period there are more than K events with scores above the threshold, only the top K events with the highest scores are reported. The max alert number should be set to a large enough value to allow a reasonable level of alert fluctuations, but not too large to avoid a situation where a few non-typical hours (e.g., system upgrade) produce too many alerts and, as a result, raise the threshold too high. A possible way to select K is $K = \max(10, 10Bd)$, where d is the duration of a single period. This restricts one period from producing more than 10 times more alerts than the average per period, or more than 10 alerts for the case that $B < 1$.

The alert budgeting approach has two important aspects for anomaly detection in a Big Data setup. First, it allows us to define and automatically control the average rate of alerts produced by the anomaly detection system. So even for very high rates of monitored events, only a predefined rate of alerts will be produced. The target alert rates should be selected according to the human resources available. Second, as we show below, our approach uses a fixed amount of disk space that doesn’t depend on the volume of monitored data and takes only a small portion of the processing time.

4.1 High-Level Budgeting Flow

The high-level flow for alert budgeting is depicted in Fig. 4. Given the alert budget B , the flow starts from scores that are periodically produced by the scorers. For each period and for each scorer, the top K scores are inserted into the **top scores** tables, where K is the max alert number parameter.

The top scores table keeps a sliding window of t historical periods. The size t of the sliding window determines how fast alert budgeting reacts to the changes in scores. For example, when a one-day sliding window is used, the system is significantly influenced by a local change of a few hours duration, which makes the system unstable. On the other hand, a one-year sliding window makes the system react very slowly to the changes, so it might take a few months for the system to start reacting to a change. For our industrial use case, we used a one-week sliding window; that gave us the right balance between the agility and stability of the system.

Based on the historical top scores, the **threshold calculation** step computes **scorers thresholds**. The score thresholds are computed in such a way that the number of historical events above the threshold matches the target alert budget. If the top scores of new periods have a similar distribution to the historical top scores, the average number of future events above the threshold will also match the target alert budget. However, if the score distribution changes, (e.g., higher scores appear more frequently) the historical scores gradually accumulate in the new distribution and the score thresholds will change accordingly.

The score thresholds are used in the **scores normalization** step to normalize different scores to the same scale. Only scores from previous periods are used to compute the current period thresholds; in this way the thresholds can be calculated before starting the analysis. This allows scores to be normalized in a distributed way, without having to wait for the processing of all the events of the current period.

The **final score** is computed based on the normalized scores and is used to determine whether an alert should be reported.

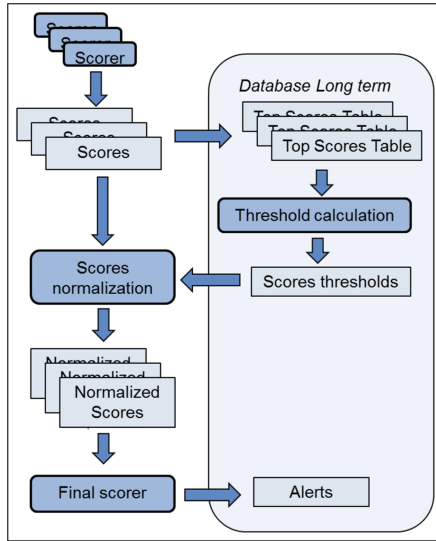


Fig. 4. Alert budgeting

4.2 Scalability

While the alert budget is not guaranteed to produce the same number of alerts for each period, on average it provides the required rate of the alerts. This allows the planned allocation of the resources needed to investigate the alerts and enables security operators to effectively monitor systems with high volumes of activities.

Using the alert budget scheme doesn't require keeping a full record of historical events; only a very limited portion is required. The critical parameters here are the number of scorers (N), the length of the sliding window of historical periods (t), and the number of top scores held for each scorer K that equals the max alert number parameter. The total number of records R required to be stored is tKN .

For example, for 5 scorers ($N = 5$), a sliding window of historical periods equal to one week ($t = 168$) and a max alert number $K = 10$, the resulting total number of records is $R = 8400$ and doesn't depend on the total volume of activities.

5 Results

To evaluate budgeting quality we used real production data collected over 3 months (from an undisclosed European telecom company). Traffic from 7949 internal-system users was recorded and analyzed. We fixed the *anomaly budget goal* to $B = 2$ alerts per day, and the *history training sliding window length* to $t = 7$. Next, we compared the budgeting quality for a variety of budget-factor update frequencies. Thus, we are able to compare our proposed automatic, hourly threshold update frequency to a manual update at different frequencies. This test can be used to assess how frequently a human operator would have to change the budget factor in order to match the performance of an automatic system. The performance is measured as the difference between the actual alert rate and the target budget goal line. To this end we compare B with the number of cumulative alerts up to each day, divided by the cumulative number of days.

Figure 5 plots for every date the number of alerts during a 7-day sliding window. It can be observed that for a 30-days update frequency, the 14 alerts per week goal ($B = 2$ alerts per day) is hardly met. Specifically, there is a strong under-budget trend from mid-September till mid-October. At the beginning of November, the 7-day and 30-day update lines are much higher than the goal line, while towards mid-October they are both much lower than the goal line. These drastic fluctuations result from the specific day when the factor is calculated. Since both the 30-day and 7-day factors are static for a very long time, calculating the budget factor at a peak or a low point of a monthly cycle has a lasting effect. In contrast the 1-hour update frequency is much more dynamic and is consistently much closer to the 14 alerts per week goal line. It can quickly adjust, and so it is impacted much less intensely from calculating the budget factor at a local low point (as can be seen at the beginning of November) and for a shorter time frame. In fact, when we compute the mean absolute difference from the 14 weekly alerts goal, we find that the hourly update frequency has by far the lowest mean of 3.8 as compared to 10.7 for the 30 day frequency.

Finally, we used a configurable simulator to assess speed performance, measured as the number of records analyzed per second. This is dependent on the number of instances, the CPU, and memory specifications. We tested our system on a local machine with 16 GB RAM and an i7 Intel 8-core processor (2.7 GHz) and were able to reach 200K records per second. Deploying the system to the cloud, we observed a near-linear scale up in performance dependent on the number of executors. Speed almost doubled when we went from a single executor to 2 and again from 2 to 4. Thus, we have confirmed our assumption of scalability under the Spark framework.

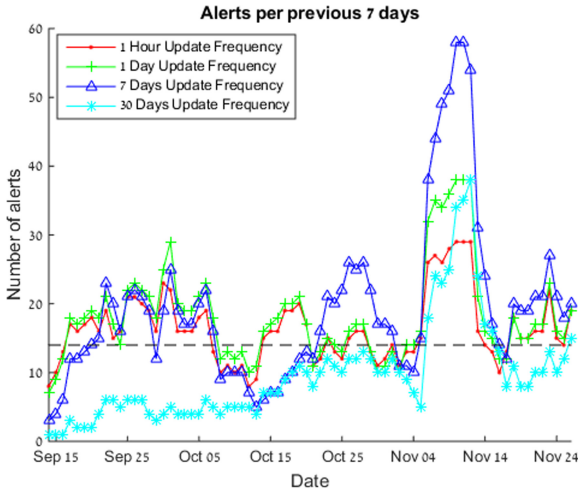


Fig. 5. Alerts per previous 7 days. The dashed line represents the 14 alerts per week goal line.

6 Summary

We presented a novel, scalable, generic, and industrially-proven system that uses anomaly detection to identify security attacks on IoT cloud platforms. The system includes a novel approach that handles the alerts, offloading the cognitive burden of the human through semi-automaton. The system uses historical data to calibrate alert thresholds and achieve the target rate on average, while still properly handling periods where anomalous behavior is exceptionally high or low.

The scalability of the system is achieved using a distributed cloud-based scalable architecture (i.e., Spark and Kafka), and efficient modeling techniques, specifically smoothing averages with gaps. The system can be deployed in multiple settings using a configuration interface to define the modeled entities (the objects that are being tracked for anomalous behavior), and the event meta types (combination of features that define distinct events). The configuration interface further allows users to select the scoring algorithms used from a library, and to extend it with new ones.

Our future work plans include continuing and enhancing the system’s use in the IoT domain, and for additional fields beyond security. We have already had success in early experiments predicting device failures in storage systems, and we plan to further explore using the outlier system for quality-related use cases. For algorithm enhancement and improved precision, we plan to introduce the clustering of modeled entities based on similarity of behavior. The anomaly detection for an individual modeled entity will consider the particular entity as well as similar entities in its cluster. More future directions include addressing adaptivity and increased automation.

Acknowledgements. Part of this work has been carried out in the scope of the FINSEC project (contract number 786727), which is co-funded by the European Commission in the scope of its H2020 program. The authors gratefully acknowledge the contributions of the funding agency and of all the project partners.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of 20th International Conference very large data bases, VLDB. vol. 1215, pp. 487–499 (1994)
2. Alabdulatif, A., Khalil, I., Kumarage, H., Zomaya, A.Y., Yi, X.: Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities. *J. Parallel Distributed Comput.* **127**, 209–223 (2019)
3. Ali, M.Q., Al-Shaer, E., Khan, H., Khayam, S.A.: Automated anomaly detector adaptation using adaptive threshold tuning. *ACM Trans. Inf. Syst. Secur. (TIS-SEC)* **15**(4), 17 (2013)
4. Arrington, B., Barnett, L., Rufus, R., Esterline, A.: Behavioral modeling intrusion detection system (bmid) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms. In: Computer Communication and Networks (ICCCN), 2016 25th International Conference on. pp. 1–6. IEEE (2016)
5. Bertino, E., Terzi, E., Kamra, A., Vakali, A.: Intrusion detection in rbac-administered databases. In: 21st Annual Computer Security Applications Conference, pp. 10. IEEE (2005)
6. Bhuyan, M., Bhattacharyya, D.K., Kalita, J.: Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools (2017). <https://doi.org/10.1007/978-3-319-65188-0>
7. Butun, I., Kantarci, B., Erol-Kantarci, M.: Anomaly detection and privacy preservation in cloud-centric internet of things. In: 2015 IEEE International Conference on Communication Workshop (ICCW), pp. 2610–2615. IEEE (2015)
8. Cook, S.: Ddos attack statistics and facts for 2018–2020 (2020). <https://www.comparitech.com/blog/information-security/ddos-statistics-facts/>
9. Deorankar, A.V., Thakare, S.S.: Survey on anomaly detection of (iot)-internet of things cyberattacks using machine learning. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 115–117. IEEE (2020)
10. Eltanbouly, S., Bashendy, M., AlNaimi, N., Chkirbene, Z., Erbad, A.: Machine learning techniques for network anomaly detection: A survey. In: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), pp. 156–162. IEEE (2020)
11. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* **2**(1), 1–22 (2019). <https://doi.org/10.1186/s42400-019-0038-7>
12. Koufakou, A., Ortiz, E.G., Georgiopoulos, M., Anagnostopoulos, G.C., Reynolds, K.M.: A scalable and efficient outlier detection strategy for categorical data. In: 19th IEEE International Conference on Tools with Artificial Intelligence, 2007. ICTAI 2007, vol. 2, pp. 210–217. IEEE (2007)
13. Liu, C.M., Chen, S.Y., Zhang, Y., Chen, R., Guo, K.L.: An iot anomaly detection model based on artificial immunity. In: Advanced Materials Research. vol. 424, pp. 625–628. Trans Tech Publications Ltd (2012)

14. Mehnaz, S., Bertino, E.: Privacy-preserving real-time anomaly detection using edge computing. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 469–480 (2020). <https://doi.org/10.1109/ICDE48307.2020.00047>
15. Stelios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., Lopez, J.: A survey of iot-enabled cyberattacks: assessing attack paths to critical infrastructures and services. *IEEE Commun. Surv. Tut.* **20**(4), 3453–3495 (2018)
16. Yan, X., et al.: Trustworthy network anomaly detection based on an adaptive learning rate and momentum in iiot. *IEEE Trans. Ind. Inf.* **16**(9), 6182–6192 (2020)
17. Yu, X., et al.: An adaptive method based on contextual anomaly detection in internet of things through wireless sensor networks. *Int. J. Distr. Sensor Netwk.* **16**(5), 1550147720920478 (2020). <https://doi.org/10.1177/1550147720920478>

Computer Vision and Dataset for Security



An Advanced Framework for Critical Infrastructure Protection Using Computer Vision Technologies

Krishna Chandramouli¹(✉) and Ebroul Izquierdo²

¹ Venaka Media Limited, 393, Roman Road, London E3 5QS, UK
k.chandramouli@venaka.co.uk

² Multimedia and Vision Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK
ebroul.izquierdo@qmul.ac.uk

Abstract. Over the past decade, there has been unprecedented advancements in the field of computer vision by adopting AI-based solutions. In particular, cutting edge computer vision technology based on deep-learning approaches has been deployed with an extraordinary degree of success. The ability to extract semantic concepts from continuous processing of video stream in real-time has led to the investigation of such solutions to enhance the operational security of critical infrastructure against intruders. Despite the success of computer vision technologies validated in a laboratory environment, there still exists several challenges that limit the deployment of these solutions in operational environment. Addressing these challenges, the paper presents a framework that integrates three main computer vision technologies namely (i) person detection; (ii) person re-identification and (iii) face recognition to enhance the operational security of critical infrastructure perimeter. The novelty of the proposed framework relies on the integration of key technical innovations that satisfies the operational requirements of critical infrastructure in using computer vision technologies. One such requirement relates to data privacy and citizen rights, following the implementation of General Data Protection Regulation across Europe for the successful adoption of video surveillance for infrastructure security. The video analytics solution proposed in the paper integrates privacy preserving technologies, high-level rule engine for threat identification and a knowledge model for escalating threat categories to human operator. The various components of the proposed framework has been validated using commercially available graphical processing units for detecting intruders. The performance of the proposed framework has been evaluated in operational environments of the critical infrastructure. An overall accuracy of 97% is observed in generating alerts against malicious intruders.

Keywords: Person detection · Person re-identification (RE-ID) · Intrusion detection · Face recognition · Region based Fully Connected Network (RFCN) · Un-supervised clustering · Knowledge model · Privacy preserving technologies

1 Introduction

Modern critical infrastructures are increasingly turning into distributed, complex cyber-physical systems that require proactive protection and fast restoration against physical or cyber incidents or attacks. Addressing the challenges faced by the critical infrastructure operators especially responsible for the production and distribution of energy services, DEFENDER¹ project has developed several cyber-physical detectors and operational blueprints to safeguard the future European Critical Energy Infrastructure (CEI) operation against new and evolving threats. In complementary to the cyber threats, the nature of physical threats is compounded by the use of drones in addition to the human intrusion against the infrastructure with malicious intent.

Following the increasing threat of malicious activity carried out against critical infrastructure by human actors there is an exponential increase in the deployment of surveillance infrastructure such as Closed Circuit Television (CCTV) for monitoring the perimeter of critical infrastructure. Traditionally in computer vision research, the task of object detection is to classify a region for any predefined objects from the training data set. Early attempts at object classification also adopted a similar approach for the detection of an image region to be consisting of a drone or not. In this context, the application of computer vision was applied for the selection of suitable representations of objects using handcrafted features. The most successful approaches using handcrafted features require Bag of Visual Words (BoVW) was reported in [24] that includes representations of the objects with the help of local feature descriptors such as Scale Invariant Feature Transform (SIFT) [18], Speeded-Up Robust Features (SURF) [2], and Histogram of Oriented Gradients (HOG) [8]. After training a discriminative Machine Learning (ML) model such as Support Vector Machines (SVM) [6], with such representations the images are scanned for occurrence of learned objects with sliding window technique. In contrast to classical machine learning approaches, the increasing use of deep-learning algorithms has led to the development of object classification and localisation. Two main approaches have been reported in the literature that adopt two different strategies namely (i) two-stage detector, the most representative one, Faster R-CNN; and (ii) one-stage detector, such as YOLO, SSD. Two-stage detectors have been reported to indicate high localization and object recognition accuracy, whereas the one-stage detectors achieve high inference speed. The two stages of two-stage detectors can be divided by RoI (Region of Interest) pooling layer. For instance, in Faster R-CNN (Region based Convolutional Neural Network), the first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes. The second stage, features are extracted by RoI Pool(RoI Pooling) operation from each candidate box for the following classification and bounding-box regression tasks [15]. The research presented in the article focusses on the use of two-stage detector due to the high-accuracy of the deep-learning network model.

¹ <https://defender-project.eu/>.

Despite the success of computer vision technologies in addressing real-world applications within research environments, the deployment of such solutions within operational environment requires additional services that takes into consideration the data privacy and citizen rights. Addressing these crucial requirements, the paper presents the activities carried out in DEFENDER project, for the development of the an intelligent framework using computer vision technologies for enhancing the critical infrastructure security. The proposed framework integrates three key technologies namely (i) person detection; (ii) person re-identification (RE-ID) and (iii) facial recognition (FR) components to enrich critical infrastructure security. The overall complementarity of these technologies are leveraged to enhance the perimeter security of critical infrastructure against physical attacks. The processed outcome from these three components are subsequently analysed based on stream analytics to enhance the robustness of the detection through metadata association between individually processed frames captured from the camera sensor. Additionally, the use of privacy-preserving technologies based on Gaussian blur algorithm ensures the compliance of the framework for data privacy. The framework interfaces with the command centre to visualise the alerts generated from the critical infrastructure. The framework integrates an encrypted media repository to comply with privacy-by-design (PbD) principles.

The rest of the paper is structured as follows. Section 2 an overview of the studies presented in the literature for person detection, person re-identification and face recognition are presented. Subsequently, in Sect. 3 the proposed intelligent situational awareness framework is presented, which incorporates several key technical innovations including the use of privacy preserving technologies and secure encryption services. Following a detailed analysis of each of the technical innovation, the outcome of the proposed framework is presented in Sect. 4. The paper concludes with observations, remarks and roadmap for the design of physical security detector in Sect. 5.

2 Literature Review

Intelligent video surveillance has been one of the most active research areas in computer vision [29]. Within the computer vision community, the research of person detection has attracted studies from interdisciplinary scientists, interested in the design of autonomous vehicles to intelligent surveillance [14], robot navigation [10,16]. In this section the literature review of three technologies are summarised in two categories as follows.

2.1 Person Detection and Person Re-identification

Person detection is considered as an object detection problem for which the deep-learning models are trained on a number of human representations that takes into account the changes to appearance, cloths, and environment parameters among others. Prior to the recent progress in Deep-Convolutional Neural

Network (DCNN) based methods [26], researchers combined boosted decision forests with hand-crafted features to obtain pedestrian detectors [30]. In the literature, the problem of person detection has been extended to associate human representation captured from multiple-cameras. This has led to the research in single camera Multi-object Tracking (MOT) algorithms. In contrary the Multi-Target Multi-Camera Tracking (MTMCT) algorithms reported in literature are based on off-line method which requires to consider before and after frames to merge tracklets, and do post processing to merge the trajectory. In the literature, hierarchical clustering [33] and correlation clustering [22] are reported for merging the bounding box into tracklets from neighbor frames. Addressing the need to generate real-time tracker without the apriori knowledge of person tracks, an online real-time MTMCT algorithm has been reported in the literature which aims to track a person cross camera without overlap through a wide area. The framework utilises a person detection based on Openpose [4], building on a multi-camera tracker extended by a single camera tracker MOTDT [5]. In order to improve the performance, lots of research focus on local feature instead of the global feature of the whole person, such as slice [27], pose and skeleton alignment [34]. While matching local features help to improve in Person Re-ID, the challenge of pose variation remain open due to the different view from camera.

2.2 Face Recognition

Face recognition (FR) has been the prominent biometric technique for identity authentication and has been widely used in many areas, such as military, finance, public security and daily life [28]. In 2014, DeepFace [25] achieved the state of the art accuracy on the famous LFW benchmark [12], approaching human performance on the unconstrained condition for the first time (DeepFace: 97.35% vs. Human: 97.53%), by training a 9-layer model on 4 million facial images. Inspired by this work, research focus has shifted to deep-learning-based approaches, and the accuracy was dramatically boosted to above 99.80% in just three years. Deep learning technique has reshaped the research landscape of FR in almost all aspects such as algorithm designs, training/test datasets, application scenarios and even the evaluation protocol. In 2015 a system named Multi-task Cascaded Convolutional Networks (MTCNN) showed that a joint implementation for face alignment and detection could achieve higher levels of accuracy and thus has been integrated in the current implementation. Some noteworthy face recognition surveys include [3, 13, 23, 32]. These comprehensively survey face recognition systems prior to DeepFace. Hence, these surveys do not discuss the new sophisticated deep learning approaches that emerged during the last decade. Surveys that discuss deep face recognition have singled out face recognition as an individual discipline rather than a collection of components adopted from different studies. These surveys generally discuss the face recognition pipeline: face pre-processing, network, loss function, and face classification [17, 19] or discuss a single aspect of face recognition such as 3-D face recognition [28], illumination face recognition [20] or pose invariant face recognition [9]. Although these surveys

are important and provide an excellent basis for the analysis of the state-of-the-art in the field, they do not provide conclusive comparisons or analysis of the underlying network architectures.

3 Critical Infrastructure Protection Framework

The proposed critical infrastructure protection framework interfaces directly with the three detectors developed in the project and enables the construction of high-level surveillance events such as intrusion, loitering and access control authentication for early stage identification of malicious actions against critical infrastructure. The proposed framework is presented in Fig. 1 and consists of three stages namely (i) video sequence, captured from the sensor deployed at the perimeter of the infrastructure; (ii) computer vision technologies, capable of processing multiple video streams using deep-learning network and (iii) the situational awareness components, in which the organisational policies and practices are encoded to ensure the security restrictions are not violated by intruders. In order to protect the privacy and citizen rights, the proposed framework incorporates the use of privacy preserving technologies as outlined in Sect. 3.4. The processed outcome from the situational awareness component is then integrated into the command centre to categorise the threat and also the severity with which the mitigation actions should be carried out according to the organisational policies.

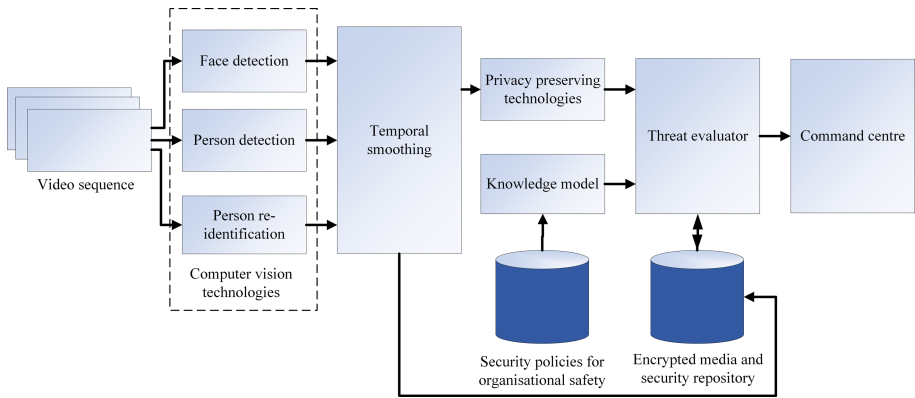


Fig. 1. Framework

3.1 Computer Vision Technologies

In this section, an overview of the implementation details carried out in the project for the integration of computer vision technologies is presented.

Person Detection and Person Re-identification (RE-ID). One of the challenges of MTMCT framework presented in the literature is the inability of the framework to anchor against a specific Person of Interest (POI) for modelling threat events such as loitering. In this regard, the MTMCT component has been further developed to include “unsupervised multi-camera person re-identification” framework. The overall framework design of the proposed framework is presented in Fig. 2. The implementation of the person detection component relies on the use of Region based Fully Connected Neural Network (RFCN), followed by the feature extraction of the detected person with a set of deep-learning features. The deep-learning features extracted from the identified bounding boxes are then subjected to the application of an unsupervised algorithm for clustering the people. The processing of the deep-learning features are further exploited to ensure the infrastructure operators can provide an anchor image of a POI, to retrieve the appearance of the person across several surveillance cameras.

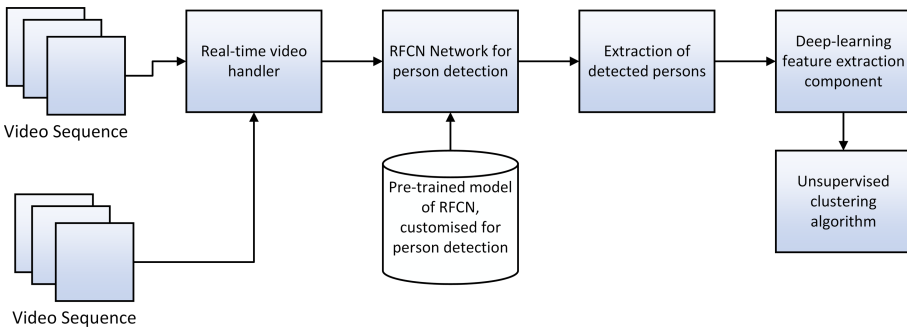


Fig. 2. Person re-identification

The novelty of the RFCN network relies in the consideration of two-stage object detection strategy namely (i) region proposal and (ii) region classification. The scientific rationale behind the use of the two-stage proposal is elaborated in [7]. Following the extraction of the regions (RoIs), the R-FCN architecture is designed to classify the RoIs into object categories and background. In R-FCN, all learnable weight layers are convolutional and are computed on the entire image. The last convolutional layer produces a bank of k^2 position-sensitive score maps for each category, and thus has a $k^2(C + 1)$ - channel output layer with C object categories (+1 for background). The bank of k^2 score maps correspond to a $k \times k$ spatial grid describing relative positions. For example, with $k \times k = 3 \times 3$, the 9 score maps encode the cases of top-left, top-center, top-right, ..., bottom-right of an object category. R-FCN ends with a position-sensitive RoI pooling layer. This layer aggregates the outputs of the last convolutional layer and generates scores for each RoI. In comparison with the literature [11], the position sensitive RoI layer in RFCN conducts selective pooling, and each of the

$k \times k$ bin aggregates responses from only one score map out of the bank of $k \times k$ score maps. With end-to-end training, this RoI layer shepherds the last convolutional layer to learn specialized position-sensitive score maps. Subsequent to the extraction of the people, the next steps is to extract deep-learning features from the blobs that are identified as people. Thus, for the purposes of DEFENDER project, where the infrastructure operator are only concerned tracking a specific perpetrator, such as person of interest, it is vital to adopt the solution to identify the anchor points, which are provided as input to the system. To address such a need, the use of unsupervised clustering is carried out, to cluster the blobs extracted from the RFCN network. Subsequently, the features used is also able to provide the infrastructure operator to identify and select a specific person who is considered a POI for the identification across multiple cameras. The implementation of the feature extraction has been carried out using two deep-learning network models namely (i) RESNET-18 resulting in the deep-learning feature of length 1×512 and (ii) AlexNet deep learning feature resulting in the feature length of 1×4096 . For the overall evaluation of the proposed unsupervised clustering framework, a set of videos across London has been captured with actors playing the role of person of interest. The experimental results of the component included a cluster size of 50 for each of the video footage, and the aggregated result of the people detector was clustered using K-Means using both the RESNET-18 and AlexNet deep-learning features. The results achieved 96% accuracy in each cluster for the 4-actors embedded within the content capture section for the validation of the component efficiency. Subsequent analysis will be carried out for the evaluation of the retrieval performance for each of the anchor as selected by the infrastructure operator.

Facial Recognition Component. The implementation of face recognition component within DEFENDER project consists of two modules namely (i) face detection and (ii) face recognition. The face detection mechanism begins with a new image representation named “Integral Image”. The integral image is computed on an image using some pixel based operations. This integral image enables fast evaluation of features and hence is used compute a set of features which are reminiscent of Haar Basis functions. However, the number of haar-like features computed on an image sub-window is very large, far larger than the number of pixels itself, and hence requires filtering to obtain only a small set of critical features. This is carried out using an Adaboost based learning process which excludes a majority of available features and narrows the feature-set down to include only the critical features. The process itself is a slight modification introduced to the AdaBoost procedure: Each step in the boosting process involves a weak classifier that depends only on a single feature, and this is modified to be used as a feature selection process. Next, the features are fed into a set of complex classifiers, which are combined in a cascade structure. This method of combination is said to enable an increase in speed by focusing attention on promising regions of the image. The face recognition system integrated in the DEFENDER detector is based on the specification of Facenet. Instead of training

a face recognition system in the form of a conventional classifier, FaceNet implements a system which directly maps the input face thumbnails to the compact Euclidean space. The Euclidean space is generated such that the L2 distance between all faces of the same identity is small, whereas the L2 distance between a pair of face images from different identities is large. This is enabled by triplet loss which, by definition, aims at minimizing the distance between pairs of same identity while maximizing the distance between pairs of different identities. For more details on the implementation of the face recognition, readers are referred to [31]. Additional details on the video analytics components integrated within the platform can be found in [1].

3.2 Temporal Smoothing

The computation of media streams captured from the detectors are often subjected to frame level processing as supported by the specification of the camera. One of the challenges in performing a frame level analysis results in the lack of ability to model the global situational awareness of the environment. To address this challenge, the proposed security framework implements a stream analytics solution with the use of latency processing that allows for buffering the input media stream for a period between 0.15 to 0.6s (for video streams captured at 60 fps and 25 fps respectively) prior to the deployment of the computational module. The stream analytics platform allows for the construction of global situational awareness through the consolidation of the media sources collected in the buffer. The initial latency period of the detector does not affect the performance of the detector, rather enhances the reliability and severity measure of the alerts generated.

3.3 Encrypted Media Repository

Following the implementation of GDPR across Europe, data privacy and protection has become an inherent necessity to adopt privacy-by-design methodologies for system implementation. Therefore, the project has adopted the use of encryption solution to process the media data captured from the detectors. An overview of the system adopted within the project has been presented in Fig. 3. The encryption is carried out using AES 256 Crypt specification². All the media data extracted from the computational units are encrypted using a pre-determined password as configured within the system deployment.

3.4 Privacy Preserving Technologies

In the literature, there are seven privacy preserving techniques reported in the literature including blur, pixelating, emboss, solid silhouette, skeleton, 3D avatar

² <https://www.aesencrypt.com/>.



Fig. 3. Encrypted media repository

and invisibility [21]. The new capabilities of such systems provide the computational tools the ability to collect and index a huge amount of private information about each individual, approaching the perimeter of the critical infrastructure. However, based on the privacy by design methodology adopted within the project, the framework transfers no personal data to be processed or made available to the command centre until a threat has been identified, which requires neutralisation. To this end, the use of privacy-by-design methodology incorporated within the media processing framework adopts the use of Gaussian blur to mask the identity of the person against the extraction of usable features. For the person re-identification component, the use of feature extraction module based on AlexNet protects the identity of the person without compromising the computational ability of the platform.

3.5 Knowledge Model

The knowledge model represents a set of high-level business rules that encodes the notion of abnormal behaviour at the perimeter of critical infrastructure based on temporal association of people detected using the surveillance infrastructure. The syntax adopted for the rule definition is based on JavaScript Object Notation (JSON) representation that systematically formalises the attributes of the detector, for anomaly detection. Each of the extracted person object from the perimeter is indexed against a unique identifier. The encrypted media repository creates a new index for every new person being detected. Internally, a large-index of temporal occurrence of the individual people are stored. The occurrence index consists of three categories of time stamps namely (i) past, (ii) current and (iii) ignored. For each of the new person being detected, a similarity metric is applied to associate the new person to existing index of person being treated. In addition, the high-level event representation syntax for event detection such as loitering, reconnaissance are also encoded in the JSON format based on the timestamps. The detector outcomes following the media processing are also exported in JSON format as specified in the knowledge base. The threat level severity

are pre-determined based on the proximity of the threat against breaching the perimeter of the critical infrastructure.

3.6 Threat Evaluator

The threat evaluator module receives input from privacy preserving technology output and the organisational guidelines on the threat models and severity. In addition, the module will also interact with the encrypted media repository to present the decrypted media data to the command centre upon the detection of the threat. The module evaluates the spatial constraints configured within the platform to determine the threat level. As an instance, the intruder detector has two levels of severity based on the proximity of threat to the critical infrastructure. The severity levels are appropriately identified and the evolution of the threat in time will be continuously monitored through alerts shared with the command centre. The spatial configuration of the infrastructure environment are coded into spatial coordinates as viewed by the sensor. The 2D image coordinates are internally mapped against the real-world distance measures. A visual description of the spatial mapping is presented in Fig. 5. For the determination of high-level threat analyser such as reconnaissance or loitering, specification of temporal rules have been defined within specific time windows to identify malicious perpetrators. These rules are quantified through a JSON syntax as outlined in Sect. 3.5. The anonymous identity labels assigned to the individual extracted from the detectors are used to visually cluster and enable correlation between the repeated appearance of the same individual near the vicinity of the critical infrastructure. The command centre provides a unified interface for the collection of media captured from distributed availability of the detectors. The web-interface allows for the easy navigation for the selection of different detector output that are spatio-temporal indexed.

3.7 Command Centre

The command centre is a central interface that interfaces with each of the detectors and integrates the different modules within the proposed framework. Upon the installation of the detector at the perimeter, the detector is configured with the command centre through the specification of IP address through which the detector will communicate with the command centre. The detector installation at the Erchie trial site for the intruder detection is presented in Fig. 4. The registration of the detector installation is carried out using a RESTful interface and JSON metadata consumed by the command centre. Subsequently, the evidence collected from the detector both raw data and the processed output are both transmitted to the threat evaluator module, which upon the determination of the data sources, decides to present the privacy protected results or the raw data based on the severity of the threat level.



Fig. 4. Detector installation for intruder detection

4 Experimental Results

The overall experimental results of the proposed security framework has been evaluated within the context of operational environments of the critical infrastructure assets being protected against external threats. The results summarised below are obtained from the two pilot trials carried out in DEFENDER namely Erchie and Okrogolo. While the human intruder detection has been extensively evaluated in the Erchie trial, the face recognition component has been evaluated as a part of the Okrogolo trial. The voluntary participation of the actors has been used to evaluate the system performance. The evaluation of the system performance included the computational latency required for the detector to send notifications to the command centre on the appearance of intruders and the evolution of the threats sequence in time. To facilitate the deployment of the countermeasures against the threat the alert sent to the command centre based on the event are separated by 20 s. Based on the evolution of the threat, from the proximity of the infrastructure to the approach of the perimeter, the status flag embedded within the alerts are changed from LOW, MEDIUM, HIGH and VERY HIGH. The experimental evaluation of the detector carried out in the DEFENDER field trail, yielded an accuracy of 96.7% in the person detection. The spatial depiction of the results obtained from the trail is presented in Fig. 5.

In order to evaluate the robustness of the framework, a continuous experimentation process was adopted in which the detector was operated for long periods of time for the detection of different threat levels. A summary of the evaluation results are presented in Table 1. The long-term durability of the physical security



Fig. 5. Intrusion detector

detectors were evaluated against the ability of the detector to identify the appropriate threat levels based on the critical infrastructure perimeter configuration. A total of 4 participants volunteered to take part in the trial for the evaluation of the detector. The infrastructure intrusion attack scenario was orchestrated with several approaches to the perimeter being considered. During the operation of the trial, a total of 267 events were identified resulting in a total of 350 threats. The report of additional 83 alerts were attributed to the dual detection of person intrusion due to mis-classification of non-human objects as intruders. For each of the intruder detected, an alert was generated and transmitted to the command centre.

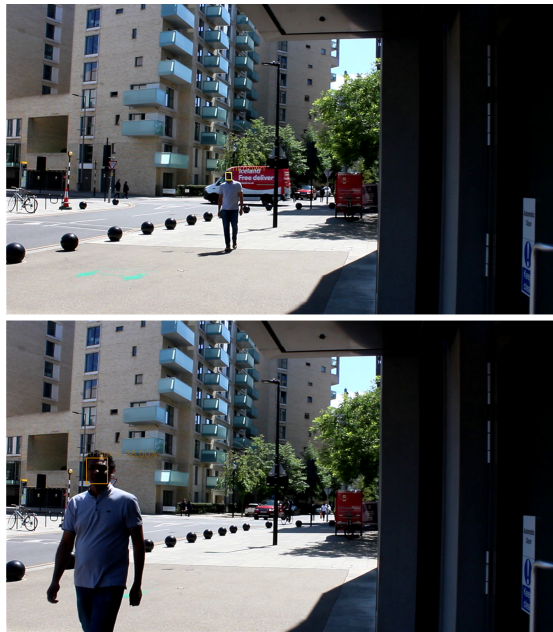
Table 1. Robustness evaluation of the proposed framework for intrusion detection

Parameter of evaluation	Metric
Total duration of the deployment	21,600 s
Total amount of video data processed	648,000 frames
Total number of participants in the trial	4
No. of incidents identified	267
No. of alerts generated	350
No. of repeated alerts generated with high severity	47
Computational latency of the component	25 m

Table 2. Robustness evaluation of the proposed framework for face recognition

Parameter of evaluation	Metric
Total number of participants in the trial	14
% accurate detections with single person in the frame	97.8
% detections with multiple persons in the frame	96.7
Computational latency of the component	35 m

The face recognition component has been evaluated against changing environmental conditions with a total of 40 participant (consisting of 14 nationalities and members from 9 ethnic background) features annotated within the database. The media data captured from the street level camera has been integrated with the module. In contrast to the scientific review of face recognition solutions, the evaluation metrics applied here includes the distance measurement at which faces can be detected and the distance at which the recognition takes place. The integrated face recognition component has shown to deliver reliable performance between 10 to 15 m distance between the detector and the subject annotated in the database. The use of MTCNN face detector delivers performance at 20 m and beyond depending on the availability of facial characteristics and features. The results of continuous evaluation of experimental results are summarised in Table 2. A total of 14 voluntary participants were evaluated within the context of operational environment. An overall accuracy of 97.8% were observed

**Fig. 6.** Face Recognition in operational environment for authentication

when a single person detection was carried out. Subsequently, the multi-person recognition upto 10 participants has yielded an average of 96.7% overall accuracy. The decreased efficiency of the algorithmic performance is attributed to the configuration of thresholds required to against the L2 norm of the algorithm output.

5 Conclusion and Future Work

The paper has summarised the integration of three computer vision technologies within DEFENDER project. The paper outlined the implementation of additional computational components to enhance the operational capacity of the laboratory validated solution. The real-world deployment of the solution has been extensively evaluated in Erchie and as a part of the Okrogolo trails. The security framework brings together several key innovations to deliver real-time operational insight to the infrastructure operators for deploying mitigating actions against perceived threats. The novelty of the proposed solution relies in the use of privacy-by-design methodology for protecting the identity and rights of citizen. The structured encoding of organisational policies for identifying threats are considered by the threat evaluator to deliver alerts to the command centre of the infrastructure operator. The future work will review the design of detectors and enhance the communication protocol for enabling bi-directional transfer of control and media signals between the command centre and the media detectors. In addition, the performance of the detector will be continuously reviewed and kept abreast with latest scientific innovations results reported in the literature. Finally, the structure of the knowledge model will adopt the use of ontology and semantic representation for encoding security specification of critical infrastructure.

Acknowledgement. The research activities leading to this publication has been partly funded by the European Union Horizon 2020 Research and Innovation program under MAGNETO RIA project (grant agreement No. 786629) and DEFENDER IA project (grant agreement No. 740898).

References

1. Arachchilage, S.W., Izquierdo, E.: A framework for real-time face-recognition. In: 2019 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4 (2019)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* (2008). <https://doi.org/10.1016/j.cviu.2007.09.014>
3. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Comput. Vis. Image Underst.* (2006). <https://doi.org/10.1016/j.cviu.2005.05.005>
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (2017). <https://doi.org/10.1109/CVPR.2017.143>

5. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: Proceedings - IEEE International Conference on Multimedia and Expo (2018). <https://doi.org/10.1109/ICME.2018.8486597>
6. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learn.* (1995). <https://doi.org/10.1023/A:1022627411411>
7. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. In: *Advances in Neural Information Processing Systems* (2016)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005* (2005). <https://doi.org/10.1109/CVPR.2005.177>
9. Ding, C., Tao, D.: A comprehensive survey on pose-invariant face recognition. *ACM Trans. Intell. Syst. Technol.* (2016). <https://doi.org/10.1145/2845089>
10. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. *Int. J. Robot. Res.* (2013). <https://doi.org/10.1177/0278364913491297>
11. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* (2015). <https://doi.org/10.1109/TPAMI.2015.2389824>
12. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report (2007)
13. JafriRabia, R., A.: *J. Inf. Process. Syst.* 5(2), 41–68
14. Jiang, Z., Huynh, D.Q.: Multiple pedestrian tracking from monocular videos in an interacting multiple model framework. *IEEE Trans. Image Process.* (2018). <https://doi.org/10.1109/TIP.2017.2779856>
15. Jiao, L., et al.: A survey of deep learning-based object detection. *CoRR* abs/1907.09408 (2019), <http://arxiv.org/abs/1907.09408>
16. Khan, A., Rinner, B., Cavallaro, A.: Cooperative robots to observe moving targets: Review. *IEEE Trans. Cybernet.* (2018). <https://doi.org/10.1109/TCYB.2016.2628161>
17. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *Proceedings of the IEEE International Conference on Computer Vision* (2015). <https://doi.org/10.1109/ICCV.2015.425>
18. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the IEEE International Conference on Computer Vision* (1999). <https://doi.org/10.1109/iccv.1999.790410>
19. Masi, I., Wu, Y., Hassner, T., Natarajan, P.: Deep face recognition: a survey. In: *Proceedings - 31st Conference on Graphics, Patterns and Images, SIBGRAPI 2018* (2019). <https://doi.org/10.1109/SIBGRAPI.2018.00067>
20. Ochoa-Villegas, M.A., Nolasco-Flores, J.A., Barron-Cano, O., Kakadiaris, I.A.: Addressing the illumination challenge in two-dimensional face recognition: a survey. *IET Comput. Vis.* 9(6), 978–992 (2015). <https://doi.org/10.1049/iet-cvi.2014.0086>
21. Padilla-López, J.R., Chaaoui, A.A., Flórez-Revuelta, F.: Visual privacy protection methods: a survey. *Expert Syst. Appl.* 42(9), 4177–4195 (2015). <https://doi.org/10.1016/j.eswa.2015.01.041>
22. Ristani, E., Tomasi, C.: Features for multi-target multi-camera tracking and re-identification. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018). <https://doi.org/10.1109/CVPR.2018.00632>

23. Scheenstra, A., Ruifrok, A., Veltkamp, R.C.: A survey of 3d face recognition methods. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) *Audio- and Video-Based Biometric Person Authentication*, pp. 891–899. Springer, Berlin Heidelberg, Berlin, Heidelberg (2005)
24. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: *Proceedings of the IEEE International Conference on Computer Vision* (2003). <https://doi.org/10.1109/iccv.2003.1238663>
25. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708 (2014)
26. Tian, Y., Luo, P., Wang, X., Tang, X.: Deep learning strong parts for pedestrian detection. In: *Proceedings of the IEEE International Conference on Computer Vision* (2015). <https://doi.org/10.1109/ICCV.2015.221>
27. Varior, R.R., Haloi, M., Wang, G.: Gated siamese convolutional neural network architecture for human re-identification. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2016). https://doi.org/10.1007/978-3-319-46484-8_48
28. Wang, M., Deng, W.: Deep face recognition: A survey. *CoRR* abs/1804.06655 (2018). <http://arxiv.org/abs/1804.06655>
29. Wang, X.: Intelligent multi-camera video surveillance: a review. *Pattern Recogn. Lett.* (2013). <https://doi.org/10.1016/j.patrec.2012.07.005>
30. Zhang, S., Benenson, R., Schiele, B.: Filtered channel features for pedestrian detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2015). <https://doi.org/10.1109/CVPR.2015.7298784>
31. Zhang, X., Chandramouli, K., Gabrijelcic, D., Zahariadis, T., Giunta, G.: Physical security detectors for critical infrastructures against new-age threat of drones and human intrusion. In: *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–4 (2020)
32. Zhang, X., Gao, Y.: Face recognition across pose: a review. *Pattern Recogn.* (2009). <https://doi.org/10.1016/j.patcog.2009.04.017>
33. Zhang, Z., Wu, J., Zhang, X., Zhang, C.: Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *CoRR* abs/1712.09531 (2017), <http://arxiv.org/abs/1712.09531>
34. Zheng, L., Huang, Y., Lu, H., Yang, Y.: Pose-invariant embedding for deep person re-identification. *IEEE Trans. Image Process.* (2019). <https://doi.org/10.1109/TIP.2019.2910414>



A Comprehensive Dataset from a Smart Grid Testbed for Machine Learning Based CPS Security Research

Chuadhry Mujeeb Ahmed^{1,2}(✉) and Nandha Kumar Kandasamy¹(✉)

¹ iTrust, Singapore University of Technology and Design, Singapore, Singapore
chuadhry@alumni.sutd.edu.sg, nandha001@e.ntu.edu.sg

² Computer and Information Sciences Department, University of Strathclyde,
Glasgow, Scotland

Abstract. Data-sets play a crucial role in advancing the research. However, getting access to real-world data becomes difficult when it comes to critical infrastructures and more so if that data is being acquired for security research. In this work, a comprehensive dataset from a real-world smart electric grid testbed is collected and shared with the research community. A few of the unique features of the dataset and testbed are highlighted.

1 Introduction

Recent progress in technology is resulting in the digitization of the physical world and things around us. It is expected that communication and computing capabilities will soon be part of all the physical objects [14]. The integration of cyber technologies (computing and communication) with the physical world gives rise to complex systems referred to as Cyber Physical Systems (CPS). CPS has changed the methods that humans used to interact with the physical world. Some examples of CPS are manufacturing, transportation, smart grid, water treatment, medical devices and the Industrial Control Systems (ICS) [17].

CPS is a broad term; in the following, we highlight a major sector applicable to our daily life, that is, the electrical power system as shown in Fig. 1. It shows the high-level architecture of an electrical power system. This is composed of electricity generation (power plants), transmission (electric grid system) and end-users (smart home). As one can imagine this power system is composed of a multitude of devices and physical processes. Power generation and transmission depend on the demand from the utilities and the users. To meet the requirements of the energy demand the critical infrastructure is utilized to ensure a continuous supply of power. Each of the processes in the critical infrastructure is a complex engineering system and needs a sophisticated control to achieve its desired objectives. For example, at the generation stage, we have generators, Intelligent Electronic Devices (IEDs) also incorporating electric relays, all these devices are autonomously controlled by the Programmable Logic Controllers (PLC). This

means that we have a lot of sensors monitoring the physical process, actuators/generators and the physical infrastructure that communicate the current physical states with each other and with the PLC. Such communication among smart devices, on one hand, provides flexibility in controlling the complex CPS but the vulnerabilities in the same technologies give rise to cyber attacks. In the following, a famous recent attack on a power grid is discussed.

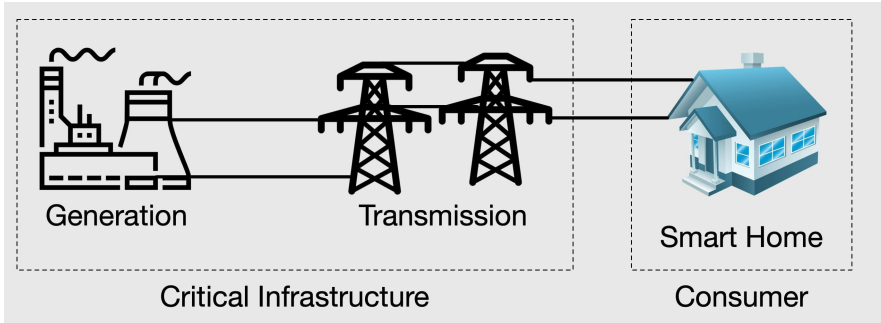


Fig. 1. A generic electrical power system as an example of CPS.

Ukrainian Electric Power Grid Attacks (2015,2016): In December 2015 cyber-attacks on Ukrainian electric power grid cut off the power supply to customers at the peak of the winter season. The attackers remotely controlled the SCADA distribution system and forced operators to switch to the manual mode which resulted in much longer recovery times [7]. This attack was over but for another attack to come in the next year around the same time. In 2016 again Ukrainian electric power grid met another cyber attack through the use of Crashoverride malware [18], This attack switched circuit breakers in an unusual open-close pattern in a fast manner, which resulted in cutting off the power supply to the customers. These attacks call for a pro-active approach towards the security of Critical Infrastructure (CI) like power grid.

Successful attacks on CI have led to a surge in the development of defense mechanisms to prevent, contain, and react to cyber attacks. One such defense mechanism is the anomaly detector that aims at raising an alert when the controlled process in a CI moves from its normal to an unexpected, i.e. *anomalous*, state. Approaches used in the design of such detectors fall into two broad categories: design-centric [1] and data-centric [5].

The use of machine learning to create anomaly detectors becomes attractive with the increasing availability of data and advanced computational resources. However, the data-based techniques rely on a rich dataset representing a real-world scenario. Such datasets are not easily accessible to academia. Our goal is to create a unique set of data that is 1) accessible and 2) represents real-world settings. In this article based on our experiments in a smart electric grid testbed, we have collected data for different states of the physical process under normal and attack conditions.

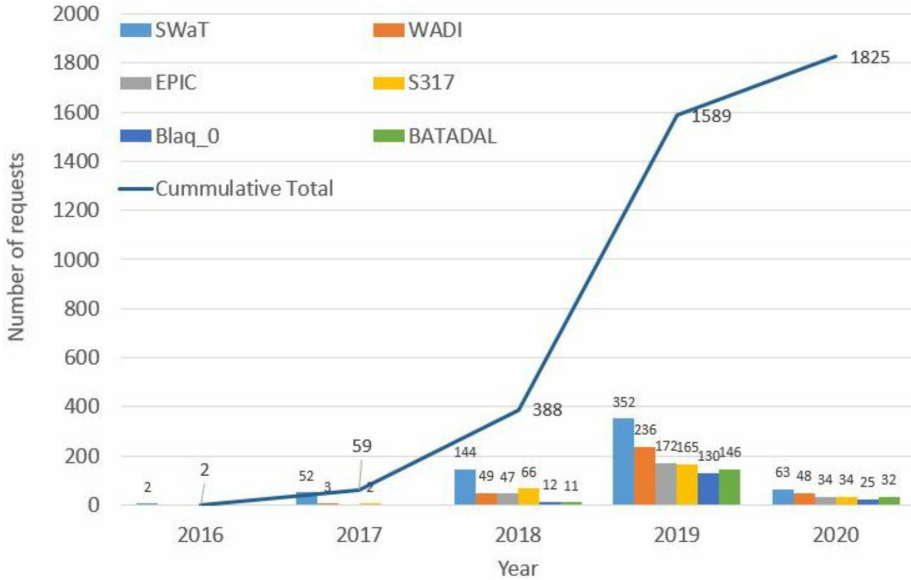


Fig. 2. Overview of dataset requests by year.

1.1 Motivation

EPIC: Electrical Power and Intelligent Control testbed are used in this study [2] that is a part of other testbeds at iTrust labs [10]. The most notable CPS datasets contributed by iTrust comes from water treatment (SWaT) [12] and water distribution (WADI) [4] testbed. The focus of this work is on the electric grid therefore, we have collected the data for EPIC testbed. To summarize the efforts carried out at iTrust labs on three testbeds Fig. 2 shows the requests for the dataset collected at these testbeds for each year. From Fig. 2 it can be observed that out of the three testbeds data for EPIC is being requested the least. We hypothesize that could be because there is not a comprehensive dataset for EPIC, unlike other testbeds that explain the data scenarios clearly.

There are some other efforts for data collection beyond iTrust but those still have few limitations. An interesting effort in electric grid testbed is simulation-based Softgrid testbed [9] but there is no data generation and sharing. For an ICS testbed in CPS [8] authors highlighted that their prototype lacked the collection and distribution of data as it involves a manual process requiring time and resources. [6] presented simulated IEC61850 traffic and no information regarding the real process and dynamics.

Previous research studies have tried to collect data from CPS settings but lack some desired features. We highlight a few of those,

- Simulated Data: Most of the datasets available are generated using simulated models [6]. Lack of real-world scenarios prompted us to do this work.

- Only Network Traffic Data: Previous data collection efforts only focused on the network traffic and as mentioned those were too simulated most of the time. One of the recent studies collected the network traffic from a realistic electric traction substation [13]. There is a lack of process data available from an electric grid based on realistic devices.

There have been efforts on building CPS testbeds for security research but very few were able to represent real-world scenarios and collect data to share with the academia and industry. We have focused on the process data from the sensors and actuators and tried to run the normal process for a range of different configurations. The idea is if we have enough real-world scenarios for the normal data, it is not hard to generate a malformed data/signal as other researchers have demonstrated in the past, e.g., using the mutation [16].

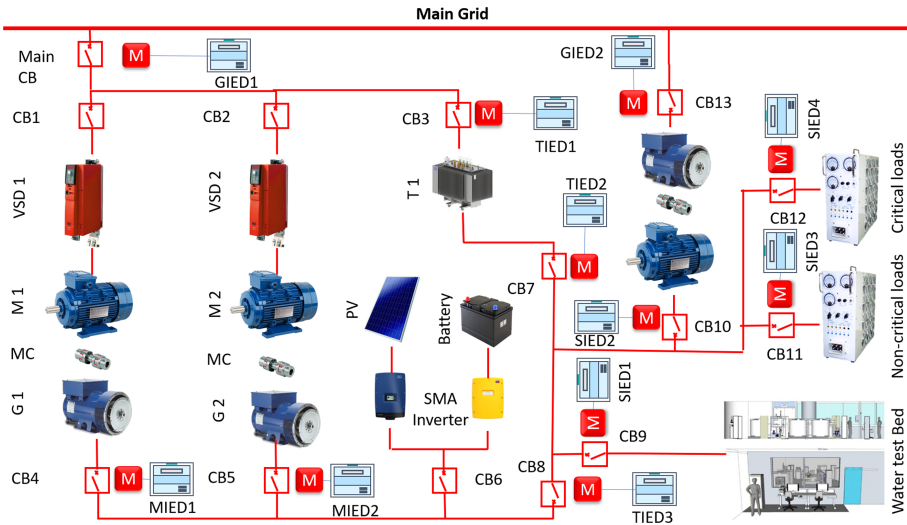


Fig. 3. Electrical layout of the testbed, electrical power lines are shown in red color lines. (Color figure online)

2 EPIC Testbed Introduction

EPIC mainly consists of four zones as described in the following. All the four zones are equipped with IEDs to collect Current, Voltage, Power and Frequency for the three phases buses.

- Generation: Generation stage is driven by electric motors connected to the main power supply.
- MicroGrid: Photo-voltaic cells, inverters and batteries compose this stage to supplement the generation of power.

- Transmission: This stage is composed of buses to transport power to the smart home unit.
- Smart Home: A programmable load bank containing RLC loads represents a home load environment. Besides, there are two water testbeds also connected to EPIC as the load.

2.1 Electrical Layout

The testbed consists of the following components as shown in Fig. 3. 1) Two conventional generators (10KVA each) run by 15kW VSD driven motors to represent the conventional combination of prime-mover and generator. 2) A 34kW PV system is available along with an 18kW battery system to represent power generation from the intermittent RES. 3) A 105kVA 3 phase voltage regulator for representing power supply from a transmission system. 4) Two load banks capable of emulating 45kVA load to represent critical and non-critical loads. It can supply power to the other two water testbeds. 5) A 10kW motor-generator load to represent spinning load. 6) Industrial standard Molded Case Circuit Breakers are used for short-circuit protection and switching functions.

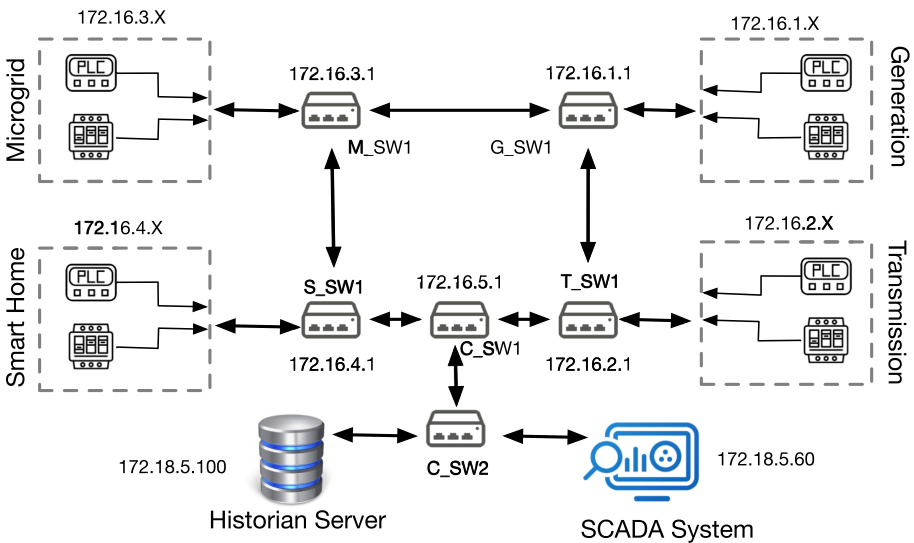


Fig. 4. A simplified network diagram for EPIC is shown. This is to help the associated network traces. IP addresses in the network traces correspond to respected devices shown in the figure. Each dotted box represents a subnet with the respective IP addresses. A X in the IP address means that a device in that subnet would have the similar subnet mask and then unique X as its own IP. Typical devices are PLCs and Intelligent Electronic Devices.

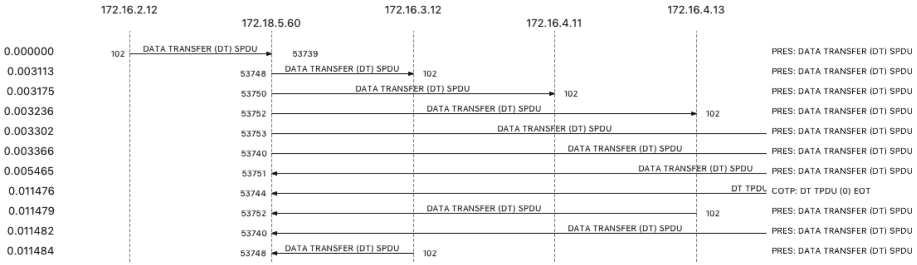


Fig. 5. A sample network traffic flow between different devices.

2.2 Communication Network

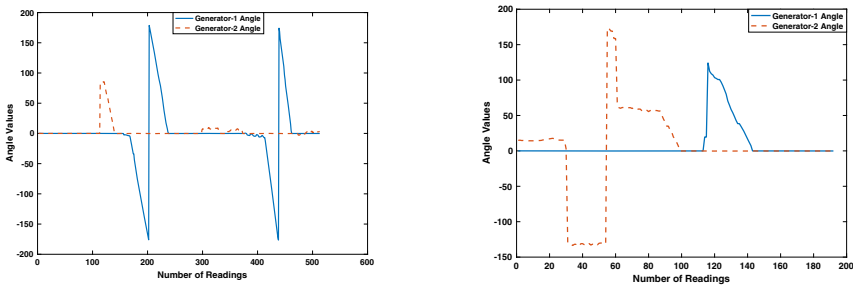
Figure 4 shows the communication network architecture in EPIC testbed. Figure 4 shows four major control zones, i.e., 1) Power Generation, 2) Transmission, 3) Microgrid, and 4) Smart Home. All of these four zones have IEDs and other devices to be controlled by dedicated PLCs. For example smart home also contains smart meters that can communicate with the PLCs and then can route the data through a central switch to the historian server and a SCADA workstation. Table 2 in Appendix A shows a map of IP addresses of the specific devices in the testbed. Network traffic is also collected at the SCADA station in a pcap format. Using Table 2 it is possible to make sense of the packet capture of the network traffic. For details on the communication protocols an interested reader is referred to the EPIC testbed papers [2, 15]. Figure 5 shows the sequence of communication between the different devices in the EPIC network. In the sequence diagram it can be seen that the SCADA workstation (IP: 172.18.5.60) holds a central position and send data transfer requests to the rest of the PLCs.

Table 1. Header Explanation in the Data set. IED:= Intelligent Electronic Device, GIED:= Generation stage IED, MIED:= Microgrid stage IED, SIED:= Smart home IED, TIED:= Transmission stage IED.

Data Header	Explanation	Devices
Measurement.Apparent	IED measures the apparent power	GIED, MIED, SIED, TIED
Measurement.Frequency	IED measures the signal frequency	GIED, MIED, SIED, TIED
Measurement.Line.Current	IED measures the line current	GIED, MIED, SIED, TIED
Measurement.V1	IED measures the voltage difference at L1	GIED, MIED, SIED, TIED
Measurement.Real	IED measures the Real power	GIED, MIED, SIED, TIED
Measurement.Reactive	IED measures the Reactive power	GIED, MIED, SIED, TIED
Measurement.Power.Factor	IED measures the power factor	GIED, MIED, SIED, TIED
Q.mode_close	mode of an electric breaker (T/F)	All Stages
VSD.property	properties can be .Current, .Speed, .Ready, .Fault	Generation stage to show the VSD status

3 Data Collection and Process Scenarios

The data is collected at the different configuration and operational settings of the EPIC testbed. Most of the physical process is driven by the demand from the smart home unit, constituted of different load types. These loads are used to simulate a real-world load requirement. Moreover, two other water plants are also connected to EPIC as the loads. The collected data can be collected from iTrust labs [10]. Network traffics is collected in pcap format and the process data is provided a Comma Separated Values (CSV) files. To make sense of the process data let us explain what each column in the data files means. Table 1 shows a list of data elements found in the dataset. The first column shows the quantity being measured and the last column shows the devices that are measuring that quantity. For example, the second row has Measurement.Apparent element meaning that the IED is measuring apparent power in different zones of the testbed. A column with header *GIED.Measurement.Apparent* means the apparent power is measured by an IED at the generation stage. In the following, we highlight the data collection scenarios and settings.



(a) Generator Synchronization without load. (b) Generator Synchronization with 10KW resistive load.

Fig. 6. Two different smart home load scenarios.

Scenario 1: Synchronization Process without Load. This scenario shows the process of synchronization for two generators without any load. Angle difference between two generators G1 and G2 changes in a cyclic manner from -180 to 0 to 180° until the synchronization is completed. Figure 6a shows this process twice.

Scenario 2: Synchronization Process with Load. To depict the synchronization for the two generators with a 10kW resistive load at the smart home stage. Angle difference between two generators G1 and G2 vary from -180 to 0° following a similar cyclic process as in Scenario 1. The additional load had little to no effect on the process. Figure 6b shows the process.

Attack on Synchronization Process. *Synchronization of two generators*, a new incoming generator i.e., the generator that needs to be connected in parallel to rest of generators in the grid needs to ensure the following three parameters,

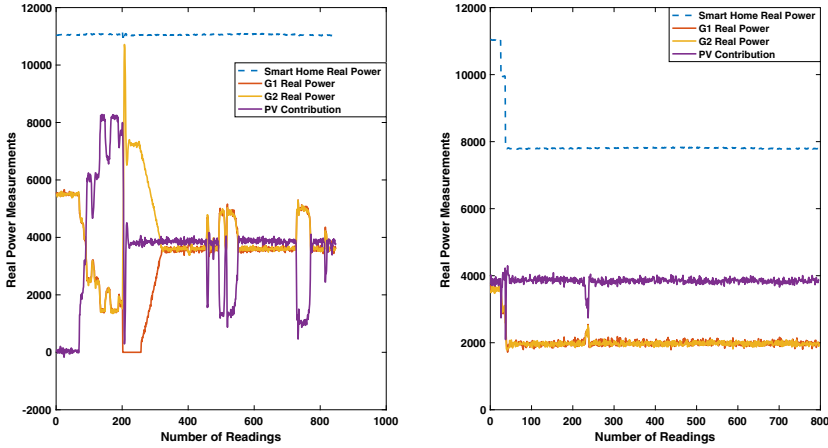


Fig. 7. Scenario-4 with 10 KW load and Scenario-5 with 7 KW resistive load.

- The frequency of the generator must be same as the frequency of the line/grid
- The magnitude of the generator’s voltage must be same as the magnitude of line/grid voltage
- The phase angle of the generator’s voltage must be same as the phase angle of the line/grid voltage (we will assume that the phase sequence is same)

The first two parameters do not depend on the state of the line/grid. However, the third parameters i.e., phase angle depends on the state of the phase angle of the line/grid. The parallel connection is enabled by circuit breaker which closes once the phase angle difference is approximately equal to zero (usually around 10° in practical cases). We launched the attack on the synchronisation process to delay it from the order of seconds to several minutes. The readers can refer [11] for a detailed analysis on the attack and plausible demand (Fig. 8).

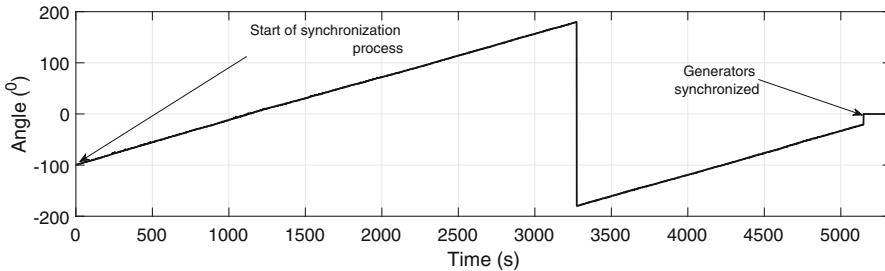


Fig. 8. Experimental results of synchronisation process of two generators after the launch of attack. Normal operation: Observe that ϕ_d changes from -180° to 180° at significantly slower pace, the breaker is closed when the generators are synchronized which took more than 1.5 h.

Scenario 3: Steady State Behavior. Here both the generators are running and synchronized. The resistive load is set to 10kW at the smart home stage. This scenario depicts a steady state behavior of the process. Any power plant, substation or section of power system will be in this stage for most of the time.

Scenario 4: PV System with Generators. This scenario depicts a situation where the user demand is being fulfilled using the two generators G1 and G2 and also using PV system With 10kW resistive load. The scenario depicts the operation of modern power system sections with renewable energy penetration. Figure 7 shows the process behavior for the scenario 4 and 5 respectively.

Scenario 5: PV System with Generators. This scenario depicts a situation where the user demand is being fulfilled using the two generators G1 and G2 and also using PV system With a reduced load of 7kW resistive load. With decreasing load the chances of creating power supply interruption attacks are relatively high, such scenarios are presented in [3].

Scenario 6: Three Generators Running. In this case all the available three generators G1, G2 and G3 are running with a load of 14kW resistive load at the smart home stage. This scenario depicts a system where there are motor loads such as buildings with Heat Ventilation and Air-conditioning (HVAC) systems.

Malicious Power Generation Attack - An Use Case. In this case, we manipulated the power generated from one generator to overload it in comparison to the other generator, so that the maintenance schedule can be offset as the overloaded generator needs more frequent maintenance due to additional wear and tear. This eventually leads to accumulated damage in the long run, as the overloaded generator was not serviced at the appropriate times due to malicious operation. During normal operation, to supply power to the critical loads, generators G1 and G2 will share the power equally. The SPLC has the control code that issues a subsequent command to the VSDs to run at a specific speed (1500RPM in this case), for enabling equal power-sharing among the two generators. The apparent power is equally shared between the generators. The time-domain representation of power-sharing before the attack was launched is shown in Fig. 9 and marked as normal.

After the attack was launched on generator G1, i.e, the speed of the prime mover of G2 reduced by 0.2RPM when generator G1 is supplying more power and hence disabling the power-sharing process. This attack scenario is marked in Fig. 9 where it can be observed that whenever G2 is supplying more power than G1, G1 takes over until equal power is shared among the two. However, when G1 is supplying more power, G2 fails to take over even after synchronization.

This resulted in G1 supplying more power under scenarios where G2 is synchronized as the second generator. The above condition will eventually result in tripping of G1 due to prolonged overload condition.

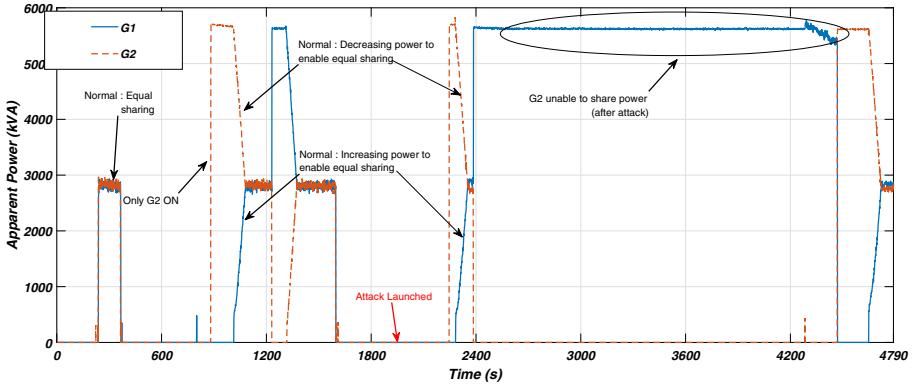


Fig. 9. Apparent power during load sharing between generators G1 and G2. When G2 is unable to share the power, the vulnerability of G1 to trip under overload, i.e., during peak load conditions, increases dramatically. The time-domain representation of power-sharing before the attack was launched is shown and marked as normal. Observe that before attack launch at around 1900s, G1 and G2 shared power irrespective of the condition, whereas after the attack launch, G2 was unable to share the power, i.e., between 2400 and 4300s.

Scenario 7: Cascading Effects. Two generators G1 and G2 running and the load is a real world application. In this case the power is being supplied to the Secure Water Treatment (SWaT) testbed. This scenario can be interesting to see the cascading effects of attack on one CPS system to another. For example, a power supply interruption attack [3] can affect the treatment process or the distribution process of water supply system.

Scenario 8: Cascading Effects. Two generators G1 and G2 running and the load is two CPS testbeds. In this case the power is being supplied to the Secure Water Treatment (SWaT) and water distribution (WADI) testbed. This scenario can be interesting to see the cascading effects of attack on one CPS system to another.

4 Discussion

It is not possible to collect data for a large number of scenarios and exhaust all the possibilities. We have done it for a range of representative scenarios and then we can use tools like mutation to generate more scenarios for both the normal and abnormal scenarios [16]. However, we tried to present a realistic data from both the process as well as network traffic perspective.

4.1 Normal Data for Real-World Process

We have presented the collected data for a range of normal process states from a real world smart electric grid network. It has a very high importance since the quantities measured are across the real process and devices giving an idea of what to expect in real world and come up with the realistic anomaly detection techniques.

4.2 Lack of Attack Scenarios on the Process

In this work we have presented the attack scenarios that could be executed without damaging the plant. The attacks those cause physical damage could not be executed on the plant as those pose dangers to the infrastructure and people around it.

5 Potential Use Cases and Future Work

Potential use cases extend from pure network attacks to attacks targeting a particular physical process. For instance, the attacker can explore the network traffic during the circuit breaker closing operation; which can be identified from the process data. Using the data the attacks with physical goal similar to [18] can be re-created. Following which appropriate defense mechanisms could be designed. Similarly, the data on the network traffic and the process status could be coupled for spoofing attacks that can hide the above attack from the operators.

From the defense front, the authors are working on a defense based on negative selection algorithm using dataset for defense against malicious power generation attacks. The dataset is also useful for developing and testing design-centric defense such as invariants based methods and authors are currently working on one such system.

A Supporting Data

Table 2. IP addresses for key devices in the network.

IP Address	Device
172.16.1.41	Generation stage PLC
172.16.2.41	Transmission stage PLC
172.16.3.41	Microgrid stage PLC
172.16.4.41	Smart Home stage PLC
172.16.5.41	Control stage PLC
172.18.5.60	SACDA System
172.18.5.100	Historian Server
172.16.2.11	Transmission $TIED_1$
172.16.2.12	Transmission $TIED_2$
172.16.2.13	Transmission $TIED_4$
172.16.3.11	Microgrid $MIED_1$
172.16.3.12	Microgrid $MIED_2$
172.16.4.11	Smart Home $SIED_1$
172.16.4.12	Smart Home $SIED_2$
172.16.4.13	Smart Home $SIED_3$
172.16.4.14	Smart Home $SIED_4$
172.16.1.11	Generation $GIED_1$
172.16.1.12	Generation $GIED_2$

References






1. Adepu, S., Mathur, A.: Distributed attack detection in a water treatment plant: method and case study. *IEEE Trans. Depend. Secure Comput.* 1–8 (2018)
2. Adepu, S., Kandasamy, N.K., Mathur, A.: EPIC: an electric power testbed for research and training in cyber physical systems security. In: Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinouidakis, C., Antón, A., Gritzalis, S., Mylopoulos, J., Kalloniatis, C. (eds.) *SECPRE/CyberICPS -2018*. LNCS, vol. 11387, pp. 37–52. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12786-2_3
3. Adepu, S., Kandasamy, N.K., Zhou, J., Mathur, A.: Attacks on smart grid: power supply interruption and malicious power generation. *Int. J. Inf. Secur.* **19**(2), 189–211 (2019). <https://doi.org/10.1007/s10207-019-00452-z>
4. Ahmed, C.M., Palleti, V.R., Mathur, A.P.: WADI: a water distribution testbed for research in the design of secure cyber physical systems. In: *CPS Week. CySWATER 2017*, pp. 25–28. ACM (2017)
5. Ahmed, C.M., Zhou, J., Mathur, A.P.: Noise matters: using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In: *Proceedings of the 34th Annual Computer Security Applications Conference, ACSAC 2018, San Juan, PR, USA, December 03–07, 2018*, pp. 566–581 (2018)

6. Biswas, P.P., Tan, H.C., Zhu, Q., Li, Y., Mashima, D., Chen, B.: A synthesized dataset for cybersecurity study of IEC 61850 based substation. In: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1–7 (2019)
7. Case, D.U.: Analysis of the cyber attack on the ukrainian power grid. Report (2016)
8. Green, B., Lee, A., Antrobus, R., Roedig, U., Hutchison, D., Rashid, A.: Pains, gains and plcs: Ten lessons from building an industrial control systems testbed for security research. In: 10th USENIX Workshop on Cyber Security Experimentation and Test (CSET 17). USENIX Association, Vancouver, BC, August 2017. <https://www.usenix.org/conference/cset17/workshop-program/presentation/green>
9. Gunathilaka, P., Mashima, D., Chen, B.: Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions. In: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. p. 113–124. CPS-SPC '16, Association for Computing Machinery, New York, NY, USA (2016). DOI: 10.1145/2994487.2994494, <https://doi.org/10.1145/2994487.2994494>
10. iTrust: iTrust Datasets. https://itrust.sutd.edu.sg/itrust-labs_datasets/
11. Kandasamy, N.K.: An investigation on feasibility and security for cyber attacks on generator synchronization process. *IEEE Trans. Ind. Inf.* (2019)
12. Mathur, A.P., Tippenhauer, N.O.: Swat: a water treatment testbed for research and training on ics security. In: *CySWater*, pp. 31–36, April 2016
13. Perales Gmez, L., et al.: On the generation of anomaly detection datasets in industrial control systems. *IEEE Access* **7**, 177460–177473 (2019)
14. Rajkumar, R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Design Automation Conference, pp. 731–736, June 2010
15. Siddiqi, A., Tippenhauer, N.O., Mashima, D., Chen, B.: On practical threat scenario testing in an electric power ICS testbed. In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, pp. 15–21 (2018)
16. Sugumar, G., Mathur, A.: A method for testing distributed anomaly detectors. *Int. J. Crit. Infrastr. Protect.* **27**, 100324 (2019). <https://doi.org/10.1016/j.ijcip.2019.100324>, <http://www.sciencedirect.com/science/article/pii/S1874548219301210>
17. Sutton, F.: An Efficient Platform and Communication Architecture for Event-triggered Cyber-physical Systems. Ph.D. thesis, ETH Zurich (2018). <https://doi.org/10.3929/ethz-b-000260384>
18. US-CERT: Crashoverride malware. US-CERT Report (2017). <https://www.us-cert.gov/ncas/alerts/TA17-163A>

Security Management and Governance



Cross-Domain Security Asset Management for Healthcare

Federico Stirano¹, Francesco Lubrano¹, Giacomo Vitali¹,
Fabrizio Bertone¹, Giuseppe Varavallo¹, and Paolo Petrucci²

¹ LINKS Foundation, via Boggio 61, Turin, Italy
{federico.stirano, francesco.lubrano, giacomo.vitali,
fabrizio.bertone, giuseppe.varavallo}@linksfoundation.com

² ASL TO5, Moncalieri, Italy
petrucci.paolo@aslto5.piemonte.it

Abstract. Healthcare is one of the most peculiar between all Critical Infrastructures due to its context and role in the society. The characteristics of openness and pervasive usage of IT systems and connected devices make it particularly exposed to both physical threats, such as theft and unauthorized access to restricted areas, and cyber attacks, like the notorious wannacy ransomware that abruptly disrupted the British National Health System in May 2017. Even the recent COVID-19 pandemic period has been negatively characterized by an increase of both physical and cyber incidents that specifically targeted hospitals and undermined an essential public service like healthcare. Effective security solutions are necessary in order to protect and enhance the resiliency of the Critical Infrastructures. This paper presents the work being developed in the context of the SAFECARE H2020 project, that specifically considers the requirements for security of hospitals. A particular focus is given to the asset management that consider cross-domain aspects of security, like the physical location and virtual connections that link different components of a hospital. This allows advanced knowledge that enables to infer and forewarn of possible elaborated cyber-physical *kill chains*. This is particularly important and useful during crisis, as allows to have a holistic overview of the status of the hospital and the potential impacts of one or more incidents to the critical assets. The description and simulation of an attack scenario is also given, together with the description of the messages exchanged by the security systems and the information made available to security operators.

Keywords: Healthcare · Critical infrastructure · Cross-domain security · Asset availability

This research received funding from the European Union’s H2020 Research and Innovation Action “Secure societies – Protecting freedom and security of Europe and its citizens” challenge, under grant agreement 787002 (project SAFECARE).

© The Author(s) 2021

H. Abie et al. (Eds.): CPS4CIP 2020, LNCS 12618, pp. 139–154, 2021.

https://doi.org/10.1007/978-3-030-69781-5_10

1 Introduction

Hospitals, power supplies, water supplies, telecommunications and transports are just few examples of Critical Infrastructures (CIs). CIs provide vital functions to modern societies, and the protection of such infrastructures is a key issue at European level. Thus, the creation of systems able to provide threat prevention, detection, response, and, in case of failure, mitigation of impacts across infrastructures, populations, and environment, is now a primary concern.

Hospitals, and in particular typical European hospitals, are peculiar CIs, because of their context and role. Hospitals, in several cases, are placed in urban environments and sometimes in historical buildings. They are characterized by a considerable influx of various people (patients, visitors, medical staff, administrative staff, technicians) and the simultaneous presence of different organizations, such as universities.

Unlike public offices or other public buildings, hospitals have limited possibilities to restrict the access in one *single point of entry* and to create checkpoints and visitors control desks to check or store identity documents. Therefore, it is very challenging to manage and control the access to the hospitals public areas, making them particularly vulnerable to physical threats.

Besides people, that are the most critical asset, healthcare infrastructures are also characterized by the presence of a huge number of different technical equipment. Cryogenic system, RX systems, radioactive isotopes, big magnetic field systems, gases tanks, hyperbaric systems, picture archiving and communication system (PACS), laboratory information systems (LIS), are a small representation of the wide variety of critical assets that can be found inside hospital facilities.

Such components, as many others present in healthcare infrastructures, are becoming increasingly integrated and connected. Indeed, nowadays hospitals deeply rely on the IT network and IT systems. For example, the PACS and the LIS systems manage and exchange data through the hospital IT network. These systems allow the possibility to work with radiological images and laboratory tests. Attacks to such systems or the internal network of the hospital, could create several issues or seriously slow down the diagnosis process or patient treatment.

For example, a set of 19 vulnerabilities has been recently disclosed under the name of Ripple20 [1]. These vulnerabilities could allow the remote execution of code on IoT devices and embedded systems, posing a serious risk for patient's health. It has been reported that the most affected sector is healthcare and more than 50000 medical devices such as infusion pumps have already been identified to be vulnerable [2].

Cyber-security solutions, implemented to protect these critical assets, provide a reasonable level of security, preventing such types of attacks and increasing the difficulty of reaching such systems from outside of the network.

While generally most of the cases of physical incidents are limited in scope, a physical intrusion like an access with tailgating or burglary or otherwise fraudulent could be preparatory for a following cyber attack, for example to exploit the mentioned vulnerabilities.

This combination of physical intrusion and the consequent cyberattack constitutes an example of a new category of threats that includes more complex but effective attacks. Thus, the border between physical and cyber domains is increasingly blurred. From the security point of view, an integrated approach that takes into account physical and cyber threats is then required.

During the recent COVID-19 pandemic, an increase of both cyber attacks (malware) and physical incidents (theft) has been observed. At the time of writing, SAFECARE tracked more than 150 incident reported in news from all over the world [3]. This is due to the opportunistic behaviour of criminals, that will try to take advantage of critical situations to gain profit.

This paper describes the integrated cyber-physical security solution being implemented in the context of the SAFECARE H2020 project¹. A special focus is dedicated to the Hospital Availability Management System (HAMS), a sub-module of SAFECARE that provides a global overview of the hospital status and is integrated with the SAFECARE incident detection and impact evaluation functionalities. With its interfaces the HAMS shows what are the occurred incidents, what are their impacts and provides assets availability, updated when incidents occur.

2 Related Work

Asset management, resources availability and bed occupancy are very important aspects to handle during the day-to-day operations of an hospital. This is even more critical during emergency situations and events like natural disaster, terrorist act or other hazards.

One of the most important example of software that is specifically designed for emergencies is SAHANA Disaster Management System (DMS) [4,5]. SAHANA DMS was adopted in 2010 during the earthquake emergency in Haiti, in the city of Port-au-Prince. This system helped manage the flow of victims in Haiti, sharing data about hospital assets (department service, bed availability, staff availability) with emergency administrators.

Health Resources Availability Mapping System (HeRAMS) [6,7] developed by the WHO and Global Health Cluster, is another important example. Its objective is to estimate the availability of services and resources in the hospitals placed in regions in crisis or health emergency. HeRAMS is based on questionnaires carried out in health care infrastructure to collect data concerning the availability of health resources and services such as health care personnel, beds, medical devices, and medicines. The questionnaires' outcomes are published in an interactive dashboard to reflect the situation of health care resources.

The analysis of these systems showed that the data related to the availability of assets is updated manually (Sahana DMS); in the second case, the data of assets is updated using the results of questionnaires (HeRAMS).

¹ <https://www.safecare-project.eu/>.

More in general, a multitude of different solutions that handles various aspects of asset management (e.g. tracking of non-fixed assets, maintenance, inventory, bedding, software licensing, etc.) exists.

While the main objective of those systems is to enhance the efficiency of the hospital in normal conditions, the asset availability management solution described in this paper is specifically conceived to take into account the cyber and physical incidents detected by the platform and accordingly update the availability of impacted or potentially impacted assets.

3 SAFECARE Architecture

The EU-funded SAFECARE project aims at developing an integrated solution to the cyber and physical threats which can affect healthcare infrastructures [8]. This is done through the implementation and deployment of multiple systems that recognize, correlate and analyze those threats and, when incidents occur, infer the impacts and start responses and damage mitigation mechanisms [9]. This will lead to an improvement in the security, protection and resilience of such services, allowing for a better response in case of external attacks specially during health related emergencies such as the current COVID-19 pandemic.

The SAFECARE architecture is composed of several modules, each with specific role, in this paper we will focus just on a subset of them that can be classified into two major functions:

- *Cyber and physical incident detection systems*
- *A centralised structure capable of storing and combining incoming data and of evaluating potential impacts when security incidents occur*

A simplified schema of the global structure is represented in Fig. 1. The connection between the different components is ensured by the Data Exchange Layer (DXL), which allows all the modules to communicate with each other in real time and provide relevant interfaces to exchange data.

3.1 Threat Detection Systems

The Threat Detection Systems consists of physical and cyber security solutions implemented by smart modules and integrated technologies.

More specifically, physical security solutions embed intelligent video monitoring and interconnect building monitoring systems as well as management systems. The Building Threat Monitoring System (BTMS) module is the aggregation point for physical security and makes the link between physical systems and the rest of the architecture.

In the same way, the Cyber Threat Monitoring System (CTMS) connects the cyber systems to the rest of the architecture and is the central point for cyber security solutions, which consist of threat detection systems related to Information Technology (IT) and e-health systems.

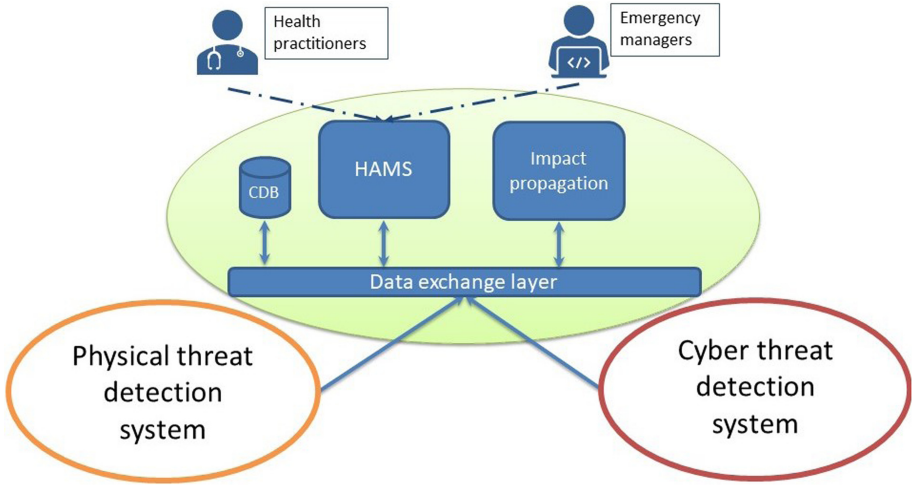


Fig. 1. SAFECARE global architecture.

Both physical and cyber security monitoring tools are interconnected thanks to the cyber-physical security solutions. These consist of intelligent modules whose role is to integrate different data sources and better take into account the combination of physical and cyber security threats.

3.2 Central Database and Decision Modules

In order to cope with such a combined approach for the management of cyber and physical security, the SAFECARE project implements a Central Database (CDB): a single, unique repository that stores multiple types of data needed by the other modules in the platform.

In particular, it stores all the information related to assets, facilities, buildings, services inside the hospital, the relations among the assets (static data) and also the information that is generated by SAFECARE modules, such as incidents, impacts, the relation between them and all the other responses/messages (dynamic data).

The modules responsible for inferring cascading impacts of physical and/or cyber security incidents and providing information about health services availability are the Impact Propagation and Decision Support Model (IPDSM) [10] and the Hospital Availability Management System (HAMS) [11], respectively. In particular, the IPDSM’s role is to combine physical and cyber incidents that occur on assets, infer cascading effects as impacts that could potentially affect the same or related assets and, finally, alert other modules about the potential impacts and severity. The HAMS, on the other hand, elaborates the messages coming from the other modules to provide asset availability information.

3.3 Incidents and Impacts Messages

The different modules communicate to each other by means of the MQTT protocol. In SAFECARE architecture, the broker is located in the DXL, while each module implements its own client. Messages are exchanged through a publish/-subscribe mechanism, where clients publish on a specific topic and receive messages only on the topics they have previously subscribed over. Messages are encapsulated in the JSON format. While there are different kind of messages exchanged between SAFECARE modules, in this paper we focus on messages related to incidents and impacts.

An incident message is generated after a validation process carried out by a security operator, in order to verify that an event detected by the monitoring modules can be classified as an actual incident. The main fields contained in the incident message are: *incident identifier*, a unique reference in the whole system to a specific incident; *category*, the incident category that helps the analysis of impacts and availability; *severity*, that indicates how severe the incident is (the possible values are: *LOW*, *MINOR*, *MAJOR*, *CRITICAL*); *events*, a list of the events that led to the incident. Each *event* includes the list of the assets involved that may change their availability status, according to the procedure described in Sect. 4.

The impact message is composed by a field with the *incident identifier* of the related incident and a list of *assets* that are not directly involved by the incident but that may be impacted by its propagation. In fact, assets are associated to a set of parameters that express the likelihood that the asset can have consequences, according to the analysis performed by the IPDSM. One of these parameters is the *ImpactScore*, which gives a forecast of how much the asset is indirectly impacted by the related incident, 1 being totally unavailable and 0 not impacted at all, while intermediate values represent different degrees of impact.

Section 5.2 will provide an example of the incident and the impact messages generated by the sample scenario described.

4 Hospital Availability Management System (HAMS)

This section describes the behavior of the Hospital Asset Availability Management system in response to events detected by the security systems.

4.1 Evaluation of Impact and Asset Availability

When the HAMS receives an incident or an impact message, it must evaluate the content of message to verify whether the availability status of an asset needs to be modified or not. Asset availability status is represented with three parameters:

1. A Boolean value, stating if the asset is available or not.
2. A colour code (green, yellow, or red), acting like a traffic light. If the colour code is green, the asset is working normally. If the colour code is red, the asset has been heavily impacted by the incident and it is not available. If the

colour code is yellow, the asset has been impacted by the incident, but it is still partially working and available, for example locally but not remotely.

3. A stability value (stable, improving, deteriorating), providing a simple description of the dynamic of status variations.

Upon arrival of an incident message, the system checks the incident category and couples it with the category of each related event, as not all the categories can provoke a change of availability in the assets. For each asset whose status needs to be changed, the severity parameter is checked to define if the new status colour for each asset will be *yellow* (in the case of *LOW* or *MINOR* severity) or *red* (in the case of *CRITICAL* or *MAJOR* severity). Any other variation in the availability of assets is computed after the reception of an impact message.

When an impact message is received, the *ImpactScore* value is checked to decide whether an asset availability status value needs to be updated or not. Using a threshold approach, an asset can be declared unavailable (with colour red if $ImpactScore \geq 0.8$), partially available (available with colour yellow if $0.5 \leq ImpactScore < 0.8$) or fully available (with colour code green if $ImpactScore < 0.5$) by the HAMS. The thresholds could be changed or customized depending on the asset category or the different requirements of each hospital.

4.2 Asset-Centric Graph Visualization

The HAMS is implemented as a web application. Besides the server side, that connects this software with the SAFECARE system and implement all the logic to retrieve information from the database and receive all the useful messages, the HAMS implements a Graphic User Interface (GUI).

End-users involved in SAFECARE project provided their feedback about how to visualise asset affected by incident and taking into account the implicit hierarchical dependencies of assets in an hospital, an asset-centric graph visualisation has been developed.

Figure 2 reports an example of this visualisation for a fictional hospital. Starting from a facility level, the graph provides information about the status of services and operations. In this terminology, services includes hospital departments and a list of medical devices belonging to each service. On the other side, there are operations, that include all the systems that are not related to the medical activities but that are essential for the correct operation of the hospital. Each node of the graph reports the status of the asset and in case an end node (an asset) is involved in an incident, user can visualise details about the incident or the incidents that caused the changes in the asset status.

4.3 Interoperable Information Exchange

During the management of an emergency in the hospital sector, one of the main requirements is the capability to exchange the information in a timely manner, with a common data format for all the actors involved (hospitals, first responders,

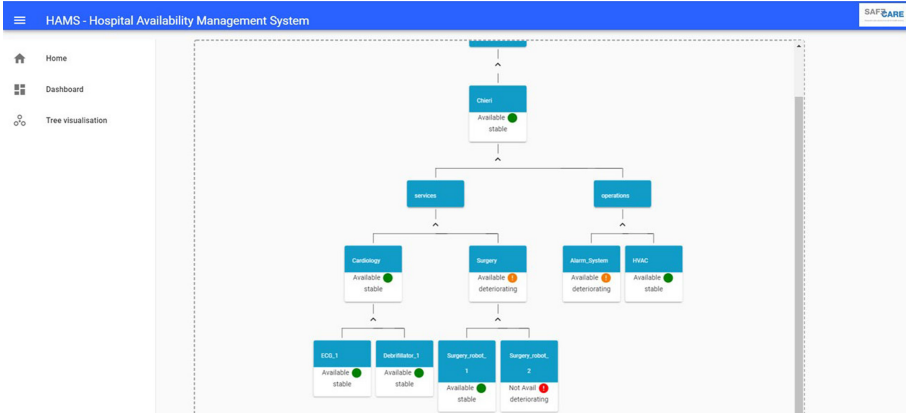


Fig. 2. HAMS graph-based visualisation (Color figure online)

etc.). HAMS deeply relies on the EDXL-HAVE [12, 13] standard to represent data internally and to share them with other systems.

EDXL [14] is a set of standards released by OASIS to manage the entire emergency life cycle and provide a common framework to exchange and share information between different emergency systems. In particular, EDXL-HAVE (HAVE) is an XML messaging standard developed by OASIS in the context of emergency management.

A HAVE schema consists of a root element that uniquely identifies the organization that is responsible for the reporting facilities. Each facility is described through several attributes and a list of sub-elements that allow a complete description of hospital departments, services, and resources.

As the HAMS data model is based on the HAVE standard, the updated availability information can be easily exported in a XML message based on the HAVE schema. In this way, the output file can be exchanged with other hospitals or emergency management actors to better coordinate the operations in case of necessity.

5 Sample Scenario

In order to design the modules and to test the developed solution, it is necessary to define plausible scenarios that combine physical and cyber attacks. Health organizations have a large variety of assets, which are essential for their operations. Therefore, it is necessary to firstly classify critical assets to have a global knowledge of the context. In the SAFECARE project, the following categories of assets that compose Healthcare organizations were identified:

- *Specialist personnel assets, (e.g., employees);*
- *Buildings and facilities, (e.g., power and gas supply, PLC room);*
- *Identification system, (e.g., badges, digital credentials);*

- *Networked medical devices, (e.g., medical devices connected to the internal data network);*
- *Networking equipment, (e.g., laptop, internet network);*
- *Interconnected Clinical Information Systems, (e.g., HIS, PACS);*
- *Mobile client devices, (e.g., mobile applications);*
- *Remote care system assets, (e.g., medical equipment for tele-monitoring);*
- *Data and records, (e.g., health records);*
- *Operating resources, (e.g. sterilization material).*

5.1 Methodology

In the SAFECARE project the scenarios are divided into strategic and technical [8]. The strategic scenario describes key variables to recognize the *risk source, opponent/objective, stakeholder/intermediate events*, and *impact/severity*. The technical scenario describes a process of how an attacker can take control of some assets and cause incidents in health care organizations. The model for developing technical scenario consists of four phases:

1. *Know*: discover the vulnerabilities of health care organizations (ecosystem mapping, information on key people, and systems). These activities are directed by the attacker to prepare for his assault and increase his probabilities of success.
2. *Get in*: actions are taken by the attacker to introduce himself in the planned goal, either in cyber or physical assets (Hospital information System, Department rooms, PLC room, etc.).
3. *Find*: during this phase, once the attacker has access to health care assets physically or digitally, he identifies where are locate the desired material or data.
4. *Control*: In the last stage, the attacker takes control of the assets or resources (e.g., material, data). The consequence is sabotage, loss of patients' data, ransomware.

In the SAFECARE project, 12 different Cyber-Physical technical scenarios were developed, involving the assets described. In the following of this paper we will consider a simplified scenario to show the interactions between the different modules of the platform.

5.2 Scenario Description

Based on one of the incident scenario developed in SAFECARE, a sample scenario has been defined in order to test and show the functionalities of the system. In this scenario, an attacker gets the control of a medical device connected to the internal network in order to expose patients' medical data. The main steps followed by the attacker are:

1. Obtain information of the medical device. Before going to the hospital, the attacker gets information on the medical devices, for example using Open Source INTelligence (OSINT) techniques, in order to identify their possible vulnerabilities.

2. Obtain local access to the medical device. The attacker goes to the area where the identified device is located.
3. Installation of malware on the medical device. The attacker install a malware software on the medical device through, for example, the connection of an USB flash drive. The malware propagates along the local network, in order to infect also the server with the medical records of the hospital patients, the final objective of the attack.
4. Exposition of medical data. When the malware infects the server with patients' health record data, the attacker can access the data and expose them, with severe consequences on the reputation of the hospital as well as the possibility to modify them, putting in danger the patients' life.

This scenario is a good example of integration between physical and cyber attacks, as in order to install the malware, the attacker must get the physical access to the area where the device is located and then wait until nobody else is present. BTMS is able to detect this suspicious loitering behaviour by analyzing live video streams and generates an event. Afterwards, two cyber anomalies are detected by CTMS: the installation of malicious software on the device and the infection of the server that contains patients' data, generating two security events that are grouped by the operator in an cyber incident.

5.3 Scenario Simulation

In order to perform the simulation of a scenario, a sample hospital has been defined, and represented in Fig. 3.

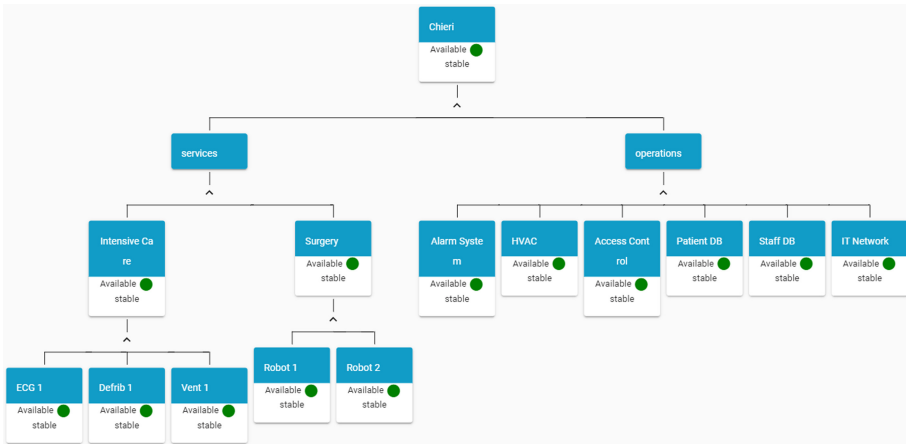


Fig. 3. HAMS view at start-up (Color figure online)

This sample scenario firstly generates an incident message from the BTMS, detecting the loitering behaviour in a restricted area of the hospital. The main fields of the corresponding JSON message contain:

```

1 { "detector": "BTMS",
2   "severity": "LOW",
3   "type": "INCIDENT",
4   "category": "Loitering",
5   "unique_identifier": "A77788",
6   "event": [
7     { "description": "Access the room",
8       "title": "Loitering",
9       "unique_identifier": "A77970",
10      "assets": [
11        { "id": 2000001,
12          "name": "Intensive Care Unit",
13          "category": "Building and Facilities"
14        }
15      ]
16    }
17  ]
18 }

```

Listing 1.1. Incident message example generated by BTMS

The *event* can include other parameters related to the equipment that initially detected the event as well as the links to media resources (videos records), that triggered the suspicious behaviour alert.

Upon receiving this incident message, the HAMS verify that the category of the incident is *Loitering*, with a severity level *LOW*. As a consequence, no actions are required on the assets, thus the status of the asset with id 2000001 (the “Intensive Care Unit” in the example) doesn’t change.

The next step of the attack is the infection of a medical device with malware in order to spread it over the network and expose patient health data. This cyber incident is detected by CTMS, the module responsible for detecting malicious cyber actions. The structure of the JSON message is similar to the previous one, but more events are considered. An extract of the resulting JSON is as follows:

```

1 { "detector": "CTMS",
2   "severity": "MAJOR",
3   "type": "INCIDENT",
4   "category": "Impact",
5   "unique_identifier": "A77988",
6   "event": [
7     { "description": "Install malware",
8       "title": "Execution",
9       "unique_identifier": "A77972",
10      "assets": [
11        { "id": 3000001,
12          "name": "Ventilator Unit",
13          "category": "Networked Medical Devices"
14        }
15      ]
16    },

```

```

17     { "description": "information leaks",
18       "title": "Execution",
19       "unique_identifier": "A77974",
20       "assets": [
21         { "id": 4000004,
22           "name": "Patient Health Information System",
23           "category": "Data and Records"
24         }
25       ]
26     }
27 ]
28 }

```

Listing 1.2. Incident message example generated by CTMS

This message describes a major incident that involves two assets: a *Ventilator Unit* (with asset id 3000001) and the *Patient Health Information System* (with asset id 4000004). Both assets are heavily compromised and don't work properly, thus their status must be changed to *unavailable*. According to the status description provided in Sect. 4, the Boolean value is set to *False*, the colour code is set to *red* and the stability is set to *deteriorating* (under the hypothesis that both assets were properly functions before the incident).

Finally, the HAMS generates an *availability* message to make the other module of the system aware of the change of the availability status of the involved assets. The *availability* message is a JSON message that contains the *id* of the incident and the list of the assets with their new status. The structure of the message is as follows:

```

1 { "incident": "A77988"
2   "assets":
3   [
4     { "assetId": 3000001,
5       "isOk": False,
6       "colour": "red",
7       "stability": "deteriorating"
8     },
9     { "assetId": 4000004,
10      "isOk": False,
11      "colour": "red",
12      "stability": "deteriorating"
13    }
14  ]
15 }

```

Listing 1.3. Availability message example generated by HAMS

Figure 4 shows the HAMS graph after the elaboration of the incident message, where it is possible to see that the Ventilator Unit in the Intensive Care ward and the Patient DB service are not available anymore.

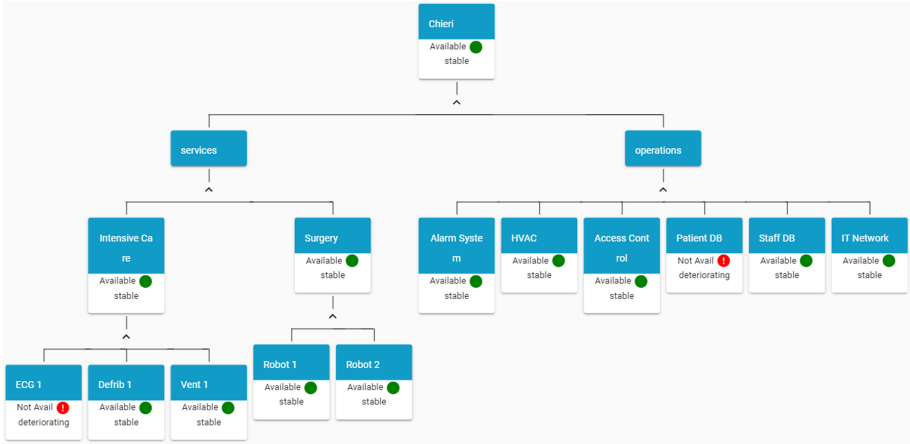


Fig. 4. HAMS view after receiving an incident message (Color figure online)

The cyber incident has a cascade effect also on other hospital assets. HAMS receives a further *impact message* generated by the IPDSM. This message contains a list of assets indirectly affected by the incident, each one associated with an *ImpactScore* parameter.

An example of *impact* message correlated with the *incident* message described above is as follows:

```

1 {
2   "Impacts": [
3     {
4       "IncidentID": "A#77988",
5       "IncidentType": "INCIDENT",
6       "IncidentCategory": "Impact",
7       "Assets": [
8         {
9           "AssetID": 1000001,
10          "AssetName": "Hospital Chieri",
11          "Risk_type": "malware diffusion",
12          "ImpactScore": 0.1
13        },
14        {
15          "AssetID": 2000001,
16          "AssetName": "Intensive Care",
17          "Risk_type": "malware diffusion",
18          "ImpactScore": 0.3
19        }
20      ],
21      {
22        "AssetID": 4000001,
23        "AssetName": "Alarm System",
24        "Risk_type": "malware diffusion",

```



```

24     "ImpactScore": 0.1
25   },
26   {
27     "AssetID": 4000003,
28     "AssetName": "Access Control System",
29     "Risk_type": "malware diffusion",
30     "ImpactScore": 0.1
31   },
32   {
33     "AssetID": 4000005,
34     "AssetName": "Staff Database",
35     "Risk_type": "malware diffusion",
36     "ImpactScore": 0.1
37   },
38   {
39     "AssetID": 4000006,
40     "AssetName": "IT Network",
41     "Risk_type": "malware diffusion",
42     "ImpactScore": 0.5
43   }
44 ]
45 }
46 ]
47 }

```

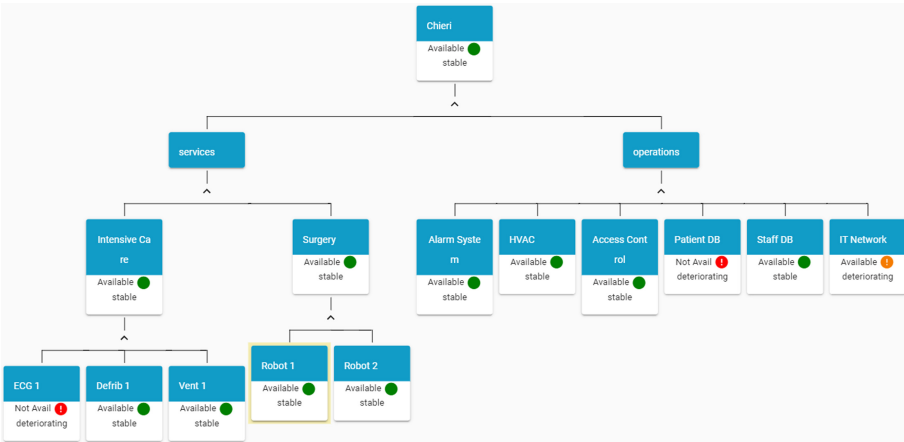


Fig. 5. HAMS after receiving an impact message (Color figure online)

As it can be seen by the message, the *incident* has impacted various assets in the hospital. However, the majority of them reports an *ImpactScore* of 0.1, meaning that the impact is quite marginal and there is no need to change the status of the asset.

The spread of a malware software on the network compromises the normal operations of the IT Network service even if it is still working. A *yellow* status is the updated availability value.

Figure 5 shows the HAMS graph after the elaboration of the impact message. In the figure, it is possible to view that the status of *IT Network* becomes *yellow* as a cascade effect from the incident.

As at least one asset changed its availability status, a new *availability* message is generated, following the same structure of the one generated after the reception of the *incident* message.

6 Conclusions and Future Work

The protection of Critical Infrastructures is a complex process that involves many pitfalls. SAFECARE project, and the modules presented in this paper, aim at enhancing the security and safety of healthcare and the *knowledge* about asset status of an hospital. This has potentially a great impact in term of benefits, especially during crisis situations like those experienced during COVID-19 peaks in the hardest hit areas, where specialized assets became quickly saturated and a percentage of patients needed to be transferred to other structures, in some cases even abroad.

The automated updating of assets status is an important feature in a context of evolving threats, where a single malware entering the internal communication networks can lead to harmful effects and fatal impacts on essential services for the people.

A further functionality that will be added to the platform will give the possibility of running a “demo mode”, simulating events and incidents without affecting the information stored in the database and trigger the other modules. This will be particularly useful to train the operators and design security exercises as a way of enhancing the awareness of all involved personnel.

References

1. Ripple20: 19 Zero-Day Vulnerabilities Amplified by the Supply Chain. <https://www.jsmf-tech.com/ripple20/>. Accessed 20 Aug 2020
2. Identifying and protecting devices vulnerable to Ripple20. <https://www.forescout.com/company/blog/identifying-and-protecting-devices-vulnerable-to-ripple20/>. Accessed 20 Aug 2020
3. Security Incidents in Healthcare Infrastructure during COVID-19 Crisis. <https://www.safecare-project.eu/?p=588>. Accessed 20 Aug 2020
4. Careem, M., De Silva, C., De Silva, R., Raschid, L., Weerawarana, S.: Sahana: overview of a disaster management system. In: 2006 International Conference on Information and Automation, pp. 361–366. IEEE, Colombo (2006)
5. Currión, P., Silva, C.D., Van de Walle, B.: Open source software for disaster management. *Commun. ACM* **50**(3), 61–65 (2007)
6. Health Resources and Services Availability Monitoring System (HeRAMS). <https://www.who.int/initiatives/herams>. Accessed 20 Aug 2020

7. Elamein, M., Woods, P., Elshaiekh, N.: Assessment of (HeRAMS) knowledge management system of humanitarian emergency in Sudan. *Int. J. Appl. Innov. Eng. Manag.* **3**(10), 63–67 (2014)
8. Maia, E., et al.: Security challenges for the critical infrastructures of the healthcare sector. In: *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures*. Now Publishers (2020). <https://doi.org/10.1561/9781680836875.ch8>
9. Bertone, F., et al.: Integrated cyber-physical security approach for healthcare sector. In: *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures*. Now Publishers (2020). <https://doi.org/10.1561/9781680836875.ch10>
10. Atigui, F., et al.: Vulnerability and incident propagation in cyber-physical systems. In: *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures*. Now Publishers (2020) <https://doi.org/10.1561/9781680836875.ch11>
11. Lubrano, F., Stirano, F., Varavallo, G., Bertone, F., Terzo, O.: HAMS: an integrated hospital management system to improve information exchange. In: Barolli, L., Poniszewska-Maranda, A., Enokido, T. (eds.) *CISIS 2020. AISC*, vol. 1194, pp. 334–343. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-50454-0_32
12. Dwarkanath, S., Waters, J.: *The Hospital AVailability Exchange (EDXL HAVE) Standard*. OASIS (2010)
13. O'Donnell, D., Wilkins, B., Brooks, R., Robertson, S.: *Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 2.0*. OASIS (2019)
14. Waters, J.: *Emergency Data Exchange Language (EDXL) Distribution Element Version 2.0*. OASIS (2013)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Towards a Global CIs' Cyber-Physical Security Management and Joint Coordination Approach

Vasiliki Mantzana , Eftichia Georgiou , Anna Gazi , Ilias Gkotsis^(✉) ,
Ioannis Chasiotis , and Georgios Eftychidis 

Center for Security Studies (KEMEA), P. Kanellopoulou 4, 101 77 Athens, Greece
{v.mantzana,e.georgiou,a.gazi,i.gkotsis,i.chasiotis,
g.eftychidis}@kemea-research.gr

Abstract. Critical Infrastructures (CIs) face numerous cyber-physical threats that can affect citizens' lives and habits, increase their feeling of insecurity, and influence the seamless services provision. During such incidents, but also in general for the security of CIs several internal and external stakeholders are involved, having different needs and requirements, trying to cooperate, respond and recover. Although CIs security management process is well analyzed in the literature there is a need to set a common ground among different CIs, thus reducing administration/coordination overhead and rendering the decision making and crisis management process more efficient. In this direction, this paper considers three different CIs (airport facilities, gas infrastructures, and hospitals); presents the current and emerging physical and cyber security related regulations and standards, operations, organisational and technical measure and; finally, through the discussion on gaps and best practices identified, proposes a global, cyber-physical security management and joint coordination approach. The proposed approach recommends among others that the adoption of a Holistic Security Operation Centre (HSOC) in each CI and a National Coordination Centre (NCC), supervising them, which will facilitate the communication and cooperation between the different CI operators and stakeholders, in case of an incident, that may have cascading effects to interconnected Infrastructures. The findings presented and the conclusions drawn are linked with three EU funded research projects (SATIE, SecureGas and SAFE-CARE), that aim to improve physical and cyber security of CIs in a seamless and cost-effective way.

Keywords: Critical infrastructures · Security management · Crisis management · Physical · Cyber · Stakeholders · Communication · Coordination centre · SOC

1 Introduction

The EC 114/2008 directive defines as Critical Infrastructure (CI) the assets, systems, and networks located in Member States which are essential to maintain the vital economic and social functions such as health, food, transport, energy, information systems, financial services, etc. The EC recognizes that these infrastructures must be protected from the disruption by natural disasters and man-made threats, and as such has launched the

European Programme for Critical Infrastructure Protection (EPCIP). The importance of physical and cyber security in CIs has never been more explicit. CIs in general, and especially in transport, energy and health sectors are exposed to various physical threats (i.e. terrorism, technological accidents, natural disasters, etc.) and cyber-attacks which are emerging especially with the increasing use of Information Systems. Now more than ever, CIs must be vigilant in establishing safeguards against physical and cyber threats, as it is imperative to have a solid understanding of the risks, vulnerabilities, security processes and technologies available. In addition, it is of paramount importance for CIs to establish a standardized crisis management process to deal with attacks that threaten to harm the organisation and stakeholders.

The aim of this paper is to describe three different CI types (airports, gas infrastructures, and hospitals); present the current physical and cyber security related regulations and standards adopted; identify their security operations, as well as the organisational and technical measures deployed by each CI; and finally describe a common, cyber-physical crisis management process encompassing the involved stakeholders. Moreover, gaps and best practices related to security issues are analysed and a global approach for CIs' cyber-physical security management and joint coordination is proposed. This approach recommends among others the adoption of a Holistic Security Operation Centre (HSOC), which will facilitate the communication and cooperation/coordination between internal stakeholders, and a National Coordination Centre (NCC) that will facilitate the communication/coordination between the HSOC and the external stakeholders. The NCC will also support the communication and cooperation/coordination between the different CI operators and stakeholders, in case of an incident that has cascading effects to interconnected Infrastructures. The findings and conclusions that are drawn, are linked with SATIE (H2020-GA832969) [1], SecureGas (H2020-GA833017) [2], and SAFECARE (H2020-GA787005) [3] projects, during the framework of which this research was conducted. These projects aim to provide solutions that will improve physical and cyber security in a seamless and cost-effective way and enhance threat prevention, threat detection, incident response and mitigation of impacts in airport facilities, gas and hospital infrastructures accordingly.

2 Critical Infrastructures Description

In the following paragraphs, a short description of the three CI types is provided, presenting information about their services, main functions and business operations, as well as basic involved assets.

2.1 Airport Facilities

Airports, being CIs that belong to the Transport sector, play a key role in people and goods transportation, as well as in regional, national, and international trade. Airports incorporate in their operational agenda passenger comfort, cost-efficiency, environmental protection, and policies for corporate and social responsibility. The interconnections and dependencies between the various systems and assets, combined with the presence of

different actors and the complexity of airport operations, expose the entire environment to cyber and physical threats and make it vulnerable to various attacks.

Being a complex ecosystem, various assets and systems are involved to support the airport operations, such as staff, sensors, cabling and fiber infrastructure, networking, and systems that support airport operations and the exchange of information and data among them. There is a plethora of systems that support the airport operations such as i) the Air Traffic Management (ATM) systems, the flight tracking systems, etc., which support the Airside Operations, ii) the fuel management systems, the parking management systems, the lightning detection systems, etc., which support the Landside Operations, iii) the access control systems, the baggage screening systems, the video surveillance systems (CCTV), the Explosive Detection Systems (EDS), etc., which support the Safety and Security Operations, iv) the self-service check-in systems, the Passenger Name Records (PNR), etc., which support the Passenger management, v) the building control systems, the environmental management systems, etc., which support the Facility and Maintenance Operations, and finally, systems that allow the exchange of information and data sharing among the various systems. The most common systems found in this category at airports are, among others, the Airport Operational Database (AODB), the Geographic Information Display System, etc.

2.2 Gas Infrastructure

Natural gas is a fossil energy source with a diverse range of uses and applications. It constitutes a major source of electricity generation, a powerful fuel for domestic (heating and cooking) and industrial use, a clean and cheap alternative as transportation fuel, and is also widely used for the production of hydrogen, fertilizers and other products. The Gas value chain can be divided in mainly three sectors: (a) Upstream (production) where the gas is extracted and processed; (b) Midstream (transmission) which generally includes the transport and storage. The gas is mainly shipped by means of high/medium pressure pipelines to downstream facilities, or other transportation media in case of Liquefied Natural Gas (LNG) (tankers, trucks and rails) and; (c) Downstream (distribution) which involves the final refinement and distribution of the gas including low pressure transport to the final users and sale. The Gas system is comprised of high-pressure gas transmission pipelines, gas compressor stations to maintain and regulate pressure, gas metering and distribution stations, cathodic protection systems installed to prevent corrosion of the pipeline, remote data transmission and telecommunication systems. Gas storage is also an important and critical part of the system. The distribution infrastructure enables the transportation of gas to the end users. Local distribution companies receive the gas in city gates, transfer points from transmission pipes and deliver it to individual customers. The delivery/distribution is done through an extensive network of small-diameter distribution pipelines throughout municipal and suburban areas. Natural gas end-users are residential, commercial, and industrial sectors and power-generation customers.

The complexity of the gas network, its diversity among transportation lines, the peculiarity of the areas crossed (remote or densely populated), the various production and storage facilities make it a challenging environment to cope with. Due to their distributed nature and often completely publicly known routings, the gas grids are prone to physical attacks, cyber-attacks (e.g. SCADA manipulations) and cyber-physical attacks.

Moreover, as interconnections of gas elements, interfaces with other grids, automated monitoring and regulation loops are increasing, besides cascading consequence effects, the emergence of novel types of threatening behavior are also expected. As gas availability is critical to so many other CIs, disruption in distribution can lead to series of consequent failures or disruptions causing cascading disasters. The overall objective of the industry and the governments is to avoid such events and in case it occurs, to minimize the impact of such events.

2.3 Hospitals

Health sector is responsible for delivering services that improve, maintain or restore the health of individuals and their communities; protect population against health threats as well as consequences of ill-health and; provide equitable access to people-centered care [4]. According to WHO, hospitals complement and amplify the effectiveness of many parts of the health system, providing continuous availability of services for acute and complex conditions [4]. Hospitals depending on their mission, offer different services such as pharmacy, pathology, radiology, nursing, acute (e.g. emergency department or specialist trauma centre, burn unit, surgery etc.); specialized (e.g. cardiology or coronary care unit, intensive care unit etc.) outpatient and chronic treatment etc. They are expected to provide appropriate and responsive care and; ensure acceptability and accessibility to its services.

Moreover, several new technologies, ranging from Internet of Things (IoT), wearable external and implanted medical devices (skin patches, insulin pumps and blood glucose monitors), order entry and administrative Information Systems (IS) to laboratory and operation theatre IS, have been adopted in hospitals. There is a widespread understanding of the need to balance utility and efficacy with privacy and security in innovation; however, technology is boosting more quickly than the creation, application, and adaptation/update of security measures. In addition to cyber-threats, physical threats are increasingly growing and even healthcare facilities are not immune to them. Hospitals are generally open to the public with multiple entrances, which means identification and baggage of visitors is almost never screened, leaving hospitals vulnerable to physical attacks. It has been reported that hospitals are twice more likely to experience a physical attack incident than a cyber-attack or breach [5].

3 Security and Protection of Critical Infrastructures in EU

In the following paragraphs, the three different CIs' (airport, gas infrastructures, hospital) crisis management related issues, such as regulatory framework, security operations, systems and technologies, as well as the crisis management process and engaged stakeholders are described.

3.1 Operation and Security Regulatory Framework

The first official effort for the preparation of a strategy to protect CIs was initiated by the European Council in 2004. In 2006, EU set the parameters for the implementation of the

EPCIP [6]. In 2008, the European Council Directive 2008/114/EC (evaluated on 2019 through public consultation and pending to be revised) established a procedure for the identification of and designation of European critical infrastructures ('ECIs'), focusing on the Energy and Transport sector, and the assessment of the need to improve their protection [7]. In accordance with the Regulation (EU) 2016/679, organisations including CIs, must protect natural persons while processing personal data and exchanging of such data. The principles of the EU Directive 2016/1148 (NIS Directive) concerning "measures of a high common level of security of network and information systems across the Union" are also applicable to CIs [8].

In the context of **airports** and additionally to the aforementioned, one year after the September 11 attacks, EU adopted a set of aviation safety and security rules based on Regulation (EC) No 1592/2002 and Regulation (EC) No 2320/2002 [9, 10]. In 2008, the EU extended the safety rules in order to cover the aircraft operations and aircrew licensing and training (Regulation (EC) No 216/2008), while in 2009 the extended Regulation (EC) No 1108/2009 covered the safety aspects of aerodromes, air traffic management and air navigation services. Currently, EC regulation 300/2008 establishes common rules in the EU to protect civil aviation against acts of unlawful interference, which pertain to the screening of passengers, cabin baggage and hold baggage, the airport security (access control, surveillance), the aircraft security checks and searches, the screening of cargo, mail, airport supplies and the staff recruitment and training. In addition, a national authority for aviation security must be appointed while establishing a national civil aviation security and quality programme. The detailed measures for the implementation of the common basic standards on aviation security are updated in "Commission implementing Regulation (EC) No 2015/1998". Moreover, the International Civil Aviation Organization (ICAO) works with Member States and industry groups to reach consensus on international civil aviation Standards and Recommended Practices (SARPs) and policies. The regulations and policies suggested by ICAO are adopted by the ICAO Member States to ensure that their local civil aviation operations and regulations conform to the suggested norms, to ensure safety and security. ICAO's annex 17 sets the preventive security measures relating to access control, screening of aircraft passengers and their cabin baggage, screening of hold baggage, screening of cargo and mail, measures for handling special categories of passengers, for protecting Information Systems (IS) etc.

With regards to **Gas CIs**, several initiatives and regulations specifically on the Gas sector focus on security, pinpointing the need for enhancing protection of such CIs, such as the European Energy Security Strategy [12], the stress tests on the resilience of the EU gas system [13] and the EU Regulation 2017/1938 on Security of Gas Supply [14]. The Directive 2009/73/EC establishes common rules for the transmission, distribution, supply and storage of natural gas. It lays down the rules related to the organisation and functioning of the natural gas sector, access to the market, the criteria and procedures applicable to the granting of authorisations for transmission, distribution, supply and storage of natural gas and the operation of systems [15]. Regulation (EC) No 715/2009 sets non-discriminatory rules for access conditions to natural gas transmission systems considering the special characteristics of the national and regional markets that ensure the proper functioning of the internal market in gas, the access conditions to LNG facilities and storage facilities taking into account the special characteristics of the national

and regional markets, and facilitating the emergence of a well-functioning and transparent wholesale market with a high level of security of supplying gas and providing mechanisms to harmonise the network access rules for cross-border exchanges in gas.

In the context of **hospitals**, there is also regulation on medical devices (Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices) [17]; on in-vitro diagnostic medical devices (Regulation (EU) 2017/746 of the European Parliament and of the Council of 5 April 2017 on in vitro diagnostic medical devices) [18] etc.

3.2 Security Measures and Technologies

For CIs of any sector, the selection among the security measures to be adopted depends greatly on the assets that need to be protected, their location and operational specifications, the threats, vulnerabilities, and risks associated to them. Thus, applying appropriate protection for the CI, by implementing the suitable security measures and technology, requires an understanding of the environment under consideration as well as the threats to which it is exposed.

3.2.1 Cyber Protection Measures

In order for CIs to: (a) prevent or at least reduce unauthorized access, use, disruption, deletion, corruption etc.; (b) respond effectively, timely and efficiently and; (c) minimize the impact of attacks to their network, information technology and systems, it is important to take both organisational and technical measures, as analysed below.

Organisational measures might include (a) the assessment of cyber risks, which is used to identify, estimate, and prioritize risk to organisational operations, organisational assets, individuals, other organizations, and the Nation, resulting from the operation and use of information systems [19]; (b) development and adoption of both generic and case specific laws, standards, plans and policies that outline cyber security measures and crisis management procedures, and; (c) staff training on cyber security protection and crisis management issues, standards, plans and protocols.

Technical measures adopted, include among others the following: authentication, access control (authorization), data confidentiality and integrity, backup, tracing systems, log files, communication security, firewalls, traffic monitoring systems, etc. A security by design approach should complete the aforementioned countermeasures, focusing on the cyber security aspects for new devices or systems, that need to be planned and implemented already from the beginning, meaning the procurement, design, development and maintenance phases. For securing networked devices and assets inventories should be created and maintained, as they can ensure a sound understanding of the systems and their components; support configuration and automated remediation management processes [20] and software should be regularly patched and updated.

In addition, it is also crucial to have a clear understanding of actual cyber security strategies and controls implemented at targeted infrastructures, such as: a) with regards to **airports**: Passenger Data Records, Flight Display System and Management, etc., b) with regards to **Gas CIs**: Supervisory Control and Data Acquisition (SCADA) which is one of the most common types of industrial control systems (ICS), being responsible for

providing automated control and remote human management of essential commodities and services, c) with regards to **hospitals**: medical devices, e-health services etc.

3.2.2 Physical Protection Measures

Investments in CIs physical security monitoring are likely to increase (21). Even if human observers theoretically offer greatest security, it is also necessary to take account the drawbacks of human inattention and limited senses. Generally, security monitoring requires several sensor devices that are based on more or less sophisticated technologies, basically according to the application need. Examples of tools and capabilities to create safe and secure **hospital** environments for patients, staff and visitors include perimeter protection, physical barriers/bollards, guards, lighting, audio and video surveillance, access control, intelligent controllers, Intrusion Detection System (IDS), Physical Security Information Management (PSIM) systems, Unmanned Aircraft Systems (UAS), anti-drone technologies, motion and temperature sensors, CBRNE sensors. These physical security measures though are applicable also to airports, **Gas Networks** or other CIs.

Airports implement and adopt some additional measures and technology solutions, such as baggage screening systems, baggage handling systems, Explosive Detection Systems (EDS), passenger screening systems, standardized screening techniques, which all passengers must undergo (e.g. baggage X-rays, metal detecting scans). Despite the fact that these measures are used for physical security, cyber security measures must be adopted to protect them.

In the context of **Gas CIs** the operation and integrity of the Gas system is usually handled by SCADA system, which monitors and controls systems remotely, both for operational and safety/security purposes. Remote Terminal Units (RTU) or local controls systems on production sites, pipelines, compressor and pump stations, regulation and metering stations, are connected to the SCADA by means of available communication media.

3.3 Security Related Operation Centers

Critical Infrastructures, based on their aim, functionalities and regulatory frameworks (as explained above), adopt several security related operations centers, given that any damage to a critical infrastructure, its destruction or disruption by natural disasters, manmade events or technological accidents, may have a significant negative impact for the security of the EU and the well-being of its citizens.

In addition to the previous and in order to deal with security issues on an organisational and technical level, **airports** have incorporated in their facilities, different operation centers to safeguard them in their daily routine. The Security Operation Centre (SOC) is a generic term describing part of or the whole platform whose purpose is to provide detection and reaction services to security incidents [21]. SOC monitors the security level of an organisation on an ongoing basis and comprises of a security team using various technological solutions in order to oversee security operations and to collect data and syslog to detect, identify, analyse, investigate and report cybersecurity incidents. SOC architecture models can differ based on airport's needs and preferences.

There are dedicated or internal SOC (team within organisation), virtual SOC (team works remotely), and co-managed SOC (internal IT collaborating with outsourcing vendor). The Emergency Operations Centre (EOC) is a facility operating to manage disaster emergencies. It is the place where information management, allocation and coordination of resources, and recovery actions take place. The Network Operations Centre (NOC) manages, controls, monitors and maintains the network functionality and operations across various platforms, media and communication channels (internal or external). The Airport Operations Centre incorporates a selection of the centers (including the previous ones) based on the operational needs of each airport. AOC constitutes an operational management structure that allows a common operational view to airport stakeholders in order to communicate, collaborate, coordinate and decide on the progress of airport operations. In case of a security incident the SOC operators would detect the security incident and immediately inform the AOC which is the focal point for information collection and sharing once an emergency is declared. Depending on the nature of the attack, the required stakeholders are determined, and the response and recovery measures are decided.

A similar structure but in a less sophisticated level applies also in **gas CIs**, where the Security Operation Centre, may include emergency and/or network operations centers. The SOC of a gas infrastructure is a secure control room within each infrastructure, where the SCADA, fire alarm systems, CCTV, and other sensors and systems are monitored and communication systems with stakeholders are established.

Hospitals though most of the times and due to the “open” philosophy of being built on in order to serve patients, they do have only security and network operational centers of small scale, or in some cases they do not operate such centers.

3.4 Crisis Management Process and Stakeholders Involved

Crisis management has been defined as “the developed capability of an organisation to prepare for, anticipate, respond to and recover from crises” [22]. The full cycle of crisis management can be described in four phases (Preparedness, Response, Recovery, mitigation), with several steps following. These steps are presented in Sect. 4, linked with the proposed security coordination and operational centers of this paper.

Within these steps, several internal and external stakeholders are involved, having different needs and requirements, trying to cooperate, respond and recover from the crisis. Security stakeholders can be categorized according to their involvement and perceived proximity to the organisation into internal and external.

Based on relevant literature review and information collected from the participating CIs (airports, gas infrastructures, hospitals), the following list summarises the common internal stakeholders: Board of Directors (BoD), Data Protection Officer (DPO), Crisis Management Team, Emergency response Team, Physical security manager/personnel, IT Security manager/personnel, Technical manager/personnel, Health and Safety manager. The common external stakeholders’ category includes individuals or groups outside the organisation who can affect or can be affected by a security incident in the CI, as they are conjoint into an interdependent relationship, namely: Law Enforcement Agencies, Fire Brigade, Emergency medical services, Civil Protection, National Authorities (Prefectures, Municipalities, etc.), Ministries (e.g. Energy, Transport, Health, etc.), National

Intelligence Agency, National Data Protection Authority, Interconnected/Interdependent Critical Infrastructures (e.g. power, communication, surface transportation), Computer Emergency Response Team (CERT).

In relation to **airports**, the following internal stakeholders were identified additionally to the aforementioned: Airport Duty Officer (ADO), Crisis Management Centre (CMC), Airport Operations Centre (AOC), Emergency Operations Centre (EOC)/Emergency Operations Team (EOT), Security Operations Centre (SOC)/Security Services Department, Media centre, Friends and relatives' assistance centre. External stakeholders involved in case of crisis are: International and EU Organisations (e.g. ICAO, EASA, EUROCONTROL), Air Accident Investigation and Aviation Safety Board (AAIASB), Civil Aviation Authority (CAA)/Aviation Authority, Air Traffic Control (ATC) (e.g. ENAV), Information Security Service Providers, Telecommunication Providers, Airlines, Ground Handlers, Cargo, Concessionaires,

In the context of **Gas CI**, external stakeholders that are also informed in case of emergency are Regulatory Authorities (e.g. Energy RA).

As regards to **hospitals**, the group of external stakeholders includes additionally the Public Health Control Centres (e.g. Disease Control and Prevention) and Regional Health Authorities.

3.5 CIs Security Management Gaps and Best Practices

Standardization of safety and security procedures has followed a sectoral approach while its maturity varies according to the criticality of the services provided. Thus, security management in the three CIs presented complies with international guidelines and standards (e.g. ICAO, ENTSO-G, WHO), which though do not offer the same level of integrity and maturity. Transfer of security organization approaches mainly from the aviation (airports) sector to the gas and to the less securitized environment of hospitals, is one of the opportunities identified in the paper to effectively address CI protection challenges. This will contribute to develop and use common management approaches and bring the security of the envisaged CIs at a comparable level.

The European Commission paved the way to integrate the management of the security of the ECIs through the EC Directive 114/2008, which was transferred to the national legislation of the member states since 2010. However, this directive addressed only the transport and energy sector and focused mainly to the threat of terrorism. Furthermore, the interconnection and interdependencies of the infrastructures have not been included at all as a systemic element of CI protection. Most of these challenges are planned to be addressed in the new relative directive, currently under elaboration.

Based on the previous analysis, reports on security management, and the daily challenges faced by the CIs, the following gaps have been identified with regards to the management of crisis and security as a whole:

Gap 1. Different physical-cyber security solutions implemented in different infrastructures: Among CIs, there is a lack of uniformity in the adoption and implementation of solutions that can support and enhance crisis management processes.

Gap 2. De-centralised control and collection of information: According to current usual practices, most CIs use multiple decentralised information gathering processes

that run in parallel (potentially overlap). Usually, there is no single coordination point acquiring the complete set of collected data for feeding it to the interested parties.

Gap 3. Lack of fast communication and information dissemination: CIs need to effectively and efficiently manage and share information (incident detection, evolution, resource allocation and management etc.), in different layers: within the CI, between the CI and its response partners, between the CI and the public, as well as among interconnected CIs.

Gap 4. Complexity of predicting the potential impact of an incident: a) within the CI (i.e. fire propagation, terrorist attacks, plum dispersion, impact of toxic chemicals, radioactivity etc.), and b) among interconnected CIs, as disruptions in one sector can have cascading effects in other sectors, including cross-border.

Gap 5. Crisis management process understanding: Although the crisis management process is well analyzed in literature there is a need for CIs to better understand the process, as well as to identify the involved stakeholders.

Gap 6. Lack of training and exercising in crisis management: To enhance readiness and cooperation to respond to any type of complex incidents and emergencies, all involved stakeholders should perform continuous training.

Gap 7. Different or no security plans within each CI: Standards and guidelines for the implementation of comprehensive plans for the security of a CI are needed at a national level (and if possible per CI sector) in order to build a common ground for all CIs. It is of high value to have a series of standardized plans (Risk and Vulnerability Assessment, Security Operations, Crisis Management, Business Continuity) related to preventive planning, day-to-day operations and business continuity management.

Despite the presented gaps, it appears that there are some best practices applied and used by the different CIs, as follows: (a) Airports and Gas CIs appear to have regulatory authorities/bodies that work with Member States and industry groups to reach consensus at international Standards and Recommended Practices (SARPs) and policies in security issues (**related to Gap 5, 6 and 7**); (b) Airports and Gas CIs use advanced physical and cyber integrated security solutions (e.g. SCADA, PLC etc.) (**related to Gap 1, 2 and 3**) and; (c) In order to deal with security issues, airports have incorporated in their structure different operation centers, e.g. AOC, SOC, in more structured and detailed way than the other two CIs (**related to Gap 2, 3, 4 and 5**).

4 Proposed Global CIs Cyber-Physical Security Management and Joint Coordination Approach

Based on the consideration of the CIs security related issues, the gaps, and best practices described in the previous paragraphs, the following recommendations are made that can enhance CIs' crisis management process, but also security as a whole: **(a) Recommendation 1.** CIs need to develop security plans and implement integrated physical and cyber security solutions (at a minimum common level depending on their needs) to protect their critical assets across their infrastructure (**related to gap 1 and 7**); **(b) Recommendation 2.** CIs should integrate in their organisational structure a Holistic Security Operation Centre (HSOC) to detect, analyze, and manage cyber and physical

attacks and to efficiently coordinate processes, people and technologies. Thus, a common operational picture will be achieved, and efficient information sharing will be facilitated, in order to alert operators and involved stakeholders to any potential threats or incidents **(related to gaps 2 and 3) and; (c) Recommendation 3**. The establishment of a National Coordination Centre (NCC) for CIP is also of high value. The NCC will interact with the HSOC of each CI, but also with external stakeholders (see Sect. 3.4) and other Coordination or Operational Centre (e.g. ERCC) on inter-national level. The main services and capabilities provided to the CI operators include risk and impact assessment, information gathering and sharing, and coordination of an incident **(related to gaps 4 and 6) and; (d) Recommendation 4**. A common cyber-physical crisis management process should be established and followed within each CI and at Member States level **(related to gap 5)**. In the following sections, a global, cyber-physical security management and joint coordination approach that addresses the aforementioned challenges, is presented. The proposed approach will facilitate the communication and cooperation between the different CI operators and stakeholders, in case of an incident, that may have cascading effects to interconnected Infrastructures and enhance security of CIs.

4.1 CIs Joint Coordination Approach and Structure

A fundamental precondition for the enhancement of the security and protection of CIs is a common centre for information sharing, reporting of problems, and exchanging of good practices. To this end, the implementation of a National Coordination Centre (NCC) for CIP which interacts with a Holistic Security Operations Centre (HSOC) within each CI is crucial. As depicted in the figure below, the HSOCs will manage internally in each CI the cyber and physical security in a seamless and integrated way, while acting as the single contact point between the CI and the NCC. The NCC will be the central node of the proposed approach, fully compliant with European standards (CIWIN [23] & ERNCIP [24]), used for: i) assessing risk and map hazards and threats, assisting in this way the CI operator to assess a specific risk and gather information for setting the appropriate level of security, ii) incident reporting and information sharing with external stakeholders, such as Emergency Response Services (e.g. Police, Fire Service, Civil Protection, CERT, etc.), Public Organizations/Bodies (e.g. Ministries, Regulatory Authorities, NIS, etc.), iii) information sharing with inter-connected Infrastructures, providing the necessary information about an incident and impact propagation, in order to be prepared and avoid cascading effects, iv) exchange of information and further communication with other Coordination or Operational Centres, on one hand at national level e.g. National Civil Protection Coordination Centre, 112, etc., which in its turn would provide the necessary information to the public in a structured and organised way; and on the other hand at European or International level, e.g. the Emergency Response Coordination Centre (ERCC) in Brussels, which will contact relevant Member States and organisations, if needed.

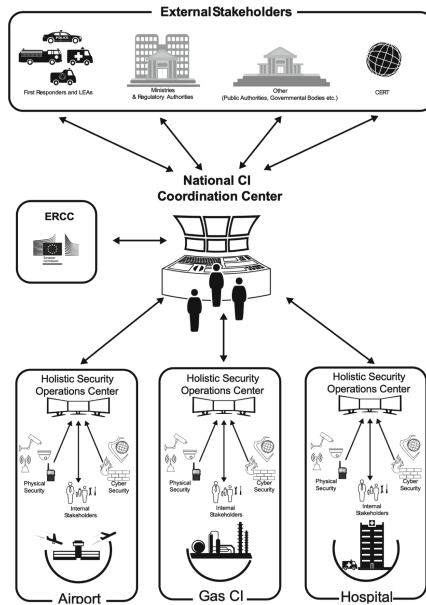


Fig. 1. Proposed global approach for CIs cyber-physical security management

The integration of the security of CIs at the national (and at the EU level for those of European interest) is linked with the concept of interdependency and interconnection of the infrastructures. It is common practice that each infrastructure takes care of the needs to ensure their business continuity, but they are rather indifferent of the challenges the disruption of their services may cause. Their responsibility ends up with providing compensation for the related impact according to signed Service Level Agreements (SLA). However, such compensation doesn't address the impact to the national economy and the societal disruption linked to loss of services due to interdependencies with CIs that fail to address a security incident. This is what a State Security Organization must ensure through a proper monitoring structure.

The main challenge to implement such structure is the lack of a concrete institutional and legal frame. Security pretexts and business interests are major obstructions that slow-down the required coming integration. The new European directive on CIP is a milestone ahead in this direction. The centralized role of CSIRT (Computer Security Incident Response Team) in the NIS Directive (EC 2016/1148) can be considered as the equivalent of the National Coordination node for monitoring and strengthening the physical protection of CIs at the national scale.

4.2 Global, Cyber-Physical Crisis Management Process

Further to the above proposed approach, a cyber-physical crisis management process should be established and followed within each CI and at Member States level. This process consists of four phases (Preparedness, Response, Recovery, mitigation), with several steps linked with the proposed centers depicted in Fig. 1.

Preparedness: The aim of this phase is to prepare CIs and develop general capabilities that will enable them to deliver an appropriate response in any crisis. It is a continuous cycle of planning, organizing, training, equipping, exercising, evaluating, and taking corrective actions that internal and external stakeholders should follow closely to ensure readiness. CIs is important to know which assets are vital for conducting their core activities, the potential threats against these assets, as well as their vulnerabilities. For this phase, appropriate institutional structures, supported by comprehensive policies, plans and legislation and the allocation of resources for all these capacities through regular budgets are also instrumental for thorough preparedness to crisis (**Step 1 – Develop Plans**). To improve the efficiency of the CI the appropriate tools must be in place (**Step 2 – Organise and equip**). These tools might include among others a list of contacts, hardware and software tools etc. Training and exercising are the cornerstones of preparedness which focus on readiness of all involved stakeholders to respond to any type of incidents and emergencies and on the identification of any discrepancies in terms of resources (**Step 3 – Train and exercise**).

Response: Response initiates when an incident is detected by an internal or external stakeholder or the Holistic Security Operation Centre (HSOC), in a manual or automated way (e.g. monitoring networks and early-warning systems, public authorities, citizens, media, private sector, etc.) (**Step 4 – Incident Detection**). Depending on the type of the incident (cyber and/or physical) different stakeholders will collect the information needed for further investigation. Additionally, information from multiple sources, such as sensors, social media and crowdsourcing could be collected by the HSOC. The information to be gathered usually includes details relevant to the type of incident, active and passive threats, the number and the type of casualties, geospatial information, images, video, etc. (**Step 5 – Information gathering**). The information should be collected and assessed by the Crisis Management Team (CMT) in cooperation with relevant stakeholders that identified the incident (**Step 6 – Incident assessment**). Getting a clear picture of the crisis (e.g. what happened, how many assets/people are or might be affected) is the basis for decision-making. The CMT should assess the extent of the crisis, evaluate the situation, determine, and define which response plan(s) should be activated (e.g. evacuation plan, etc.), inform the HSOC, which in its turn will communicate it to internal stakeholders and through the National Coordination Centre (NCC), external stakeholders. Based on the activated plans, response processes and procedures are executed, co-ordinated and adapted (**Step 7 – Determine plan**). It is also crucial to know the availability and current status of resources, in order to allocate them efficiently (**Step 8 – Resource Management**). HSOC is also responsible for communicating in timely and accurate manner information to internal stakeholders and to the NCC (**Steps 9 – Communication & 10 – Decision implementation**). The aforementioned steps could be repeated, until processes and assets return to business as usual or to another accepted status (demobilization) and the crisis is terminated. Demobilisation will be communicated by BoD and CMT coordinator to HSOC, which in its turn will communicate it to internal and external stakeholders through the NCC. (**Step 11 - Demobilisation**).

Recovery: When crisis occurs, CIs must be able to carry on with their tasks during crisis, while simultaneously planning on how they will recover from the damage the crisis

caused. Undeniably, required actions to return to normal operations and limit damage to CI and stakeholders continue after the incident or crisis. The CMT should decide the recovery actions to be taken (based on recovery plans), by cooperating closely with the HSOC, NCC, as well as internal and external stakeholders (**Step 12 – Recovery actions**). The CMT should collect and analyse evidence from the incident (**Step 13 – Collect and analyse**); and then should create an evidence report (**Step 14 – Create evidence report**). The CMT in cooperation with its coordinator should share relative information with all internal (**Step 15 – Share relative information with internal stakeholders**) and external stakeholders (e.g. Ministries, LEAs, fire brigade, interconnected CIs). Moreover, related investigations should be assisted (**Step 16 – Share relative information with external stakeholders**). As a crisis serves as a major learning opportunity, stakeholders should review the overall process as well as plans, procedures, tools, facilities etc., and identify areas for improvement (**Step 17 – Review incident response**). Following the evaluation, lessons learnt should be identified (**Step 18 - Debriefing**) and recommendations/revisions should be made to relevant plans, and processes (**Step 19 – Update plans**).

Mitigation: Mitigation refers to the process of reducing or eliminating future loss of life/injuries, assets and operations resulting from threats/risks through short and long-term activities. The results of the evaluation of the response actions should lead to recommendations for change, responsibilities allocation and relevant timelines in order to ensure that it will be carried out (**Step 20 – Take mitigation measures**).

5 Conclusions

The work presented in this paper was based on the findings and conclusions drawn from projects implemented on three different CIs, namely SATIE, SecureGas, and SAFE-CARE, as well as on the normative literature. The existing operations and security regulatory framework, the security measures adopted by CIs to protect their infrastructure, the security related operations centers incorporated in the CIs' organisational structure to safeguard them, as well as the crisis management process followed by the different CIs and the involved stakeholders, were studied. Moreover, the main gaps and potential areas of improvement of the CIs' security management processes have been discussed. Based on the conducted analysis, a global cyber-physical security management and joint coordination approach is proposed and presented. This approach recommends, among others, the adoption of a Holistic Security Operation Centre and a National Coordination Centre, aiming to support the communication, coordination and cooperation i) between a specific CI operator and its internal and external stakeholders, and ii) among the various affected CI operators and involved stakeholders, in case of an incident that has cascading effects to interconnected Infrastructures. Further to the proposed approach, a common cyber-physical crisis management process is described and proposed to be established and to be followed by CIs, at Member States level. Following the above, the proposed CIs' security approach aims to set common ground among stakeholders in managing incidents, thus reducing administration overhead and enhancing the process of efficient

decision making and information sharing, including best practices and lessons learned. Thus, this paper offers a broader understanding of the CIs' security management.

Acknowledgements. The work presented in this paper has been conducted in the framework of SATIE, SecureGas and SAFECARE projects, which have received funding from the European Union's H2020 research and innovation programme under grant agreements no. 832969, 833017, and 787002 respectively. This output reflects the views only of the author(s), and the European Union cannot be held responsible for any use which may be made of the information contained therein.

References

1. SATIE: Security of Air Transport Infrastructure of Europe. <https://satie-h2020.eu/>
2. SecureGas: Securing the European Gas Network. <https://www.securegas-project.eu/>
3. SAFECARE: Integrated cyber-physical security for health services. <https://www.safecare-project.eu/>
4. WHO: Health Systems (2019). <https://www.euro.who.int/en/health-topics/Health-systems/pages/health-systems>
5. Adelafa, L.: Healthcare experiences twice the number of cyber attacks as other industries (2018). <https://www.csoonline.com/article/3260191/healthcare-experiences-twice-the-number-of-cyber-attacks-as-other-industries.html>. Accessed Feb 2020
6. European Commission (2006). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52006DC0786>
7. European Commission (2008). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2008.345.01.0075.01.ENG
8. European Commission (2016). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016L1148>
9. European Commission: European Aviation Safety Rules. https://ec.europa.eu/transport/modes/air/safety/safety-rules_en
10. European Parliament: Air Transport: Civil Aviation Security. https://www.europarl.europa.eu/factsheets/en/sheet/132/air-transport-civil-aviation-security#_ftn1. Accessed July 2020
11. European Commission (2014). <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A52014DC0330>
12. European Commission (2014). https://ec.europa.eu/energy/sites/ener/files/documents/2014_stresstests_com_en.pdf
13. European Commission (2017). <https://eur-lex.europa.eu/eli/reg/2017/1938/oj>
14. European Commission (2009). <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32009L0073>
15. European Commission: Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices (2017). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32017R0745>
16. European Commission: Regulation (EU) 2017/746 of the European Parliament and of the Council of 5 April 2017 on in vitro diagnostic medical devices (2017). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R0746>
17. NIST: NIST - Digital Identity Guidelines (2019). <https://pages.nist.gov/800-63-3/sp800-63-3.html#def-and-acr>

18. ENISA: Securing Hospitals: A research study and blueprint. Independent Security Evaluators (2016). https://www.securityevaluators.com/wp-content/uploads/2017/07/securing_hospitals.pdf
19. Markets and Markets: Critical Infrastructure Protection Market (2020). <https://www.marketsandmarkets.com/Market-Reports/critical-infrastructure-protection-cip-market-988.html>. Accessed 10 Aug 2020
20. Bidou, R.: Security Operation Center Concepts & Implementation (2005)
21. British Standard Institute (BSI): BS11200: Crisis management – guidance and good practice. BSI (2014)
22. European Commission: Critical Infrastructure Warning Information Network (CIWIN). https://ec.europa.eu/home-affairs/what-we-do/networks/critical_infrastructure_warning_information_network_en
23. European Commission: The ERNCIP Project Platform. <https://erncip-project.jrc.ec.europa.eu/>
24. European Commission (2009). <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32009R0715>
25. European Commission (2015). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32015R1998>



Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Toward a Context-Aware Methodology for Information Security Governance Assessment Validation

Marco Angelini^(✉) , Silvia Bonomi , Claudio Ciccotelli ,
and Alessandro Palma

Department of Computer, Control, and Management Engineering “Antonio Ruberti”,
Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy
{angelini,bonomi,ciccotelli}@diag.uniroma1.it
palma.1871556@studenti.uniroma1.it

Abstract. Conducting a cybersecurity assessment is a central activity in protecting a generic organization from cyber-attacks. Several methods exist in research and industry to assess the security level of an organization, from manual activities to automated attack graphs. Unfortunately, automated approaches fail in taking into account the governance aspect that still need to be evaluated manually by the assessor, introducing possible biases or problems deriving from the level of expertise. In this paper, we provide a methodology to support the assessor in the task of evaluating the coverage of cybersecurity controls coming from technical standards, regulations, internal practices. This is done by providing him/her with a multi-layer model that takes into account several organizational layers, a mapping procedure to tie the security controls to the multi-layer model, and the definition of a validation factor that can be used to possibly refine the level of coverage and to suggest possible layers where evidences should be collected to verify and assess the coverage of a security control. A usage scenario provides an initial validation of our approach based on ISO 27001. Developments of this methodology are on-going toward its application to the support of broader cyber-risk assessment activities through discounting risk factors.

Keywords: Information Security Governance · Risk assessment · ISO 27001 · Multi-layer model

1 Introduction

According to NIST SP 800-100 [13], Information Security Governance (ISG) can be defined as *“the process of establishing and maintaining a framework and supporting management structure and processes to provide assurance that information security strategies (i) are aligned with and support business objectives,*

This work has been partially supported by the EU H2020 PANACEA project under the Grant Agreement n. 826293.

(ii) are consistent with applicable laws and regulations through adherence to policies and internal controls and (iii) provide assignment of responsibility, all in an effort to manage risk”.

Said differently, ISG is the global effort needed to ensure the well-being of the company’s electronic resources and it should be supported by effective and efficient processes. In order to evaluate company’s cyber security posture and to measure the adequacy and maturity of its ISG processes, periodic risk assessment must be carried out in order to estimate (quantitatively or qualitatively) the risk related to possible cyber incidents by evaluating (i) the level of exposure to specific threats analyzing existing vulnerabilities and the maturity level of the governance environment and (ii) the impact of the incidents generated by threats materialization.

How to correctly structure the risk management process is suggested by multiple standards and best practices. However, how to support, through automatic tools, assessors in this complex and delicate task is still an open problem, especially when assessing the governance aspects.

Currently, risk assessment is still mainly a manual process carried on by expert assessors through interviews with relevant stakeholders, analysis of technical elements (e.g., data extracted from network and vulnerability scans) and collection of evidences that allow them to properly evaluate the maturity of the governance process. The result is a non-completely objective and time-consuming activity where the result may be deeply influenced by the expertise of the assessor and his/her personal sensitivity.

Recently, automated approaches to dynamic risk estimation have been proposed to support risk assessors with a fast technical assessment (e.g., [7]). These approaches are based on attack graphs to estimate the likelihood of possible attack patterns and on mission impact models to estimate the consequences of a successful attack. Unfortunately, they fail in taking into account the governance aspects that still need to be evaluated manually by the assessor.

In this paper, we take a step to close this gap by providing a methodology to support the assessor in the task of evaluating the coverage of cyber security controls (e.g., the coverage of NIST Cybersecurity framework controls). This is done by providing him/her with the definition of a validation factor that can be used to possibly refine the level of coverage and to suggest possible layers where evidences should be collected to verify and assess the coverage of a security control.

This is done basically in two steps: (i) we map security controls performed by the assessor to a multi-layer model representing the most relevant components of an enterprise (i.e., we create a contextual map between controls and the company under analysis) and (ii) based on the obtained mapping, we compute a validation factor that provides the assessor with an indication about the need of a possible further analysis of the control due to its incidence on multiple organizational layers.

We also provide a usage scenario where we discuss how to apply our methodology to an assessment performed against ISO 27001.

The rest of the paper is organized as follows: Sect. 2 provides an overview of the main risk assessment methodologies and on attack graph models, Sect. 3 introduces the multi-layer model we used to identify relevant organizational layers used by the proposed methodology to map security controls, Sect. 4 presents our methodology, Sect. 5 introduces the definition of the validation factor for coverage, Sect. 6 discusses a usage scenario and finally Sect. 7 concludes the paper.

2 Related Work and Background

Risk Assessment Methodologies. The most common risk model is based on two factors: *likelihood* and *impact*. Currently, there exist different methodologies and tools supporting the two-factor risk assessment. OWASP [10] includes a risk assessment framework based on the two-factor evaluation. In the OWASP risk rating methodology the likelihood is estimated by assessing parameters related to threat agents (skill level, motive, opportunity and size) and vulnerabilities (ease of discovery and exploit, awareness and intrusion detection) while, for the impact, it takes into account technical impact (loss of confidentiality, integrity, availability and accountability) and business impact (financial and reputation damage, non compliance and privacy violation).

MEHARI (MEthod for Harmonized Analysis of RIsk) [6] is a free, open-source risk management methodology where the risk assessment task is decomposed in three main activities: (i) *risk identification* i.e., identification of assets, vulnerabilities and threats, (ii) *risk estimation* in terms of seriousness and (iii) *risk evaluation* in terms of its acceptability. As for OWASP, also MEHARI considers the two-factor risk but, in this case, both likelihood and impact are considered *intrinsic* (i.e., with no consideration of security measures) and then *reduction factors* may be applied (i.e., dissuasion and prevention for likelihood, and protection and palliation for impact).

EBIOS [3] is a risk assessment tool supporting the two-factor risk model. Differently from the others, it stresses the importance of the impact generated from different sources as humans, services, financial, legal and reputation. The assessment criteria used in EBIOS deal with exposure (dependency and penetration) and cyber reliability (maturity and trust).

When considering cybersecurity frameworks, we can relate the mentioned risk-assessment methodologies as follows. The mapping of OWASP risk rating methodology to the security framework NIST is such that some NIST functions are covered (Identify, Protect, Detect). However, considering the Respond and Recover NIST functions, the methodology suggests the general rule to fix first the most severe risks, but it does not offer a detailed approach to do it. Instead, MEHARI and EBIOS are strictly compliant with ISO 27000 family, with direct references to standard ISO 27005. Moreover, EBIOS offers details about security principles (e.g., anticipation, protection, defense, resilience) very similar to NIST functions, meaning that such tool can be applied to NIST framework.

Attack Graph Model and Risk Estimation. An attack graph represents possible ways via which a potential attacker can intrude into the target network by exploiting a series of vulnerabilities on various network hosts and gaining certain privileges at each step. A huge body of literature exists about attack graph generation and analysis and such models can be used both on-line (e.g. [2]) and off-line (e.g. [1]) to support security operators in their decision making process. More in details, focusing on the off-line usage, attack graphs can be used to

- determine optimal locations for the firewalls and intrusion detection/prevention systems [9,14],
- compute network security evaluation metrics [12,18,21],
- perform network security risk analysis [2,4,7] and
- compute near-optimal proactive defense measures [7,22].

Depending on the way information are represented, we may have two main categories of graphs:

- *State-based representations* [19] depict the whole state of the network for each node in the graph. The main advantage of this representation is its completeness (given the set of vulnerabilities in the network, the Attack Graph is able to represent all the possible attack scenarios). However, this is also its main limitation as it brings to an exponential cost (computation, size of the graph) with respect to the size of the network and the number of vulnerabilities.
- *Logical Attack Graphs* [16] are bipartite graphs representing the dependencies between vulnerabilities and security conditions. In this representation, duplicate paths are eliminated and a more compact representation is provided that scales polynomially with the number of vulnerabilities.

There are a number of attack graph generating tools and techniques, i.e., TVA (Topological Analysis of Network Attack Vulnerability) [15], NETSPA (A Network Security Planning Architecture) [8] and MULVAL (Multihost, multistage, Vulnerability Analysis) [17], that starting from a description of the environment (mainly from topology, routing restrictions and vulnerability scans) are able to generate the resulting attack graph. An alternative approach is to apply correlation techniques on network datasets to create attack graphs.

It is interesting to note that almost all the attack graph models and tools described here support the risk estimation. However, the risk is computed by considering only technical aspects, with governance aspects not taken into account.

3 Multi-layer Model

In this section, we briefly describe the multi-layer model defined in the context of the PANACEA Project¹ [5,11] that will be used as reference to identify relevant layers considered by our methodology to map security controls. This model extends the classical concept of attack graph by including multiple dimensions

¹ <https://www.panacearesearch.eu>.

where vulnerabilities can be identified and exploited to generate attack paths. We can distinguish two main different but interconnected components in the model:

- *Multi-layer Attack Graph*: modeling possible multi-step attacks exploiting the organization’s vulnerabilities (both of the assets and the personnel) to reach a target.
- *Business Dependency Model*: modeling the business processes of the organization and their dependencies to other business processes, services and assets.

Multi-layer Attack Graph. As pointed out in Sect. 2, existing attack graph models are relatively easy to build, e.g., by scanning the corporate network for technical vulnerabilities (i.e. CVEs) through automatic tools (e.g. [8, 15, 16]) but unfortunately they do not consider other sources of vulnerabilities like, for example, the human being. Indeed, an organization might install the most advanced technical solutions to protect its assets, still an attacker may circumvent them by exploiting a poorly trained employee which, e.g., leaves its credentials unprotected or is prone to provide sensible information through a social engineering attack or a phishing campaign.

The multi-layer attack graph model [5] is based on three interconnected layers (cfr. Fig. 1): (i) the *human layer* aiming at modeling employees, their relationships and personal vulnerabilities., (ii) the *network layer* modeling the ICT part of the company and (iii) the *access layer* modeling the credentials that humans (represented in the human layer) may use to access devices (residing in the network layer).

The aim of the *human layer* is to model how an attacker can compromise individual identities by exploiting human vulnerabilities of the personnel and their relationships. As shown in Fig. 1, the human layer is a subgraph of the multi-layer attack graph. Each node represents a possible level of use that an individual may get on digital identities. Edges are associated to exploitable human vulnerabilities. A directed edge from a node x_i to a node x_j , associated to a human vulnerability v , represents the fact that the human h_i having level of usage x_i on its own digital identity (or an attacker which has gained such usage privilege

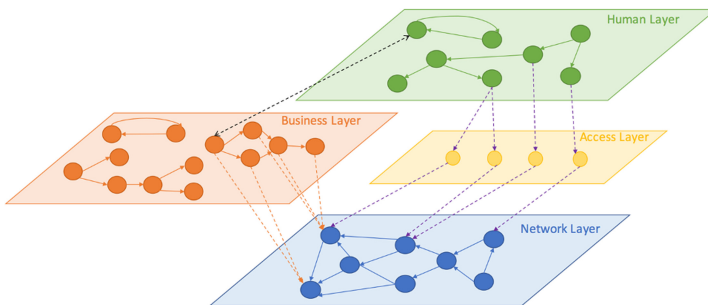


Fig. 1. Overview of the multi-layer model.

by exploiting a previous vulnerability) can get level of usage x_j on the digital identity of another individual h_j by exploiting the human vulnerability v on h_j (e.g., a social engineering attack). Notice that the human layer is a multi-graph, where multiple edges between the same pair of nodes are characterized by different human vulnerabilities.

The *network layer* has the aim to represent the ICT network infrastructure serving the organization mission and used by individuals represented in the human layer. Devices expose (technical) vulnerabilities that can be exploited by an attacker to gain access on them. An important concept to model cyber-attacks over IT devices is the *privilege level* that an attacker can gain on such assets. For instance, an intruder might start an attack from the Internet, i.e., with no privilege on the internal IT infrastructure of the organization, or in the case of an insider threat, they might have an initial given privilege on a machine. As a consequence of attack steps that involve vulnerability exploits, they might raise their privilege level on the current machine (privilege escalation) or gain privileges on other machines (remote privilege gain). To model this, each node of the network layer subgraph is associated to a given privilege level on a given device. A directed edge from a node z_i to a node z_j , associated to a (technical) vulnerability v , represents the fact that an attacker having privilege level z_i on a device d_i can get privilege level z_j on a device d_j by exploiting vulnerability v on d_j . Notice that also the network layer is a multi-graph, meaning that there can be multiple edges between each pair of nodes associated to different vulnerabilities.

Individuals are authorized to use assets via various kinds of access credentials, such as badges, tokens, or user accounts, which provide, to various extents, authorization/authentication mechanisms to the network assets. A credential represents the ability of an individual, having a given digital identity, to access a particular asset.

The aim of the *access layer* is to represent such credentials. Thus it is the layer that connects the human layer with the network layer (cfr. Fig. 1). In particular, nodes of the access layer subgraph represent credentials (e.g., a pair (username, password), a badge, a biometric key, etc.) or a two-factor authentication that is composed by multiple credentials. Nodes are characterized by a type (e.g., user/password pair, badge, token, etc.) and a level of robustness that can be used in order to weight associated risks when computing attack paths.

The *inter-layer edges* (cfr. Fig. 1), are those that connect the subgraphs of the three layers to form the multi-layer attack graph. An edge from a node x in the human layer to a node y in the access layer, represents the fact that an individual with digital identity x has a credential y , while an edge between y and a node z in the network layer represents the fact that credential y allows to get privilege z on the associated device.

Business Dependency Model. The business dependency model describes the business-level entities and their interdependencies. The business-level entities are partitioned into three disjoint sets: (i) *business processes*, (ii) *services* and (iii) *assets*. The assets of the business layer are the direct counterparts of the assets

of the network layer (e.g., physical/virtual hosts, network equipment, hardware devices, etc.) The services are the direct counterparts of the services and applications running on network layer assets (e.g., software components, applications, etc.). Therefore, there is an interconnection between the network layer of the multi-layer attack graph and these two classes of nodes of the business dependency model (cfr. Fig. 1). Conversely, business processes have no direct counterparts in the other layers of the model and represent the business processes of the organization. However, there may be an interconnection between a business node and a node in the human layer, e.g., modeling the fact that a particular member of the organization is fundamental to run a given business process (cfr. Fig. 1). The business dependency graph is a directed graph where there is a node for each business-level entity. A directed edge from a business-level entity be to a business-level entity be' models the fact that be depends on be' . In other words, in order for be to function properly, be' has to function properly.

4 An Assessment Validation Methodology

In this section we present our methodology to support the assessor in the security control coverage validation task and how it is supported by the multi-layer model. The proposed methodology is composed mainly of three sequential steps:

1. performing a control-based security assessment
2. weighted mapping of each security control on the multi-layer model
3. computation of validation factors confirming or suggesting review for the coverage level of the security controls.

In the first step, the security assessor conducts interviews with stakeholders to evaluate the compliance with respect to a set of security controls, collected by applicable technical standards, regulations, best practices. During this activity, the assessor collects, for each relevant security control, a set of evidences that will be used to finally score its coverage and the maturity of implementation. Let us note that, in real scenarios, this activity is prominently conducted manually by the assessor and his/her team, and is heavily based on the expertise of the assessor and on an inherent bias that can influence the final coverage estimation. Most of these activities tend to overweight technical aspects over other considerations (e.g., business processes, human component). Finally, the results of the evaluation can be biased also by the specific set of collected evidences.

The second step of our methodology provides the instrument to exploit the multi-layer model in the security assessment, in the form of a mapping between the set of security controls that is used during the assessment and the multi-layer model. In this way, the assessor can, for each security control, focus the attention on the specific layers one at a time and check if proper evidences have been collected for it, refining in this way the initial activity. This effort puts the focus on the construction of the mapping. Each security control can be mapped along the following dimensions:

Table 1. A sample of mapping for a subset of controls ISO 27001:2013 (H - Human, N - Network, A - Access, B - Business).

Control ID	Control name	H	N	A	B	Compl	Operat	Design time	Run time
A.5.1.1	Information security policies	1	0	0	3	4	0	2	0
A.6.2.1	Mobile device policy	1	3	2	0	4	3	2	0
A.7.2.3	Disciplinary process	4	0	0	2	4	0	2	0
A.8.1.2	Ownership of assets	NaN	NaN	NaN	NaN	4	0	2	0
A.9.2.3	Management of privileged access rights	0	0	4	0	0	4	0	2
A.10.1.1	Policy on the use of cryptographic controls	0	4	0	2	4	2	2	2
A.11.2.1	Equipment siting and protection	0	4	4	0	0	4	0	2
A.12.4.1	Event logging	0	0	0	3	0	4	0	2
A.13.2.4	Confidentiality or non disclosure agreements	4	4	4	4	3	4	2	0
A.14.2.2	System change control procedures	0	0	0	4	4	0	2	1
A.15.1.3	Information and communication technology supply chain	NaN	NaN	NaN	NaN	4	0	2	0
A.16.1.7	Collection of evidence	4	4	4	4	4	0	2	2
A.17.2.1	Availability of information processing facilities	0	0	0	4	0	4	0	2
A.18.2.3	Technical compliance review	0	0	0	4	4	0	2	0

- *Lifetime*: allows to link a security control to a specific part of the security lifetime by distinguishing between controls verified against design aspects of the system (e.g., policy design, network design, system configuration, etc.) and those verified on aspects related to execution time. In most cases these two dimensions are disjoint; however, some controls may be related, to different degrees, to both. An example is represented by security control A.7.2.1 of ISO 27001. This control, which regards “management responsibilities”, deals with compliance with policies and procedures, and therefore is mainly run-time. However, it implies also that policies and procedures must be well-designed in order to be applied, so it is linked to design time aspects too.
- *Impacted layer weight*: the controls are mapped to each layer of the multi-layer model (i.e., network, human, access, business) with a certain weight specifying how much they deal with human, access, network and business layers.

- *Management level*: models the impact of the security control over the business operational structure or the security organizational structure. It follows the definition by Von Solms [20] about the Information Security Governance model.

Concerning layers and management levels we map each security control with an integer weight between 0 (control totally not related to the layer/management level) and 4 (control totally related to the layer/management level). With such scale we can map the many different aspects that are inside layers and management levels. Instead, for the lifetime we introduced an ordinal scale with values LOW-MEDIUM-HIGH (translated in integer scale with values 0–1–2) defining how much the security control is related to the design and/or run-time.

We mapped all the security controls of ISO 27001 to our multi-layer model. Table 1 reports the mapping for a subset of these controls. We provide the full mapping as a supplemental material². Let us note that some controls (e.g. A.9.2.3, A.14.2.2, A.17.2.1, A.18.2.3) are completely mapped in only one of the layers of the multi-layer model while other controls such as A.6.2.1 and A.13.2.4 insist on multiple layers due to their generality.

In addition, there exist few controls for which a meaningful mapping cannot be established. Indeed, for controls A.8.1.2 and A.15.1.3 a “NaN” value is set, meaning that they cannot find a suitable mapping to the multi-layer model.

In the third and final step, we are going to review the assessment, based on the obtained mapping, and we are going to compute the validation factor as described in the following section. The steps of the methodology are then repeated in sequence until all validation factors confirm the coverage levels.

5 Computing Coverage Validation Factors

Validation factors play a central role in the methodology described in this paper because they support the understanding of how, in what way and to what degree we can better fit the security controls coverage with respect to assessment results. Once the validation factor has been figured out, then the cybersecurity assessor has the information about how reliable the assessed coverage level is for each security control. We can generally identify three main cases, identified by two reliability thresholds $T1$ and $T2$:

- *validation factor* $> T2$: the coverage level of the security control resulting from the assessment can be considered reliable enough (*Confirmed - OK*);
- $T1 < \textit{validation factor} < T2$: the results of the assessment are partially reliable, but a supplement of analysis could be necessary. At this point the mapping described in the previous section suggests the assessor which are the layers in which applying such procedures (*Recommended Review - RR*);

² The complete mapping can be found at the following link: <https://drive.google.com/file/d/1PHEbU38H4NtyzLiqHrZ-YczN-4NhBe5z/view>.

- *validation factor* $< T1$: the procedures considered in the assessment are not enough for covering the security control, therefore a revision of the security control and evidences is critically necessary (*Absolutely Review - AR*).

Reasonable values for such thresholds are $T1 = 0.3$ and $T2 = 0.6$, but other assignments are possible depending on the context. The computation of the validation factor is based on the assessment information collected in step 1 and the mapping produced in step 2. In the following we report how each of these pieces of information is interpreted in the computation of the validation factor:

- Coverage (*cv*): is the coverage level of the security control (i.e., the extent to which the provisions of the control are implemented). The coverage level is derived directly from the assessment and can assume the values C (Fully Covered), PC (Partially Covered) or NC (Not covered).
- Lifetime (*lt*): whether the control can be implemented at run-time or design-time; the rationale for this parameter is that if a control is implementable at run-time, it is generally easier to remedy and study. The parameter *lt* should be put at 0 if the control is totally design-time, 2 if it is totally run-time and 1 in all cases in which the distinction between the two is not evident.
- Management Level (*ml*): represents the extent to which the control is related to security operational management rather than compliance management. We account for this information as, generally, operational actions are more explicitly defined than compliance ones, thus, the coverage level of a security operational level control has more significance with respect to that of a compliance level control. Therefore, the confidence in a coverage level will be higher if the control concerns operational actions rather than compliance ones. We assign to this parameter a value between 0 and 4 (0: totally not operational, 4: totally operational).
- g_M : is the maximum gap between the level of mapping on the multi-layer model. We assign to each security control a level of mapping to the layers (human, access, network and business) with an integer scale $[0, 4]$ (the same we used for the *ml* value). The maximum gap between all the layers for a specific control models the uncertainty of the mapping of the control to the multi-layer model: the greater is the gap, the more the layers are disjoint, meaning that the security control is more defined.
- p_{RC} : is the reliability-coverage factor, interpreted as the degree of reliability to which a security control coverage contributes to the analysis of the security coverage of the assessment. We found three degrees of reliability:
 - HIGH: the security control is implemented at run-time and it is mapped in the multi-layer model;
 - MEDIUM: the security control is implemented at design-time and it is mapped in the multi-layer model;
 - LOW: the security control is not mappable into the multi-layer model.

With such information and taking into account the coverage level, we produced the following table containing the values of p_{RC} parameter for every case:

Reliability coverage	HIGH	MEDIUM	LOW
C	4	3	1
PC	3	2	1
NC	1	1	1

Considering the parameters described above the validation factor can be computed for each security control mapped in the multi-layer model with the following formula:

$$validation\ factor = \frac{lt + ml + g_M + p_{RC}}{lt_{max} + ml_{max} + g_{M,max} + p_{RC,max}}$$

Let us remark that if the lifetime is run-time instead of design-time and the management level is operational instead of compliance, then the validation factor increases due to the fact that the coverage of the controls is more precise. The more the gap between layers increases, the more the control is well positioned into the multi-layer model and the aspects to take under observation are more precise. Finally, the more the mapping is reliable, the more the formula is precise and the validation factor is high. The validation factor is normalized in the interval $[0, 1]$ by having in the denominator the sum of maximum values for each parameter ($lt_{max} = 2$ and $ml_{max} = g_{M,max} = p_{RC,max} = 4$). Once calculated, the validation factor represents the confidence of how much the coverage level of the security control fits. As an example, if a security control coverage is assessed as “completely covered” (C), and the computed validation factor is 0.4, then it means that the security controls should be reviewed (RR). The additional inspection of the relative mapping to the multi-layer model suggests which layers and associated security elements should be reviewed.

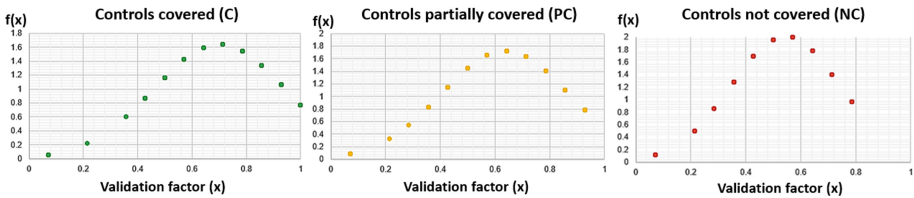


Fig. 2. Validation factor distribution in different cases of controls coverage.

5.1 Measurement Accuracy

In this section we illustrate the accuracy and reliability of the validation factor. For this purpose, we evaluate all possible cases of assessment for each security control of ISO 27001:2013, and we use such information to analyze the statistical distribution of data. The results are reported in Fig. 2: the behavior of the three cases (C/PC/NC) is the normal distribution evaluated through mean (μ) and

standard deviation (σ) in each case of coverage level (all C, all PC and all NC), and then applying the normal distribution function:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

The mean and standard deviation in the three cases are respectively:

- $\mu = 0.701754$ and $\sigma = 0.243285$ if controls are all covered;
- $\mu = 0.637845$ and $\sigma = 0.230917$ if controls are all partially covered;
- $\mu = 0.545739$ and $\sigma = 0.198709$ if controls are not covered.

We use such analysis for the evaluation of the accuracy and reliability of the validation factor because it expresses how much the reliability of coverage level can vary in the different cases, considering the average value. What the data shows is that the more the security controls are covered, the more the dispersion increases; in other words, the more a security control has a high coverage level, the more the variability of the validation factor increases (but still remains limited). This is due to the fact that when a security control is assessed near the full coverage, it means that many aspects of the control are assessed, and on each of them, depending on the mapping, the multi-layer model can express their validity or not. Instead, when a security control is assessed as not covered (or near low values for coverage), it means that many aspects of the security control has been found as not covered, or not considered, and the resulting validation factor from the multi-layer model is more clear-cut, resulting in less variability.

The mean expresses the expectation of the impact over different layers: if the validation factor has a very low value (e.g., lower than 0.3), it means that the control under analysis impacts on many different layers with different weights and thus the suggestion for the assessor is to verify that all the layers impacted have been considered in the determination of the coverage level. The verification can be done by using the mapping in order to identify layers that may be included in the evaluation. Contrarily, if the validation factor has a high value (i.e., between 0.7 and 1), it means that the control is clearly related to a specific layer and thus the expectation is that the assessor already verified the relevant elements in its analysis.

6 Usage Scenario

This section presents the application of the assessment validation methodology to a realistic scenario. The context in which it is applied is related to the health-care domain, that is showing an increased number of cyber-attacks in the last two years (even exacerbated by COVID-19) and an attention to cybersecurity that only lately is starting to become more deeply considered. As a resultant, this domain represents a good scenario in which the multi-layer model can be instantiated and provides benefits.

We applied our methodology considering as reference a unit of a hospital. The scenario of application is based on the security controls defined in ISO 27001,

with an assessment based on interviews and evidences collection limited only to the technological layer. We denote this procedure as “classic assessment”. While in principle this could be seen as a limitation, in practice many cybersecurity assessments are based on these assumptions. We take this situation as a reference for security assessment initiatives. ISO 27001 presents 18 categories for 114 security controls. The usage scenario covers all the security control categories, while not considering all the security controls for each category. Rationale behind this choice is that some of the controls are not applicable in the scope of this usage scenario and others are still under assessment or modeling. At the same time all the security control categories are covered with at least 1 security control bringing the total to 15 security controls.

Table 2. A sample of mapping with assessment for a subset of controls and the related validation (H - Human, N - Network, A - Access, B - Business).

Control ID	Control name	H	N	A	B	Compl.	Operat.	Design time	Run time	Cov.	Valid. factor	Valid result
A.5.1.2	Review of the policies for information security	1	0	0	3	4	0	2	0	PC	0.425	RR
A.6.1.1	Information security roles and responsibilities	4	0	0	1	4	0	2	0	PC	0.429	RR
A.7.2.1	Management responsibilities	3	0	0	3	3	4	1	2	NC	0.714	OK
A.8.1.3	Acceptable use of assets	3	3	3	4	1	4	2	0	C	0.571	RR
A.9.2.3	Management of privileged access rights	0	0	4	0	4	0	0	2	C	0.714	OK
A.9.4.3	Password management system	2	0	4	0	0	4	0	2	PC	0.929	OK
A.10.1.1	Policy on the use of cryptographic controls	0	4	0	2	4	2	2	2	C	0.857	OK
A.11.1.2	Physical entry controls	3	2	4	1	0	4	1	2	C	0.928	OK
A.12.4.1	Event logging	0	0	0	3	0	4	0	2	PC	0.857	OK
A.13.2.4	Confidentiality or non disclosure agreements	4	4	4	4	3	4	2	0	NC	0.357	RR
A.14.3.1	Protection of test data	NaN	NaN	NaN	NaN	0	4	2	0	NC	0.357	RR
A.15.1.3	Information and communication technology supply chain	NaN	NaN	NaN	NaN	4	0	2	0	PC	0.07	AR
A.16.1.7	Collection of evidence	4	4	4	4	4	0	2	2	NC	0.214	AR
A.17.2.1	Availability of information processing facilities	0	0	0	4	0	4	0	2	PC	0.928	OK
A.18.1.4	Privacy and protection of personally identifiable information	0	0	0	4	4	3	2	0	C	0.714	OK

Table 2 presents the results of this activity. The first two columns (Control ID, Control Name) reports the used security controls, where the following eight columns report the mapping to the multi-layer model valued for each of the security controls on the usage scenario. Finally:

1. “Coverage” column reports the coverage level related to security controls coming from the classic assessment;

2. “Validation Factor” column reports for each security control the resulting validation factor computed from the mapping;
3. “Validation Result” column reports the validity of each security control, coming from the interpretation of the validation factor, in terms of three possible results: assessment reliable (OK), assessment partially reliable (RR: Recommended Review) and assessment not reliable (AR: Absolute Review).

Looking at the results, the classic assessment presents five security controls fully covered, six partially covered and the remaining four not covered. We observe that security controls A.5.1.2 (Security policies review) and A.6.1.1 (security roles and responsibilities) represent similar situations in which the original assessment set a partial coverage, and that both controls are mapped to two layers with one layer prominent with respect to the other (Business layer for the first and Human layer for the second). This brings to a moderate validation factor that potentially could lower the coverage of the relative security control: for this reason it is recommended to review the assessment looking at the relative mapped layers. Focusing on security control A.8.1.3 (Acceptable use of assets), the classic assessment produced a full coverage for this control (C). The multi-layer model identifies Human, Access and Business layers as important, on top of the technical layer (Network). This mapping brings to a moderate validation factor, meaning that the assessment should be verified for elements concerning Human layer (e.g., misuse of organizations devices) or Access layer (e.g., verification of permissions policies). For this reason the final outcome is a recommended revision, that could probably lead to a PC coverage level. Security control A.9.2.3 (Privileged access rights) is assessed as fully covered (C). The multi-layer model identifies the Access layer as the only one interested by this control, with some contributions coming from lifetime (Design time) and management level. Overall the validation factor is above the $T2$ threshold and it confirms the coverage level as fully covered. Finally, we observe that the security controls A.14.3.1 (security of test data) and A.15.1.3 (ICT supply chain) have no mapping with the multi-layer model. A.14.3.1 is mapped on the security domain while A.15.1.3 is mapped on the operational domain, and both are labeled as design-time activities. The difference in management level provides a very low validation factor for the latter and a moderate one for the former, that weighted with high contribution for lifetime (Design time for both) brings to recommended review for the first and absolute review for the second security control. Overall, our methodology confirms the coverage level for 8 out of 15 (54%) security controls while asks review for the other half of them, with moderate revisions for 5 controls and strong revisions for other 2.

Concluding, the usage scenario showed where the proposed methodology can help in validating the coverage level of security controls with respect to classic security assessment procedures, supporting a more fit estimation for coverage level, highlighting security controls that needs a supplement of analysis and identifying for them the layers on which additional information should be collected.

7 Conclusion

This paper explored the idea of supporting the security assessment processes through a methodology that allows to better estimate the coverage with respect to security controls. The proposed methodology is based on a multi-layer model that captures the cyber-exposure from multiple perspectives, like human component and business inter-relations. It allows the derivation of validation factors that help in better estimating the coverage level of a security control and eventually identifying parts of the original security assessment that needs a supplement of analysis. The identification of additional areas on which to conduct the supplemental analysis can be directly inferred by the layers of the multi-layer model. The proposed usage scenario, even if of limited scope, successfully demonstrated the validation capabilities of the proposed methodology and the added value it provides with respect to classic security assessment methods, allowing to model the reliability of a security assessment considering (but not limited to) information security governance aspects.

Limitations exist in the proposed approach that we plan to resolve in future works. The first limitation concerns the hypothesis of a purely technological assessment conducted with classic methods. We plan to integrate the methodology in order to be able to assign each collected evidence to one or more of the layers of the multi-layer model, making the methodology able to support mixed initiative assessments. The second limitation concerns the scope of the usage scenario. We are currently working on broadening it, eventually obtaining a controlled environment that better suits a more robust validation of the obtained results. This aspect is linked to future enhancements of this methodology, to fully exploit the multi-layer model in supporting a complete risk assessment process. This future evolution will allow to consider not only the coverage level of a set of security controls, but also a formal definition of cyber-risk. In this scenario we are developing a “risk discount factor” supported by data computed through the multi-layer model (e.g., attack paths, human vulnerabilities) that can positively or negatively affect the risk estimation instead of only the coverage level. Finally we are working on generalizing the mapping to a broader set of security assessment frameworks (e.g. NIST Cybersecurity framework, CIS CSC).

References

1. Angelini, M., Blasilli, G., Catarci, T., Lenti, S., Santucci, G.: VULNUS: visual vulnerability analysis for network security. *IEEE Trans. Visual Comput. Graphics* **25**(1), 183–192 (2019)
2. Angelini, M., Bonomi, S., Borzi, E., Pozzo, A.D., Lenti, S., Santucci, G.: An attack graph-based on-line multi-step attack detector. In: *Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN 2018*, Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3154273.3154311>
3. ANSSI: EBIOS Risk Manager. <https://www.ssi.gouv.fr/en/guide/ebios-risk-manager-the-method/>. Accessed 12 July 2020

4. Beckers, K., Heisel, M., Krautsevich, L., Martinelli, F., Meis, R., Yautsiukhin, A.: Determining the probability of smart grid attacks by combining attack tree and attack graph analysis. In: Cuellar, J. (ed.) SmartGridSec 2014. LNCS, vol. 8448, pp. 30–47. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10329-7_3
5. Bonomi, S., et al.: Understanding human impact on cyber security through multilayer attack graphs. Technical report, Department of Computer, Control and Management Engineering, Sapienza University of Rome (2020). <https://bonomi.diag.uniroma1.it/research/publications>
6. CLUSIF: MEHARI (MEthod for Harmonized Analysis of RIsk). <http://meharipedia.x10host.com/wp/>. Accessed 12 July 2020
7. Gonzalez Granadillo, G., et al.: Dynamic risk management response system to handle cyber threats. *Future Gener. Comput. Syst.* **83**, 535–552 (2018). <https://doi.org/10.1016/j.future.2017.05.043>
8. Ingols, K., Lippmann, R., Piwowarski, K.: Practical attack graph generation for network defense. In: Proceedings of the 22nd Annual Computer Security Applications Conference, ACSAC 2006, USA, pp. 121–130. IEEE Computer Society (2006). <https://doi.org/10.1109/ACSAC.2006.39>
9. Jajodia, S., Noel, S.: Topological vulnerability analysis. In: Jajodia, S., Liu, P., Swarup, V., Wang, C. (eds.) *Cyber Situational Awareness. Advances in Information Security*, pp. 139–154. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-0140-8_7
10. Williams, J.: OWASP Risk Rating Methodology. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology. Accessed 12 July 2020
11. Coventry, L., et al.: D2.2 - Human Factors, Threat Models Analysis and Risk Quantification. PANACEA Project <https://www.panacearesearch.eu>
12. LeMay, E., Ford, M.D., Keefe, K., Sanders, W.H., Muehrcke, C.: Model-based security metrics using adversary view security evaluation (advise). In: 2011 Eighth International Conference on Quantitative Evaluation of SysTems, pp. 191–200 (2011)
13. Nist, Aroms, E.: NIST SP 800-100 Information Security Handbook: A Guide for Managers. CreateSpace, Scotts Valley (2012)
14. Noel, S., Elder, M., Jajodia, S., Kalapa, P., O'Hare, S., Prole, K.: Advances in topological vulnerability analysis. In: 2009 Cybersecurity Applications Technology Conference for Homeland Security, pp. 124–129 (2009)
15. Noel, S., Wang, L., Singhal, A., Jajodia, S.: Measuring security risk of networks using attack graphs. *IJNGC* **1**(1), 135–147 (2010)
16. Ou, X., Boyer, W.F., McQueen, M.A.: A scalable approach to attack graph generation. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, p. 336–345. Association for Computing Machinery, New York (2006). <https://doi.org/10.1145/1180405.1180446>
17. Ou, X., Govindavajhala, S., Appel, A.W.: MulVAL: a logic-based network security analyzer. In: Proceedings of the 14th Conference on USENIX Security Symposium, SSYM 2005, vol. 14, p. 8. USENIX Association, Berkeley (2005)
18. Pamula, J., Jajodia, S., Ammann, P., Swarup, V.: A weakest-adversary security metric for network configuration security analysis. In: Proceedings of the 2nd ACM Workshop on Quality of Protection, QoP 2006, p. 31–38. Association for Computing Machinery, New York (2006). <https://doi.org/10.1145/1179494.1179502>
19. Sheyner, O., Wing, J.: Tools for generating and analyzing attack graphs. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) *FMCO 2003. LNCS*, vol. 3188, pp. 344–371. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30101-1_17

20. Solms, S.V., Solms, R.V.: Information Security Governance. Springer, Boston (2009). <https://doi.org/10.1007/978-0-387-79984-1>
21. Wang, L., Jajodia, S., Singhal, A., Cheng, P., Noel, S.: k-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities. *IEEE Trans. Dependable Secure Comput.* **11**(1), 30–44 (2014)
22. Wang, L., Albanese, M., Jajodia, S.: Network Hardening. SCS. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-04612-9>

Impact Propagation and Power Traffic Analysis



Impact Propagation in Airport Systems

Corinna Köpke¹(✉), Kushal Srivastava¹, Louis König¹, Natalie Miller¹,
Mirjam Fehling-Kaschek¹, Kelly Burke², Matteo Mangini², Isabel Praça³,
Alda Canito³, Olga Carvalho⁴, Filipe Apolinário⁴, Nelson Escravana⁴,
Nils Carstengerdes⁵, and Tim Stelkens-Kobsch⁵

- ¹ Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI,
Am Klingelberg 1, 79588 Efringen-Kirchen, Germany
Corinna.Koepke@emi.fraunhofer.de
- ² DGS S.p.A. - NIS Network Integration and Solutions S.r.l., Via XX Settembre 41,
Genova 16121, Italy
kelly.burke@dgsspa.com
- ³ GECAD - Research Group on Intelligent Engineering and Computing for Advanced
Innovation and Development, School of Engineering (ISEP), Polytechnic of Porto
(IPP), R. Dr. António Bernardino de Almeida, 431, Porto, Portugal
icp@isep.ipp.pt
- ⁴ INOV-INESC INOVAÇÃO, Rua Alves Redol, nr. 9, 1000-029 Lisboa, Portugal
filipe.apolinario@inov.pt
- ⁵ Institute of Flight Guidance, German Aerospace Center (DLR),
Braunschweig, Germany
Tim.Stelkens-Kobsch@dlr.de

Abstract. The effective protection of critical infrastructure against cyber and physical security threats involves many different steps from initially the identification of risks to finally the implementation of counter measures in the infrastructure. To derive counter measures and to come to intelligent decisions facing the identified risks, the impact calculation plays a central role. The impact of a specific threat can propagate through the systems of the infrastructure and thus needs to be analysed carefully. In this paper, the role of impact propagation of cyber-physical threats for infrastructure protection is discussed, exemplified for airport systems. In the ongoing EU-H2020 project SATIE (Security of Air Transport Infrastructure of Europe) a toolkit is developed containing two tools for impact propagation, namely the Business Impact Assessment (BIA) and the Impact Propagation Simulation (IPS). Both tools are described and for a small test case the propagation of a cyber threat and the transformation into a physical threat is demonstrated in a network representation as well as an agent-based model of the airport's systems employing the IPS.

Keywords: Airport · Cyber-physical attack · Impact propagation

1 Introduction

In the world of critical infrastructure, the systems involved have direct and indirect interdependencies and are vulnerable to each others threats, risks, impacts and disruptions, whether deliberate or accidental, [34]. Interdependencies between systems are crucial for the protection of critical infrastructures because they can allow a failure which is seemingly isolated in one system to cascade and impact several other systems [39]. For example, when there are severe weather conditions at one airport, turning down its throughput, this causes delays and can cause a whole continent to feel the effects as the delays propagate to all other connected airports, creating further delays as the aircrafts continue their flight schedule to other destinations, but with the delays ever more compounded [30]. This cascading effect is predictable with appropriate propagation models.

Cyber-threats are by no means immune to this phenomenon as they can quite easily spread across networks, especially in our increasingly connected world, but also cyber-threats can even turn into physical or even safety threats [17]. Cyber-threat impact assessment allows organizations to understand how a cyber-threat can damage their information technology (IT) infrastructure and identify the critical services that could be impacted by the threat. These techniques distinguish themselves in three key points: impact modelling, propagation and assessment.

For impact modeling, a correlation model is often constructed using an interconnected graph [1, 7, 11, 12, 21, 28, 29, 31, 43] that profiles relationships between organization assets and critical services. Some portray impact modelling [1] by inspecting communication exchanged by IT devices to represent the organization's environment as an interconnected graph. Other works [12] expand this graph to also include a higher-level mapping between assets and the organization critical services. Also, some studies [43] integrate a security layer to include information about how vulnerable assets can influence impact propagation.

Within the air transport environment, it is not difficult to imagine how risks to a cyber-attack can result in potential physical damage. In fact, a ransomware attack to a Cleveland Airport server in April of 2019 prevented the displaying of the flight information to passengers, preventing them from finding the correct gate, causing confusion, crowds, and panic among the passengers, then overburdening the airport staff, and causing flights to be grounded [27]. There was no further physical attack, but the risks to one were greatly increased because of that single cyber-attack.

Airports in particular are highly complex systems with a multitude of stakeholders, each of them with their own intentions and goals. Various interconnections exist between these stakeholders, and between the assets of the airports (e.g. airports themselves, baggage, passengers, etc.), making it difficult to assess the impact of a particular action on the whole airport and even system of connected airports. Therefore, it is vital to quickly detect threats and understand its potential impact on the rest of the complex system. This is not a novel

issue, though, so there have been some attempts at aiding this understanding of propagating threats and coordinating efforts between stakeholders. The first was the airport collaborate decision making (A-CDM, [5, 6]), which just shared some basic airside information (e.g. the Target Off-Block Time, TOBT). Then the Total Airport Management (TAM, [9, 36]) research aimed to provide stakeholders with a holistic view of the airport, its performance and causes for degradations. The necessity of better collaboration was underpinned by a job and task analysis at German airports which revealed that coordination was still lacking between stakeholders, and airport performance was sub-optimal due to the lack of timely or precise information [35]. As a concept, TAM is built around the idea of an Airport Operations Centre (APOC) where the main stakeholders are collaboratively working on a plan to improve the airport performance. More recently, [41] presented a concept for a what-if tool for TAM, to support decision-making and predicting overall effects of stakeholder actions and the comparison between several alternatives.

The lessons learned in TAM can be transferred to security research in aviation as shown by SATIE. Here it is foreseen that security practitioners and airport managers collaborate more efficiently during a crisis to achieve its mitigation. This can be done in a security operation centre (SOC), where the operators are informed about alarms and the reasons of the alerts. In a SOC emergency, procedures can be triggered simultaneously through an alerting system in order to reschedule airside/landside operations, notify first responders, cyber-security and maintenance teams towards a fast recovery. Information about impact propagation information will ultimately improve the situational awareness of the decision makers and will lead to better and faster decisions.

However, the availability of additional information may also overwhelm human operators and, especially in these situations, an automated decision support has the potential to assist the quick resolution of a critical situation or even a crisis [2]. Therefore, a unified tool, taking into consideration all of the physical and digital assets involved, their interconnections, and how threats can propagate through the system would be ideal.

The mechanisms of probabilistic propagation between assets is applicable in a variety of fields, but is particularly critical in the air transport world where lives are at stake. Effective prediction of such effects allow for much more efficient and responsive countermeasures to reduce those risks at a single point which could become devastating once they propagate through the system. In this paper, impact propagation in airport infrastructure and the corresponding systems is discussed in the context of the project SATIE. Two systems of the SATIE toolkit perform this type of impact analysis, namely the Business Impact Assessment (BIA) and the Impact Propagation Simulation (IPS), which are described in Sect. 2. In Sect. 3, the IPS is applied to a small test scenario where cyber-physical threats impact some specific airport systems. Note, we present preliminary results of the tool development in the course of the ongoing project. In Sect. 4 the results are summarized and an outlook is given.

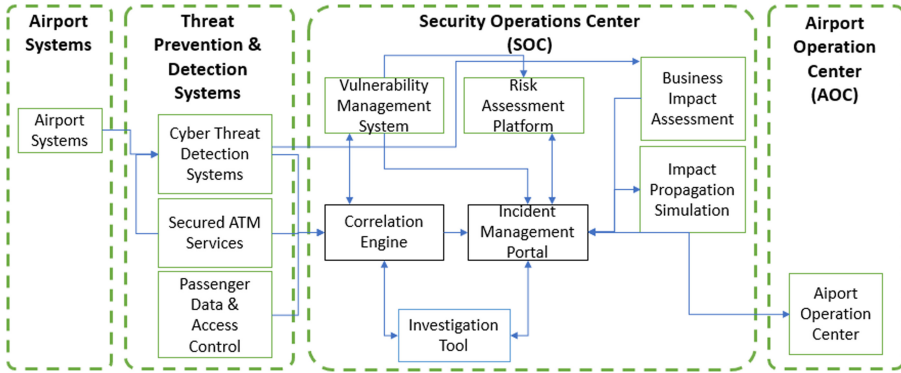


Fig. 1. Simplified architecture of the SATIE toolkit.

2 Approach

2.1 SATIE Toolkit and Ontology

SATIE proposes an architecture that combines new and existing tools in order to timely detect cyber-physical incidents, simulate their impact and deliver information to the Security and Airport Operation Centers. Live data is captured through several sensors and existing tools in the airport systems, which is then analysed through several threat prevention and detection systems. These systems will analyse data according to different perspectives and may describe events and trigger alarms. These are sent to the Security Operation Center, namely to the Correlation Engine, which will attempt to find possible correlations between the messages obtained through the different sources. Querying either the Vulnerability Management System and the Risk Assessment Platform will supply additional information about the events that generated the alerts, particularly regarding possible vulnerability exploitation and affected assets. Information is centralized in the Incident Management Portal, where a human operator can opt to aggregate different events into incidents and query the impact simulation tools in order to know which assets could be affected by a single incident, and obtain possible mitigation strategies. Figure 1 shows a simplified version of the overall SATIE architecture. The IPS receives incidents from the Incident Management Portal and requests for simulation. It facilitates to the Incident Management Portal a visualization of possible threat propagation paths and mitigation strategies, depending on the threat under consideration.

An ontology is developed in SATIE to describe the contexts of the exchanged messages, namely the sub-domain of incidents, impacts and assessments. The impact's specification is directly related to the needs of the BIA and IPS, describing how the performance of assets may be affected, how assets affect each other and suggesting possible mitigation strategies, each with their own expected

performances. How different events and assets may affect each other is described by the threat propagation path and the threat propagation event concepts respectively.

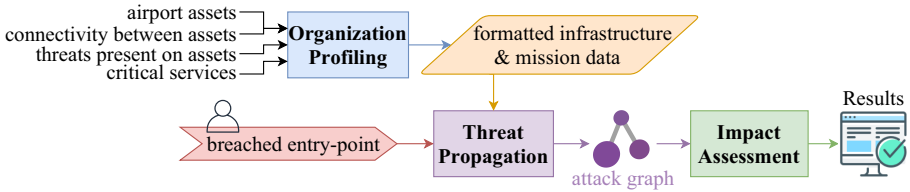


Fig. 2. Architecture of the Business Impact Assessment (BIA)

2.2 Business Impact Assessment (BIA)

The SATIE toolkit includes the BIA, that analyses how cyber-threats propagate to the organization assets and assesses the impact caused on the organization’s business critical services and goals. As can be seen in Fig. 2, BIA performs impact assessment in three stages, organization profiling stage, threat propagation stage and impact assessment stage. At the first stage, BIA gathers information about the organization’s infrastructure using reconnaissance techniques [22] that automatically query inventory system and continuously inspect IT network traffic to discover the organization assets, and how threats can propagate by means of communications between organization assets. The system also resorts to process mining techniques [45] for identifying asset involvement in the organization critical services by inspecting registry of operations performed by each asset. At the second stage, BIA simulates the propagation of a user-chosen threat based on an attack graph model, where the goal is to determine whether a compromised asset is likely to deleteriously affect any of the critical services of the organization. To this end, logic programming and attack graphs are used to express the rules and preconditions that must be met for threat propagation to occur [33]. At the third stage BIA traverses the attack graph to identify assets affected by threat propagation and determine the critical services compromised. The result of this final analysis produces a report of the cascading effects the simulated threat had on the organization infrastructure, highlighting the assets affected in each threat propagation steps and critical services potentially damaged by the threat. Ultimately, this final report may be used to aid risk analysis by simulating impact propagation of a potential exploited threat.

For impact propagation, a model-based analysis is used and can be categorized into *logic-based models* [1, 3, 4, 12, 14, 18, 21, 28, 29], *probabilistic-based models* [3, 23, 37, 42] and *sensitivity-based models* [7, 20, 24–26, 31]. Logic-based analysis approaches are based on an attack graph model that gradually analyzes the cascading effects a cyber-threat can have on the organization assets and how they can be exploited to compromise the critical services. Probabilistic-based

impact propagation often uses Bayesian Networks to express how assets can be compromised and how critical services are impacted by conditional probabilities. Sensitivity-based approaches use active perturbation by purposely compromising an asset to identify how it impacts the critical services. For impact assessment, several works distinguish themselves on how they evaluate the impact propagation cascading effects on asset and critical services operability. Some use qualitative metrics [8, 15, 16, 21, 37, 38], to evaluate the overall impact based on risk categorization on how vulnerable assets and critical services are to cyber-threats and how easily they can be exploited by attackers. Other works use quantitative metrics [12, 14, 16, 43], to measure how cyber-threats impacts the operationality, exposure and efficiency of assets and critical services.

2.3 Impact Propagation Simulation (IPS)

The IPS follows a different approach than the BIA, i.e. the focus is more on system's assets and passenger behaviour than on business processes. Furthermore, the IPS accounts for cyber and also physical threats. Even if this is not foreseen in the project, the BIA and IPS approaches could be integrated in future work. The IPS is based on two complementary modeling approaches, namely a network and an agent-based model.

Network Model. The network model of IPS is based on CaESAR (Cascading Effects Simulation in urban Areas to assess and increase Resilience), a tool developed at Fraunhofer EMI [10]. This tool simulates networks topologically as nodes and edges. In the context of airports, the nodes are assets such as serves, card readers, switches and edges are e.g. cables, information exchange or influence in case of an incident. Using node and edge lists, more details can be implemented into the simulation, including the direction of flow with the edges, and the mean time to repair for the nodes.

Different types of threats can be modelled including ones that only attacks certain types of nodes, or threats that change or move over time (such as a natural disaster), or ones that only affect nodes in a specific region. Once an adverse event is simulated, nodes are removed from the model as they are damaged. Once the threat has completed its propagation, the recovery begins with the nodes being added back to their network once the repair times have finished.

Mitigation measures can also be implemented into this simulation tool which allows for resilience improvements to be made. Different measures are selected and the simulations are completed again. The effectiveness of the measures are compared utilizing the performance time curves that the tool outputs. The measures can have different effects on the different aspects of the curve, and thus effect the system's resilience. Effects can be seen for example, on the damage sustained, or the recovery required. Depending on the goal for the resilience of the networks, the best mitigation measures can be determined. Additional models can be combined in CaESAR to determine how the threat can propagate to other networks and investigate cascading effects. This can be done with very different networks such as e.g. telecommunications, power and water networks.

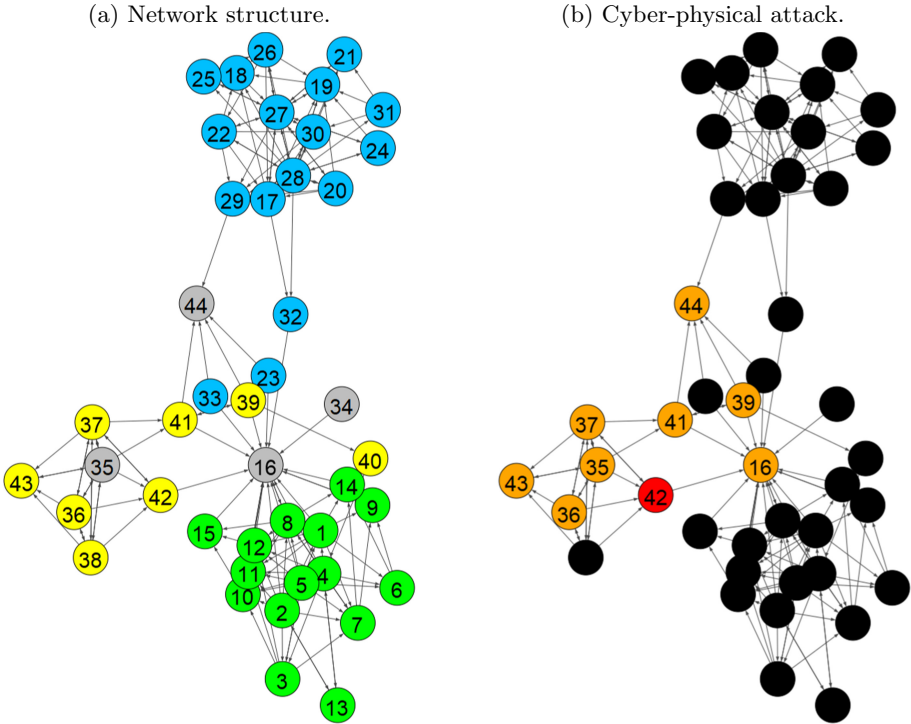


Fig. 3. Network structure for test scenario, involved systems are: Physical Access Control System (green), Flight Information Display System (FIDS) (blue), Public Announcement (PA) System (yellow) and others (grey). During the cyber-attack: The initial attack is placed on node #42 (red) and propagates to #43 finally impacting #44 and #16. The impacted nodes and the nodes of the propagation path are given in orange. (Color figure online)

Agent-Based Model (ABM). To observe an infrastructure in various threat situations and to explore possible states of the system, ABM can be employed [44]. ABM has been applied in different fields such as e.g. train logistics [32], air traffic management [40] and evacuation in case of fire [13] and respective languages and platforms have been developed [19, 46]. In this paper, we present an ABM approach especially developed in SATIE to model and simulate the behaviour of passengers in an airport. Passengers are modelled as individual agents that move independently in the airport dependent on their current situation and state. The path in the airport depends on passenger attributes such as number of bags to check-in, online check-in, walking speed, memory, flight number and airport attributes such as treatment times and general layout. The walking speed, number of bags and online check-in vary randomly. The flight number decides on the check-in desk and gate. The memory of each agent influences how often they check the Flight Information Display System (FIDS) monitors

to get information on their flights. Further, they follow public announcements (PA) and will receive new goals to walk to as announced. For both, FIDS and PA it is assumed that passengers follow the requests announced or displayed without questioning. However, more aspects of human behaviour will be taken into account in the further development of the implementation.

Hybrid Model. The transformation of threats from cyber to physical motivated the creation of a combined model. In the project SATIE, a hybrid model consisting of a network representation and an ABM is developed. The network threat propagation is triggered by received incidents, as described in Sect. 2.1 and the ABM is executed once a transformation from cyber to physical is detected or critical nodes such as employees or passengers are endangered. The network model enables the holistic overall view and the ABM gives insight into detailed processes and passenger behaviour in the airport infrastructure. The goal of the hybrid model is to combine the holistic and detailed view to enable impact propagation on each level of detail whenever it is required.

3 Application

3.1 The Test Scenario

The airport network structure for impact propagation is based on three airport systems, namely the Physical Access Control System, the FIDS and the PA System. The corresponding network representation is given in Fig. 3(a). These networks are interrelated and the main connecting nodes are #16 and #44, which represent employees and passengers respectively. This finding underlines the importance of the ABM. A detailed representation of passengers and employees in the infrastructure enables to better understand the interrelations between airport systems. The test scenario that is employed in this paper to test and demonstrate the impact propagation using the hybrid model is based on a cyber-attack on the PA system. During normal operation and without any threat or interruption, the attacker announces an evacuation of the whole terminal. Here, we simulate (i) the propagation of the cyber-attack in the network and (ii) the impact of the evacuation announcement on the passengers with the ABM.

3.2 IPS Results Based on the Network Model

The scenario assumes an attack on the workstation, node #42, which is linked to the remote PA server via connecting components #37, #36, #35 (in this order). The PA server holds the database (DB) of pre-recorded messages to be used for announcements in the airport. Another node, the remote station, node #41, directly depends on the server and is also connected to it in the same manner (see Fig. 3(b)). The scenario is divided into two simulations, i.e. (i) when no mitigation option is applied during the scenario and (ii) when the PA server

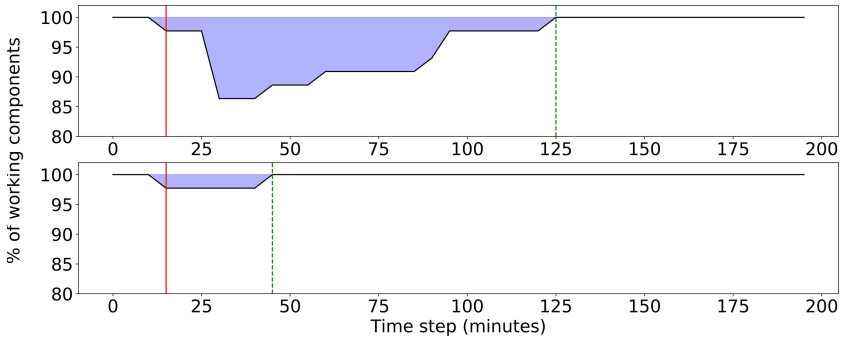


Fig. 4. Performance of the network during the test scenario. The vertical solid red lines shows the impact on the workstation and the vertical dashed green line shows the full recovery. The blue area above the curve represents the impact and the system's resilience. **Top:** No mitigation option. **Bottom:** With mitigation option. (Color figure online)

has installed a software for mitigation which identifies unauthorized requests for modification in the DB.

In absence of this mitigation measure, the workstation is attacked at $t = 15$ mins in the simulation. This is shown in the upper graphic in Fig. 4 as the red vertical line. With the access to the workstation, the server credentials are stolen and a faulty entry is updated in the DB. The attack on the DB is placed in the simulation at $t = 30$ mins. As a result, all the connected components and the PA amplifiers (node #39) are assumed to be compromised and as an impact the employees and the passengers are supposed to evacuate the airport building. It is assumed that it takes 30 min to identify and clear the workstation of the malware and another 30 min to restore the server to the previous working stage. As soon as the server is restored, all the connected components are considered to be working as expected in the next time step (5 min later). The passengers take another 35 min from this stage to get to their normal operating stage (represented by the green vertical line in Fig. 4).

In presence of the mitigation technique, the workstation is attacked at $t = 15$ mins, but the server is not impacted. So, all the connected components are shielded from the attack and the airport operates in normal manner, but with reduced performance, as the workstation is not functioning properly (see the lower graphic in Fig. 4). It takes 30 min to restore the workstation and the performance of the system is back to 100% at $t = 45$ mins (green vertical line). These results are not very surprising but demonstrate the principal that applies in small networks with limited mitigation possibilities. The impact propagation and mitigation gets easily quite complex in larger networks which will be considered in the course of the project.

3.3 IPS Results Based on the ABM

Once the server and the adjacent nodes are compromised, it is assumed that the PA system can be manipulated. In this scenario, the evacuation of all passengers is announced which is represented in the network model by an effect on asset #44. However, the degradation and restoration of this node in the network can only be roughly assumed. At this point the hybrid model triggers the ABM to better understand and estimate the impact on the passengers. First, normal operation is simulated with the ABM, which is passengers entering the airport from the outside area, going to check-in and bag-drop in the landside area and moving through security to the airside area and gates. In the simulation, passengers are assumed to move from the gates directly to their planes. The airport layout and simulation under normal conditions is presented in Fig. 5.

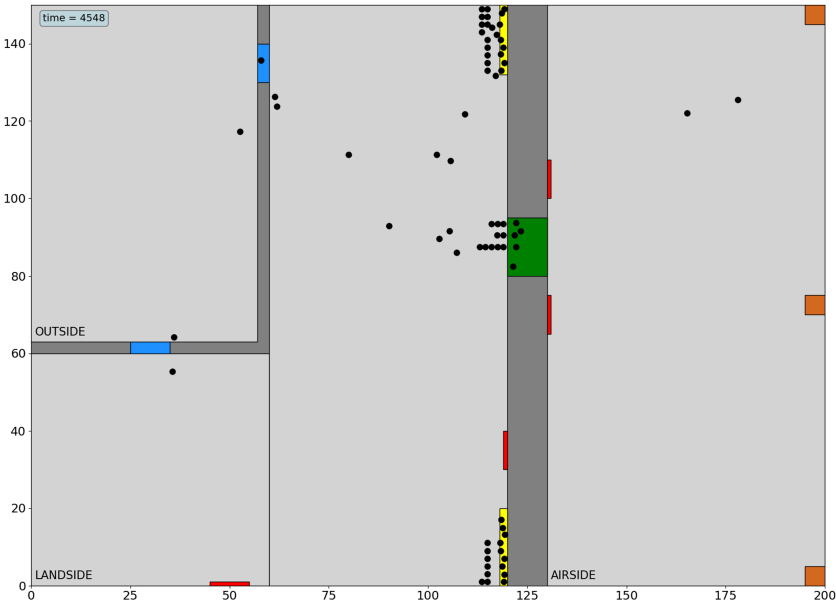


Fig. 5. Layout for the ABM containing doors (blue), check-in areas (yellow), security screening area (green), gates (brown) and FIDS monitors (red). Agents are presented as black dots. (Color figure online)

Second, at $t = 30$ mins during the simulation (vertical red line in Fig. 4), the evacuation is announced. Passengers in the airside area, will leave the airport towards the gates. All other passengers leave the airport towards the outside or parking area. The respective number of passengers in each area are shown in Fig. 6.

Note, that the number of passengers in the airside area is not given in Fig. 6 for readability of the graphic. From minute $t = 10$ on passengers arrive at the

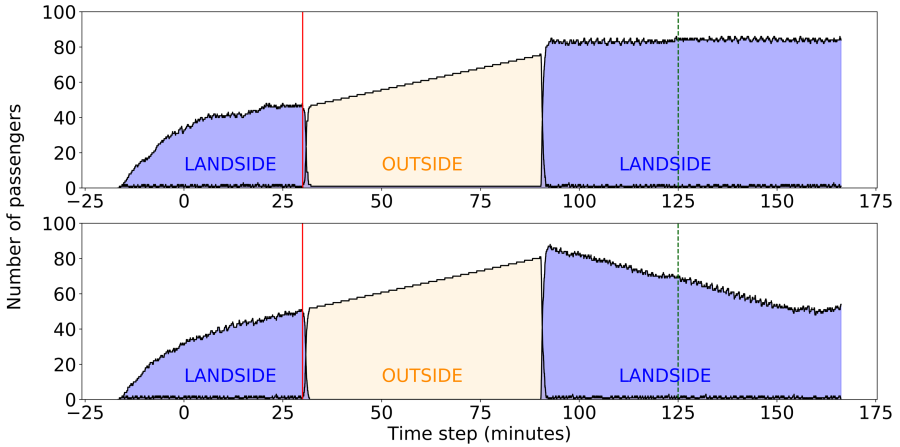


Fig. 6. Passenger behaviour in two of the airport areas, i.e. outside (parking area) and landside as a function of time during the test scenario. The area below the curve for passengers on the landside is given in blue (dark area) and for passengers on the outside in yellow (bright area). At $t = 30$ mins the evacuation is announced (vertical solid red line). At $t = 90$ mins the passengers are allowed to enter the airport again. For reference the time the network has fully restored after the attack without mitigation option is given as vertical dashed green line at $t = 125$ mins. **Top:** No mitigation option. **Bottom:** With mitigation option: The time for passing the security check was reduced assuming more desks have been opened. (Color figure online)

airside and converge to a relatively constant number of around 5 passengers. During the evacuation they leave to the gates and after the reopening of the airport the number rises again to around 5 passengers. Further, the time of the simulation starts at $t = -16$ mins representing a correcting factor to align the time of the announcement of the evacuation between the network model and the ABM at $t = 30$ mins. In the ABM the evacuation was triggered 16 min later than in the network model as it needs some time to converge. This needs to be taken into account when further developing the hybrid model.

It is shown in the upper plot of Fig. 6 that the original number of passengers before the attack happened, is no longer reached because a crowd of people is entering the airport after the evacuation and additional passengers arrive at the airport. It is observed that the current layout of the airport and the respective number of security check desks is not sufficient to handle the large number of passengers. Therefore, a mitigation option has been introduced (see Fig. 6, lower plot). The treatment time at security check desks is reduced which is assumed to be equivalent to opening additional desks. The number of passengers after the evacuation again increases but the additional capacities at the security check helps to normalize the number of passengers landside at around $t = 150$ min.

Finally, we observe for the small airport layout and the basic principals in use that attack situations impact the airport processes and that normal operation

is interrupted. The test scenario shows how the passengers gather in the outside area which represents the transformation of the attack and the corresponding impact from cyber to physical. Further, the crowd of passengers forming in the parking area might be an easy target for additional physical attacks. In comparison to the results using the network model and the assumptions made for restoration of nodes, it is observed that only with mitigation options in place after the evacuation, a restoration to normal operation is possible even in this very simple example. The IPS will allow for a more complex infrastructure to quantitatively describe the impact taking into account resilience principals.

4 Conclusion

Small changes in one place, to one object, can proliferate and spread throughout connected networks causing devastating and seemingly unpredictable large changes elsewhere. Critical infrastructure is fundamental to protect and yet is quite vulnerable to these kinds of effects due to its complexity and interconnect-edness, and to the fact that it is often sought after as a target for attacks because of its potentially fatal repercussions. Therefore, having meaningful and accurate modeling of how particular threats and risks can propagate through these critical infrastructure systems would greatly enhance the stakeholders' ability to better determine where countermeasures would be most effective in limiting the effects, should such an attack occur.

The described SATIE toolkit collects live data from airport systems, which is then analysed by threat prevention and detection systems, which can then elicit an event or even an alarm when a potential vulnerability is being exploited. The centralized user interface allows the Airport Operation Center personnel to have an easy overview of the situation, not requiring inquiring with multiple systems and having to manually determine if suspicious activity is occurring. The two described threat prevention and detection tools address two different subsets of threats: the BIA assess cyber-threats and how they affect the performance of business processes, while the ISP analyses both physical and cyber-threats and how they impact assets negatively. The combination of the two ensures the full coverage of assets and processes within an organization, to both physical and cyber-threats.

Through the test cases of an attack on the PA system, demonstrating not only how the cyber-attack can turn into a physical attack to people gathered outside of the airport, also revealed how it can influence other systems and processes of the organization (i.e. the check-in counters and security check desks). A hybrid approach consisting of a network representation of the airport and an ABM was employed to simulate the processes during the attack on the involved airport systems. The simulations presented here are a first step in the development of the hybrid model for the IPS in SATIE. The network model represents the top-down view based on assumptions about connectivity of nodes and repair times whereas the ABM enables to observe what will happen with only passenger attributes being defined, representing the bottom-up perspective. Further work

involves that the network model and ABM need to be aligned in time and the communication between the tools needs to be automated. The ABM specifically is still lacking some parts of human behaviour such as e.g.. group- and panic dynamics. The network model in particular needs to be extended with further airport systems and more mitigation options need to be defined. Finally, the model will be designed to be applicable to other critical infrastructures and the respective impact analysis.

Complex systems are ever-more present in our world as systems and people become even-more connected. When the functioning of these systems is instrumental in upholding and supplying our society with its necessities (e.g. water, electricity, food, movement, etc.), then a more holistic and complex threat modelling is necessary to better predict, understand, and then mitigate its effects. The toolkit used in SATIE aims to clear the sky for future endeavours of improving cyber- and physical security around the world.

Acknowledgement. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 832969. This output reflects the views only of the author(s), and the European Union cannot be held responsible for any use which may be made of the information contained therein. For more information on the project see: <http://satie-h2020.eu/>.

References

1. Argauer, B.J., Yang, S.J.: Vtac: virtual terrain assisted impact assessment for cyber attacks. In: Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008, vol. 6973, p. 69730F. International Society for Optics and Photonics (2008)
2. Asgari, H., et al.: Provisioning for a distributed ATM security management: the gamma approach. *IEEE Aerosp. Electron. Syst. Mag.* **32**(11), 5–21 (2017)
3. de Barros Barreto, A., Costa, P., Hieb, M.: Cyber-argus: Modeling c2 impacts of cyber attacks. Technical report george mason univ fairfax va center for excellence in command control (2014)
4. Cao, C., Yuan, L.-P., Singhal, A., Liu, P., Sun, X., Zhu, S.: Assessing attack impact on business processes by interconnecting attack graphs and entity dependency graphs. In: Kerschbaum, F., Paraboschi, S. (eds.) DBSec 2018. LNCS, vol. 10980, pp. 330–348. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95729-6_21
5. Eurocontrol: Airport CDM. Implementation Manual (2006)
6. Eurocontrol: Airport collaborative decision-making (A-CDM) implementation manual (2017). <https://www.eurocontrol.int/publication/airport-collaborative-decision-making-cdm-implementation-manual>
7. Genge, B., Kiss, I., Haller, P.: A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures. *Int. J. Crit. Infrastruct. Prot.* **10**, 3–17 (2015)
8. Grimaila, M.R., Fortson, L.W.: Towards an information asset-based defensive cyber damage assessment process. In: 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, pp. 206–212. IEEE (2007)

9. Günther, Y., et al.: Total airport management (operational concept and logical architecture) (2006)
10. Hiermaier, S., Hasenstein, S., Faist, K.: Resilience engineering-how to handle the unexpected. In: 7th REA Symposium, p. 92 (2017)
11. Jakobson, G.: Extending situation modeling with inference of plausible future cyber situations. In: 2011 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp. 48–55. IEEE (2011)
12. Jakobson, G.: Mission cyber security situation assessment using impact dependency graphs. In: 14th International Conference on Information Fusion, pp. 1–8. IEEE (2011)
13. Kasereka, S., Kasoro, N., Kyamakya, K., Goufo, E.F.D., Chokki, A.P., Yengo, M.V.: Agent-based modelling and simulation for evacuation of people from a building in case of fire. *Proc. Comput. Sci.* **130**, 10–17 (2018)
14. Khalili, A., Michalk, B., Alford, L., Henney, C., Gilbert, L.: Impact modeling and prediction of attacks on cyber targets. In: *Cyber Security, Situation Management, and Impact Assessment II; and Visual Analytics for Homeland Defense and Security II*. vol. 7709, p. 77090M. International Society for Optics and Photonics (2010)
15. Kheir, N., Mahjoub, A.R., Naghmouchi, M.Y., Perrot, N., Wary, J.P.: Assessing the risk of complex ICT systems. *Ann. Telecommun.* **73**(1–2), 95–109 (2018). <https://doi.org/10.1007/s12243-017-0617-0>
16. Kim, A., Kang, M.H., Luo, J.Z., Velasquez, A.: A framework for event prioritization in cyber network defense. Technical report naval research lab Washington DC center for high assurance computing systems (2014)
17. Kim, Y.-G., Lee, T., In, H.P., Chung, Y.-J., Kim, I.J., Baik, D.-K.: A probabilistic approach to estimate the damage propagation of cyber attacks. In: Won, D.H., Kim, S. (eds.) *ICISC 2005*. LNCS, vol. 3935, pp. 175–185. Springer, Heidelberg (2006). https://doi.org/10.1007/11734727_16
18. Kotenko, I., Chechulin, A.: A cyber attack modeling and impact assessment framework. In: 2013 5th International Conference on Cyber Conflict (CYCON 2013), pp. 1–24. IEEE (2013)
19. Kravari, K., Bassiliades, N.: A survey of agent platforms. *J. Artif. Soc. Soc. Simul.* **18**(1), 11 (2015)
20. Lange, M., Krotofil, M., Möller, R.: Mission impact assessment in power grids. In: *NATO IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact*. Istanbul, Turkey: Information Systems Technology Panel (2015)
21. Liu, C., Singhal, A., Wijesekera, D.: A layered graphical model for mission attack impact analysis. In: 2017 IEEE Conference on Communications and Network Security (CNS), pp. 602–609. IEEE (2017)
22. Mavrakis, C.: Passive asset discovery and operating system fingerprinting in industrial control system networks. Wayback archive, pp. 840171-1 (2015). <http://web.archive.org/web/20190307110951/>. <https://pure.tue.nl/ws/files/46916656/840171-1.pdf>
23. Motzek, A., Möller, R., Lange, M., Dubus, S.: Probabilistic mission impact assessment based on widespread local events. In: *NATO IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact* (2015)
24. Musman, S., Tanner, M., Temin, A., Elsaesser, E., Loren, L.: Computing the impact of cyber attacks on complex missions. In: 2011 IEEE International Systems Conference, pp. 46–51. IEEE (2011)

25. Musman, S., Temin, A.: A cyber mission impact assessment tool. In: 2015 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1–7. IEEE (2015)
26. Musman, S., Temin, A., Tanner, M., Fox, D., Pridemore, B.: Evaluating the impact of cyber attacks on missions. In: Proceedings of the 5th International Conference on Information Warfare and Security, pp. 446–456 (2010)
27. Naymik, M.: Cleveland breaks silence on airport ransomware attack (2019). www.govtech.com
28. Noel, S., Harley, E., Tam, K.H., Limiero, M., Share, M.: Cygraph: graph-based analytics and visualization for cybersecurity. In: Handbook of Statistics, vol. 35, pp. 117–167. Elsevier (2016)
29. Noel, S., et al.: Analyzing mission impacts of cyber actions (AMICA). In: NATO IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact (2015)
30. U.S. Office: Safe skies for tomorrow: Aviation safety in a competitive environment. University of California, Congress of the U (1988)
31. Orojloo, H., Azgomi, M.A.: A method for evaluating the consequence propagation of security attacks in cyber-physical systems. *Future Gener. Comput. Syst.* **67**, 57–71 (2017)
32. Othman, N.B., Legara, E.F., Selvam, V., Monterola, C.: Simulating congestion dynamics of train rapid transit using smart card data. *Proc. Comput. Sci.* **29**, 1610–1620 (2014)
33. Ou, X., Govindavajhala, S., Appel, A.W.: Mulval: a logic-based network security analyzer. In: USENIX Security Symposium, vol. 8, pp. 113–128. Baltimore, MD (2005)
34. Owusu, A., Mohamed, S., Anissimov, Y.: Input-output impact risk propagation in critical infrastructure interdependency. In: Proceedings of the International Conference Computing in Civil and Building Engineering (2010)
35. Papenfuss, A., Carstengerdes, N., Günther, Y.: Konzept zur Kooperation in Flughafen-Leitständen. In: 57. FAS DGLR L6.4 Anthropotechnik, 25.-26.11.2015, Rostock (2015)
36. Piekert, F., Carstengerdes, N., Suikat, R.: Dealing with adverse weather conditions by enhanced collaborative decision making in a TAM APOC. In: Air Traffic Management and Systems IV: Selected Papers of the 6th ENRI International Workshop on ATM/CNS (EIWAC2019) Air Traffic Management and Systems - IV Lecture Notes in Electrical Engineering, October 2019, in press
37. Porras, P.A., Fong, M.W., Valdes, A.: A mission-impact-based approach to INFOSEC alarm correlation. In: Wespi, A., Vigna, G., Deri, L. (eds.) RAID 2002. LNCS, vol. 2516, pp. 95–114. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36084-0_6
38. Sawilla, R.E., Ou, X.: Identifying critical attack assets in dependency attack graphs. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 18–34. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88313-5_2
39. Stergiopoulos, G., Kotzanikolaou, P., Theocharidou, M., Gritzalis, D.: Risk mitigation strategies for critical infrastructures based on graph centrality analysis. *Int. J. Crit. Infrastruct. Prot.* **10**, 34–44 (2015)
40. Stroeve, S.H., Bosse, T., Blom, H.A., Sharpanskykh, A., Everdij, M.H.: Agent-based modelling for analysis of resilience in ATM. Proceedings of the Third SESAR Innovation days. Stockholm (Sweden), November 2013

41. Suikat, R., Schier-Morgenthal, S., Carstengerdes, N., Günther, Y., Lorenz, S., Piekert, F.: What-if analysis in total airport management. In: Stanton, N. (ed.) AHFE 2020. AISC, vol. 1212, pp. 517–523. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50943-9_65
42. Sun, X., Singhal, A., Liu, P.: Who touched my mission: towards probabilistic mission impact assessment. In: Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense, pp. 21–26 (2015)
43. Sun, Y., Wu, T.Y., Liu, X., Obaidat, M.S.: Multilayered impact evaluation model for attacking missions. *IEEE Syst. J.* **10**(4), 1304–1315 (2014)
44. Van Dam, K.H., Nikolic, I., Lukszo, Z.: Agent-Based Modelling of Socio-Technical Systems, vol. 9. Springer, Dordrecht (2012). <https://doi.org/10.1007/978-94-007-4933-7>
45. Van Der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, vol. 2. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
46. Wilensky, U., Rand, W.: An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo. MIT Press, Cambridge (2015)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Comparative Analysis of Emulated and Real IEC-104 Spontaneous Traffic in Power System Networks

C.-Y. Lin^(✉) and Simin Nadjm-Tehrani

Department of Computer and Information Science, Linköping University,
Linköping, Sweden
{chih-yuan.lin,simin.nadjm-tehrani}@liu.se

Abstract. Supervisory and Data Acquisition (SCADA) systems control and monitor modern power networks. As attacks targeting SCADA systems are increasing, significant research is conducted to defend SCADA networks including variations of anomaly detection. Due to the sensitivity of real data, many defence mechanisms have been tested only in small testbeds or emulated traffic that were designed with assumptions on how SCADA systems behave. This work provides a timing characterization of IEC-104 spontaneous traffic and compares the results from emulated traffic and real traffic to verify if the network characteristics appearing in testbeds and emulated traffic coincide with real traffic. Among three verified characteristics, two of them appear in the real dataset but in a less regular way, and one does not appear in the collected real data. The insights from these observations are discussed in terms of presumed differences between emulated and real traffic and how those differences are generated.

Keywords: SCADA · Traffic characterization · IEC-104 · Timing analysis

1 Introduction

A modern power distribution system is a cyber-physical system comprising a network of geographically distributed devices and processes. Supervisory Control and Data Acquisition (SCADA) systems are used to control and monitor the network and processes. The emergence of attacks targeting SCADA systems and the controlled processes makes SCADA security a pressing issue [6, 8, 18]. Research on defending SCADA networks against such intrusions requires traffic datasets to develop, evaluate, and compare different defence mechanisms. Due to the secrecy of SCADA systems as part of critical infrastructure, real traffic is not openly available for the research community. Where data sharing from a large testbed is available, for example, in the case of iTrust testbed data from the EPIC and SWaT testbeds [11], the packet flows have been generated from one of the many possible SCADA protocols (CIP, GOOSE, MMS). Unfortunately,

© The Author(s) 2021

H. Abie et al. (Eds.): CPS4CIP 2020, LNCS 12618, pp. 207–223, 2021.

https://doi.org/10.1007/978-3-030-69781-5_14

many defence mechanisms that need to be tested with other protocols, e.g. Intrusion Detection Systems (IDS) for IEC-60780-5-104 and IEC-61850 protocols are tested on a small-scale testbed [24] or with simulated/emulated datasets [13, 25]. Hence, how reliable are the simulated/emulated datasets has become a crucial question for the development of SCADA-specific IDS.

IEC-60780-5-104 (hereafter referred to as IEC-104) protocol is an international standard of data transmission between an electric power SCADA center and outstations over TCP/IP and widely used in Europe [7]. Unlike other SCADA protocols, such as Modbus, comprising mainly request-response communications triggered by a polling mechanism from the SCADA center, IEC-104 traffic contains a great deal of spontaneous communications [21]. In the spontaneous communication mode, field devices in the outstations initiate messages when the monitored measurements of process variables change or fall outside a predefined range.

Most of the research on IDS for SCADA networks model the request-response communications [5, 12, 23, 25] but fewer solutions are available on spontaneous communications due to the lack of understanding of spontaneous traffic. Although Lin and Nadjm-Tehrani [19] applied pattern mining techniques based on Probabilistic Suffix Tree (PST) to two emulated IEC-104 datasets in order to discover timing patterns, study of IEC-104 traffic characteristics is still in its infancy. Detailed knowledge of how spontaneous traffic behaves in a real network is necessary for the development of SCADA-specific IDS and improved SCADA network simulations/emulations.

This work first reviews potential flow-based characteristics suggested in literature, and then provides an empirical study of spontaneous traffic generated in a real-world utility with respect to the reviewed characteristics. It also performs a comparison with the emulated traffic used in previous works [19, 24].

Our primary contribution in this study is a detailed timing-based characterization of IEC-104 spontaneous traffic collected from a real power station. The results can be a first step to arrive at a traffic model when deciding about features and modeling approaches for anomaly detection, expanding the possibility of testing those IDS so far only tested with simulated/emulated traffic. The secondary contribution is the outcome of the comparison between behaviour of traffic from real and emulated power networks. It suggests the emulated traffic generated in earlier works may not be realistic enough. Some modifications could be made in those testbeds to improve the usability of the datasets generated for SCADA security research.

2 Related Work

To guide and facilitate intrusion detection research for SCADA systems, network analysis and characterization of SCADA traffic has been an active research area. Most of the works focus on characterizing high level attributes such as bandwidth, port number, and the number of protocols. Barbosa et al. [3] conducted a comparative analysis of SCADA traffic from water treatment facilities and normal IT traffic. This study found that SCADA traffic lacks traffic patterns that

are used to model IT traffic. The results indicate the need for SCADA-specific modeling approaches for anomaly detection. In separate work, Barbosa et al. [4] conducted another comparative analysis of SCADA traffic and SNMP traffic and found that both of them exhibit periodicity. Jung et al. [14] characterized SCADA traffic of a power station with variations in frame sizes, TCP connects, port number, and initial sequence number. In a later work, Formby et al. [9] further studied the initial sequence number attribute in the same traffic and found predictable patterns.

As more intrusion detection studies focus on protocol-specific models, more protocol-specific attributes are explored. Formby et al. [10] characterized DNP3 power grid traffic and examined a few common hypotheses such as stable traffic volume and regularity of DNP3 poll time. Mai et al. [22] characterized IEC-104 power grid traffic regarding the number of occurrences of different instructions and the directions and magnitude of IEC-104 flows, where the flow is defined by the source and destination addresses with the 4-tuple $\langle srcIP, srcPort, dstIP, dstPort \rangle$. Lin and Nadjm-Tehrani [19] characterized emulated IEC-104 spontaneous traffic with a focus on the predictability of timing patterns.

The current paper examines three characteristics of IEC-104 spontaneous traffic using data collected from a real power station. Two of the characteristics were proposed or observed in earlier work [19]. One characteristic observed in the emulated datasets of the earlier work is confirmed when analysing the real traffic while another characteristic is shown to exist only in the emulated datasets. The third characteristic will be discussed in more detail in Sect. 5. The confirmed characteristics have already guided the development of an anomaly detector [20] for IEC-104 spontaneous traffic.

3 Background on the IEC-60780-5-104 Protocol

The IEC-104 protocol is widely used in modern SCADA systems to control and monitor geographically dispersed processes, especially for power station automation. The main advantage of IEC-104 is that it connects a control station (Master Terminal Unit, MTU) and one or more substations (Remote Terminal Unit, RTU) via a standard TCP/IP network. The IEC-104 protocol defines two directions for data transmission: (1) monitor direction, the direction of transmission from an RTU to the MTU and (2) control direction, the direction of transmission from the MTU to an RTU. The monitored data that are transmitted from an RTU to the MTU are also known to be sourced at *Monitor Points*. Every monitor point is configured to locate in a specific address in an RTU device and will be identified by the address at application level.

In order to improve the communication efficiency, the IEC-104 protocol enables not only the MTU to poll for monitor points periodically but also the RTUs to generate spontaneous events about data changes at monitor points. The following explains the important terminologies of IEC-104 protocols used in this study.

- **Information Object:** A piece of data containing information from a monitor point such as measured value and time tag. A spontaneous packet may carry more than one information object.
- **Information Object Address (IOA):** The address and identification of a monitor point where an information object is issued from.
- **Cause of Transmission (COT):** A field in the application layer to specify the type of the packet. A spontaneous packet is noted as SPONT.
- **Type IDentification (TID):** A field in the application layer to specify the type of the monitor point(s) in a packet. The most common data type is Monitored MEasured point in different formats such as M_ME_NA (normalized value) and M_ME_NB (scaled value). This data type contains a measured value from a certain IOA. The system administrator needs to set a deadband (i.e., a range) for each monitored measured point and the RTU will send a spontaneous event when the value falls outside the deadband. In addition, Monitored Single Point (e.g., M_SP_NA) and Monitored Double Point (e.g., M_DP_NA) specifies the state of a point, such as a switch or circuit breaker. For these points, the RTU will send a spontaneous event whenever the value changes.

4 The Studied Datasets

This section first presents an overview of the examined datasets. Then, it describes how the datasets are collected and preprocessed for the analysis.

In this study we analyze three different IEC-104 datasets: two emulated power network datasets and one dataset collected in a real power station at a utility.

- **SmallTB-RTUx:** SmallTB-RTUx dataset is collected from a small testbed with real commercial hardware maintained by the Royal Institute of Technology (KTH). The setup contains four RTUs, one switch, and a user terminal machine. The data is collected through a mirroring port on the switch. Traces from two out of the four RTUs are used in previous work [24] and available to us. For the sake of consistency, we follow the naming scheme of the RTUs in the previous work and name the traffic as SmallTB-RTU1 and SmallTB-RTU4.
- **VirtualTB:** VirtualTB dataset is collected in a virtual testbed developed within the RICS project [1]. The testbed consists of an office network and a SCADA network. The setup of SCADA network contains some twenty substations, two SCADA servers, and a virtual WAN (Wide Area Network) with 15 nodes connecting the control room and the substations. The data is collected at the communication gateway to the WAN on the main SCADA server. There are no network delays or traffic congestion in this virtual network. Traces of one substation with an emulated RTU is used in earlier work [19] and available to us.
- **Real-RTUx:** Real-RTUx dataset is collected from a real power facility. The SCADA network contains several RTUs communicating with the SCADA

server with different protocols. Among them, there are two RTUs that communicate through IEC-104 which are included in earlier work [20] and this study. The traffic is collected by the utility personnel running our data collection software in their operation site, here named Real-RTUA and Real-RTUB.

To perform our timing-based characterization, we need to transform the collected PCAP traces into desired formats: *event sequence* and *time series* of flows. The process includes the following steps. (1) It starts by identifying spontaneous packets with COT = SPONT. (2) For all the spontaneous traffic, the process separates them into unique flows, where a flow is defined by the tuple $\langle RTU(SrcIP), IOA, TID \rangle$. Note that a packet may contain multiple information objects and thus multiple IOA but only one TID as stated in Sect. 3. (3) The next step in preprocessing that forms an event sequence for each flow and records the PCAP timestamps as event arrival times for timing analysis. (4) Finally, the process transforms each event sequence to time series by calculating the number of events per some configurable interval of time.

No matter in which format, we split the events per flow into 10 parts, use the first part for learning and the remaining nine parts to evaluate the stability of the attributes. Table 1 shows an overview of the studied datasets with the associated throughput for each RTU. The TID column lists instructions found in the traffic from each RTU, and the last two columns present the number of flows found and used. In the previous works, the flows with low event rates were not included to avoid biased learning results. This study too excludes the flows with an event rate of fewer than 0.3 events per hour since these flows contain only sporadic events that apparently show very different attributes.

Table 1. Overview of time series obtained from the datasets.

Dataset	Duration	Throughput (#events/hr)	TID	# Flows	# Used Flows
SmallTB-RTU1	12 days	19182	M_ME_NA	4	4
SmallTB-RTU4	12 days	10712	M_ME_NA	3	3
VirtualTB	6 days	2433	M_ME_NA	15	12
Real-RTUA	30 days	13981	M_ME_NA	21	19
			M_DP_TB	8	0
			M_SP_TB	3	0
Real-RTUB	30 days	401	M_ME_NA	16	14

5 Data Characterisation Methods

This section describes flow-based characteristics observed in the literature and the way we examine the characteristics in this paper.

5.1 A Review of Potential Characteristics

This subsection briefly reviews three hypotheses about spontaneous traffic timing characteristics observed from earlier papers. [2, 15, 16, 19].

- **Spiky Distribution.** Lin and Nadjm-Tehrani’s work [19] studies the timing predictability of the spontaneous events based on an assumption, namely that inter-arrival time distribution for events is spiky, without verifying it. A spiky distribution means the probability of some inter-arrival time to be present is higher than others as shown in Fig. 1(a).
- **Timing Predictability.** Timing predictability analysis addresses the research question: can we predict when the next spontaneous event will come by learning the historical timing data? In earlier work [19] it is shown that in 11 out of 14 tested data sequences, there exists evidence of sequential patterns. Hence, there is a hypothesis that historical data provides timing predictability even in spontaneous traffic.
- **Correlation.** There are a number of works that model sensor signals with clustering techniques based on correlations between sensors [2, 15, 16]. As stated in Sect. 3, sensor measurements of the processes and spontaneous events have a cause-effect relationship. The results indicate that sensors in SCADA systems are correlated. Therefore, we propose the correlation hypothesis that posits spontaneous events from different IOAs (i.e., connecting to different sensors) could be also correlated.

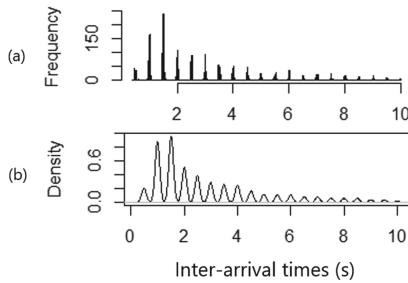


Fig. 1. Distribution of inter-arrival times from a inter-arrival time sequence in the emulated VirtualTB dataset [19]: (a) Histogram for inter-arrival times $\delta_i \leq 10$ s. (b) The smoothed version of the sequence, bandwidth = 0.008

5.2 A Review of Characterization Methods

This subsection describes the known methods that will be used to analyze the characteristics.

Spiky Distribution. Lin and Nadjm-Tehrani [19] propose an algorithm to learn the areas with high probability to have spikes as *legitimate areas*. The algorithm finds the relatively low point pairs on the smoothed curves of histogram as

shown in Fig. 1(b), where the smoothing is done by the kernel density estimation method with a bandwidth parameter that decides the smoothness level. The low point pairs are considered as the boundaries of legitimate areas.

Timing Predictability. Following application of the algorithm mentioned in the previous paragraph, the same work separates inter-arrival times into groups, with one spike per group. The method translates the numeric event inter-arrival time sequence observed in a PCAP file (e.g., 3.15, 3.17, 0.51, 0.48) into a symbolic sequence by replacing each numeric inter-arrival time with its group symbol in the symbolic alphabet (e.g., aabb). Then, using the symbolic sequences, the method builds a PST for each flow in the learning phase and tests the predictability of the learned PST. In the testing phase, the method runs with a sliding window over the symbolic sequence. With a given window size (6 symbols in the mentioned study), the method queries the built PST for the next element that is most likely to happen as its prediction.

The mentioned work evaluates the timing predictability with predication accuracy and Kappa statistics [17]. With the resulting confusion matrix, the observed prediction accuracy P_0 is defined as:

$$P_0 = \frac{\sum_{i=1}^c n_{ii}}{N} \tag{1}$$

where N is the number of predictions performed in the testing phase, c is the number of possible symbols (i.e., number of rows/columns of the confusion matrix), and n_{jk} is the number of times the symbol k (ground truth) is predicted as j . The expected prediction accuracy by a random observer is:

$$P_e = \sum_{i=1}^c \left(\frac{n_{i+}}{N} \times \frac{n_{+i}}{N} \right) \tag{2}$$

where n_{i+} is the total number of times the symbol i appears in the testing data and n_{+i} is the total number of times any symbol is predicted as i . Kappa statistics is:

$$Kappa = \frac{P_0 - P_e}{1 - P_e} \tag{3}$$

A random observer is a pseudo observer who randomly picks up a value from the learned probability distribution of inter-arrival times. Kappa statistics compares the observed accuracy and expected accuracy. If our prediction model is similar to a random observer, the Kappa value will be around 0. On the other hand, if our prediction model and the testing data contains clear sequential patterns, the Kappa value will be close to 1.

Correlation. Spearman correlation coefficient (ρ) is a measure of the monotonic relationship of two time series. For any two time series $X^p = x - i^p, \dots, x_m^p$ and $X^q = x_i^q, \dots, x_m^q$, we have ranked time series $R(X^p) = R(x_1^p), \dots, R(x_m^p)$ and $R(X^q) = R(x_1^q), \dots, R(x_m^q)$, where the numeric values are replaced by their rank

in the sorting. Then, the Spearman correlation coefficient is:

$$\rho_{pq} = \frac{COV(R(X^p), R(X^q))}{\sigma_{R(X^p)}\sigma_{R(X^q)}} \quad (4)$$

where $COV(R(X^p), R(X^q))$ denotes the covariance of the ranked time series and $\sigma_{R(X^p)}$ and $\sigma_{R(X^q)}$ are the standard deviations.

The correlation coefficient values are between -1 and 1 . The values close to 1 or -1 indicate a strong relation between the two time series in the same or opposite direction, and values close to 0 indicate a low association between time series.

5.3 Methods and Parameter Choices for the Comparative Analysis

The comparative analysis in this paper aims to not only understand whether the above characteristics exist in the three datasets from Sect. 4 but also how persistent they are. This subsection elaborates the workflows and parameter choices for the comparative analysis.

Spiky Distribution. The analysis first illustrates and categorizes the Probability Density Function (PDF) of inter-arrival times. Then, it tests whether the characteristics are stable and persistent. In this paper we will learn the legitimate areas with a high probability to have spikes as shown in Sect. 5.2. The major difference between the implementation in this paper and the earlier work is the limitation of maximum number of spikes. Our implementation can find as many spikes as possible while the previous work has a limit on number of spikes set as 12, which means only the twelve largest spikes can be modeled.

Further, we test if the learned results remain in the testing part of the data using the metric of *unknown data rate*. The unknown data rate (UDR) is defined as:

$$UDR = \frac{n_x}{N} \quad (5)$$

where n_x denotes number of observations in the testing data that do not locate in any of the learned legitimate areas, and N denotes number of observations in the testing data. Thus, the lower the UDR, the higher the degree of stability of the spiky distribution.

Timing Predictability. The timing predictability comparative analysis applies the proposed method in Sect. 5.2 to all three datasets from Sect. 4. There are two different enhancements compared to the earlier work. First, based on the results of spiky distribution analysis, we can get as many symbolic alphabets as possible from an inter-arrival time sequence. This change makes the PST sequence model more accurate. Compared to the earlier work, the inter-arrival times located in the spikes that are smaller than the twelfth largest spike won't be bundled together. Second, this study uses one-tenth of the datasets as learning data, which contains more observations than the 2-hour (short) learning

data in the earlier work. This change enables the PST model to discover longer sequential patterns if there are any.

Correlation. We calculate how many pairs of time series are significantly correlated by computing p-values for null-hypothesis $H_0 : \rho_{pq} = 0$, and compare the correlation rates between different datasets. The bin size of the time series in this study is 1 min. With the resulting correlation rate of flows, we further examine how the spontaneous traffic flows are correlated with each other using dendrograms and if there's any change on the dendrograms for learning period and testing period.

6 Observations and Discussions

This section summarises the results of our comparisons between the emulated and real data sets, using the above hypotheses and applied methods.

6.1 Spiky Distribution

A few common patterns appear in the PDFs of event inter-arrival times for each flow from the emulated datasets. These patterns contain multiple spikes with different heights and weights that are distributed as different shapes of curves. Figure 2(a) presents a centered pattern. The pattern contains one major spike and a few minor spikes located around it. Figure 2(b) is a long-tail pattern. The spikes are distributed with a long-tail. Figure 2(c) presents the multiple centered pattern in a long-tail distribution. Figure 2(d) presents spikes in a dispersed unknown distribution.

All the PDFs presented in emulated datasets show roughly equal spacing between the spikes and all the flows from emulated datasets have a constant size of gaps between spikes. The gap size between spikes in emulated datasets is the update rate at which the emulated RTUs update the information of simulated processes. If the value of monitored points changes or exceeds a predefined range when an RTU updates the information, the RTU sends a spontaneous event. VirtualTB has a gap size of 5 s and SmallTB has a gap size of 0.5 s.

As expected, most of the flows from the real traffic present spiky inter-arrival time distributions and equal spacing between the spikes. However, not all of the flows have the same gap size even if they are from the same RTU. The real data has the lowest gap size of 0.625 s and the largest gap size of 7.5 s. Moreover, in this dataset, traffic from different RTUs exhibit very different timing characteristics. Table 2 presents the inter-arrival time distribution type and UDR in column *Spiky Distribution*. Most of the flows issued by RTUA present centered patterns, whereas, in 10 out of 14 flows issued by RTUB, we did not find clear spikes as shown in Fig. 2(e). The flows without spikes have relatively low event rates (around 20 events per hour). We speculate that the real system monitors the processes with different granularities. Some are updated more often while some are not regularly updated.

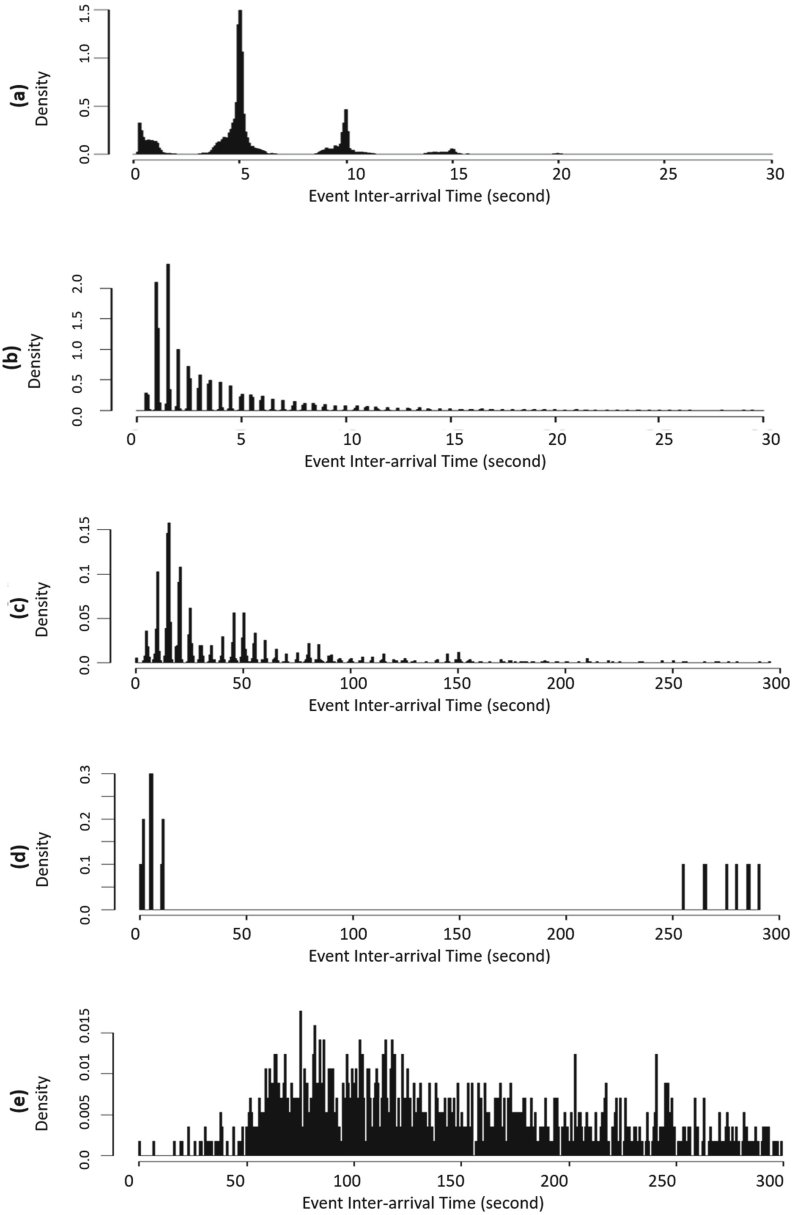


Fig. 2. Common patterns in the PDF of event inter-arrival times. (a) A centered pattern from VirtualTB, IOA 10091 (b) A long-tail pattern from SmallTB-RTU4, IOA 2 (c) A multi-center pattern from VirtualTB, IOA 10002. (d) A dispersed pattern from VirtualTB, IOA 10010. (e) No clear pattern from Real-RTUB, IOA 3018.

The flows with a resulting unknown data rate (see Eq. 5) below 3% are highlighted in gray. Except for the flows that have less than 23 events per hour, all the flows with spiky inter-arrival time distributions exhibit a low UDR, which means the learned characteristics are stable and persistent within the data collection period. After a manual examination of the flows having low event rate and showing higher UDR, we observe that there are not enough elements in the learning period for the used algorithm to properly estimate the legitimate areas.

6.2 Timing Predictability

The timing predictability analysis results are presented in Table 2, the last two columns.

There are a few insights obtained in this analysis. First, as discovered in the earlier work [19], there are some flows in the emulated datasets that show evidence of the existence of sequential patterns. In 14 out of the 19 flows, we get a Kappa value that agrees on the existence of sequential patterns¹ (i.e., Kappa is not around 0). Among them, 9 Kappa values show slight agreement (Kappa values around 0.1) and 4 show medium agreement (Kappa values 0.3–0.4). However, most of the flows from the Real-RTUx datasets have a Kappa value around 0 and only four flows have a slight agreement on the existence of sequential patterns. We speculate that the underlying sequences found in the emulated datasets could be generated by the repeated workflow of the process simulation programs.

Second, a first look at prediction accuracy may provide an impression that real data have lower accuracy. However, if we only look into the flows containing sufficient elements for learning (i.e., rows highlighted in gray), prediction accuracy is more related to distribution type rather than the type of dataset. Among all the highlighted flows, centered patterns give better accuracy in predictability irrespective of evidence of sequential patterns or not. Most of the flows of centered distribution type have high accuracy and a low Kappa value because most of the intervals fall into the major spike². Long-tail patterns have barely predictability when the Kappa value is close to 0³, whereas they show higher prediction accuracy when there exists evidence of sequential patterns⁴. Multi-centered patterns have higher prediction accuracy when the distribution is closer to a centered distribution, i.e. most of the intervals fall into a few major spikes.

Third, compared with earlier work, our analysis gets improved accuracy for some flows from the emulated datasets⁵ due to the choice of learning parameters. The changes of the parameters include higher bandwidth for the kernel density

¹ SmallTB-RTU1 IOA: 1, 3, SmallTB-RTU4 IOA: 2, 4, VirtualTB IOA: 10002, 10005, 10011, 10013 10014, 10015, 10016, 10017, 10091, 10092.

² SmallTB-RTU1 IOA: 2, 4, SmallTB-RTU4 IOA: 3, Real-RTUA IOA: 3002, 3003, 3004, 3005, 3007, 3008, 3010, 3011, 3012, 3013, 3014, 3015, 3018, 3019, 3020.

³ Real-RTUB IOA: 3016, 3019.

⁴ SmallTB-RTU1 IOA: 1, 3, SmallTB-RTU4 IOA: 2, 4, VirtualTB IOA: 10002.

⁵ SmallTB-RTU1 IOA: 2, 4 and SmallTB-RTU4 IOA: 3.

Table 2. Analysis results for spiky distribution and timing predictability hypotheses. UDR stands for unknown data rate (Eq. 5).

Dataset	IOA	Event rate (#events/hr)	Spiky distribution		Timing predictability		
			Distribution Type	UDR (%)	Accuracy	Kappa	
SmallTB-RTU1	1	2384	long-tail	≈ 0	0.57	0.1	
	2	6955	centered	≈ 0	0.99	≈ 0	
	3	2875	long-tail	0.08	0.38	0.1	
	4	6968	centered	≈ 0	0.99	≈ 0	
SmallTB-RTU4	2	2053	long-tail	≈ 0	0.52	0.1	
	3	7024	centered	≈ 0	0.99	≈ 0	
	4	1095	long-tail	0.11	0.61	0.4	
VirtualTB	10002	61	long-tail	3.00	0.20	0.1	
	10005	128	multi-center	0.63	0.26	0.2	
	10010	3	dispersed	80.36	0.67	≈ 0	
	10011	763	centered	0.01	0.74	0.3	
	10012	23	multi-center	5.48	0.06	≈ 0	
	10013	4	dispersed	68.95	0.38	0.1	
	10014	56	multi-center	1.81	0.22	0.1	
	10015	12	multi-center	18.48	0.09	0.1	
	10016	57	multi-center	1.18	0.23	0.1	
	10017	12	multi-center	17.46	0.08	0.1	
	10091	642	centered	≈ 0	0.65	0.3	
	10092	671	centered	≈ 0	0.68	0.3	
	Real-RTUA	3002	394	centered	0.19	0.71	≈ 0
		3003	372	centered	0.02	0.65	≈ 0
		3004	414	centered	0.07	0.49	≈ 0
		3005	628	centered	0.03	0.79	≈ 0
		3007	425	centered	0.02	0.76	≈ 0
3008		372	centered	0.04	0.65	≈ 0	
3009		261	multi-center	0.14	0.32	≈ 0	
3010		683	centered	0.01	0.86	≈ 0	
3011		973	centered	0.09	0.78	0.1	
3012		1051	centered	0.24	0.82	≈ 0	
3013		1088	centered	0.24	0.82	≈ 0	
3014		1084	centered	0.14	0.82	≈ 0	
3015		902	centered	0.02	0.67	0.1	
3016		793	multi-center	≈ 0	0.52	≈ 0	
3017		886	multi-center	≈ 0	0.60	≈ 0	
3018		568	centered	0.19	0.88	≈ 0	
3019		1288	centered	0.01	0.86	≈ 0	
3020		1103	centered	0.01	0.73	≈ 0	
3021		697	multi-center	0.02	0.58	0.1	
Real-RTUB	3002	28	no pattern	3.59	0.01	≈ 0	
	3004	22	no pattern	5.93	≈ 0	≈ 0	
	3005	14	no pattern	10.86	≈ 0	≈ 0	
	3006	66	multi-center	0.73	0.09	0.1	
	3008	18	no pattern	6.87	0.01	≈ 0	
	3009	28	no pattern	3.55	0.01	≈ 0	
	3011	21	no pattern	6.11	0.01	≈ 0	
	3012	21	no pattern	4.37	0.01	≈ 0	
	3013	15	no pattern	9.52	≈ 0	≈ 0	
	3014	18	no pattern	6.43	≈ 0	≈ 0	
	3015	7	dispersed	18.86	0.05	≈ 0	
	3016	63	long-tail	0.72	0.02	≈ 0	
	3018	19	no pattern	6.37	≈ 0	≈ 0	
	3019	63	long-tail	0.83	0.02	≈ 0	

estimation, extended learning phase, and unlimited number of symbols for the PSTs as described in Sect. 5.3.

6.3 Correlation

With a p-value of 0.05, there are 86%, 89%, and 74% significantly correlated flows respectively within SmallTB, VirtualTB, and Real datasets. Figure 3 presents the dendrograms using Euclidean distance between observations/clusters based on the absolute correlation. They show the observations/clusters for SmallTB, VirtualTB, and Real datasets in learning (left side) and testing period (right side), respectively. The leaves are the flow IDs, the height stands for Euclidean distance and the dotted line is an example cut-off line that separates the flows into clusters. The dendrograms of VirtualTB dataset have the same structure for learning and testing data. That is, for every cut-off line in the learning dendrogram, one can find a corresponding cut-off line in the testing dendrogram that generates the same clustering results.

In the dendrograms for SmallTB and Real dataset, there are a few flows that jump from one group to another but the structure remains the same for most of the time. For example, the cut-off line for Real data generates 6 groups in both the learning and testing period. There are two highlighted groups G1 and G2 in both trees. Flow RTUA 3016 is included in G2 of the learning tree, but it moves to G1 of the testing tree in the testing period.

The results suggest that correlations between flows are complicated. A flow can be correlated with multiple flows and the magnitude of correlations between different flows may change from time to time. We speculate that the virtual testbed has fewer dynamical processes so that it exhibits overly stable relations between flows.

7 Conclusions

Due to the secrecy nature of SCADA traffic, lack of openly available datasets for intrusion detection research has been an open question. Many research efforts on intrusion detection systems in SCADA networks are tested with emulated or simulated datasets. This study examines three hypotheses about IEC-104 spontaneous traffic attributes that were proposed or observed in previous work with a comparison between emulated and real datasets. The results show that emulated datasets are prone to simple and regular patterns.

In the spiky distribution analysis, the emulated datasets exhibit a unified update rate of information that shows up as a unique gap size between spikes in the whole system. The real datasets, on the other hand, exhibit a wide variety of gap sizes in a system. Some of the flows even do not present a spiky inter-arrival time distribution.

In the predictability analysis, the emulated datasets exhibit evidence of underlying inter-arrival time sequences that make the timing of the next event

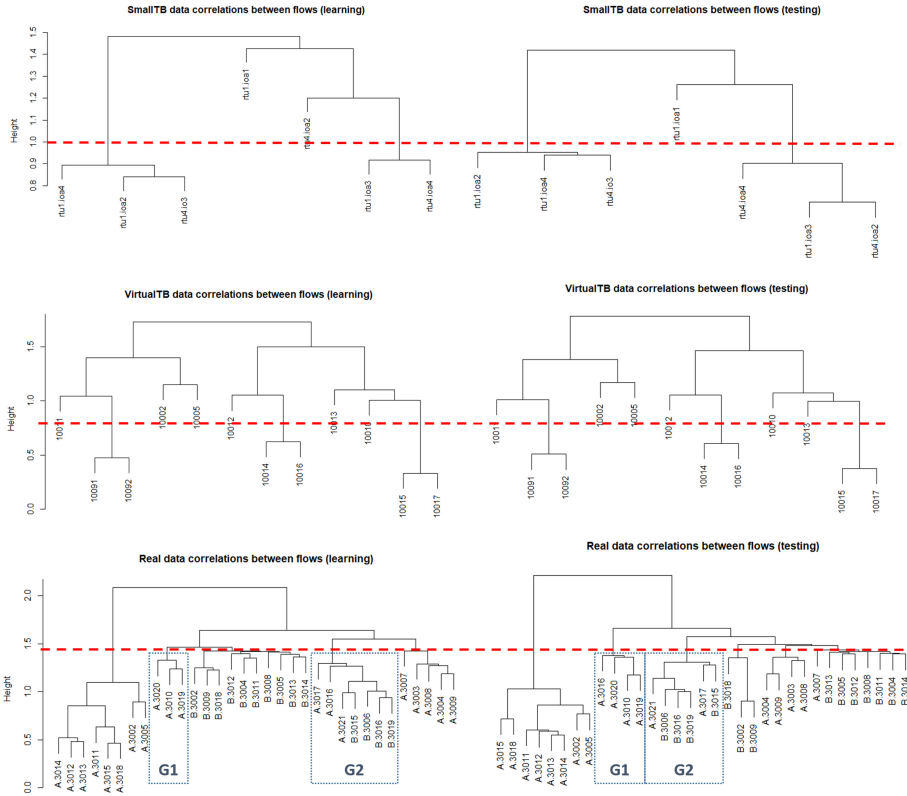


Fig. 3. Correlation dendrograms for learning and testing period. Top: the small emulated network, Middle: the RICS-el virtual network, Bottom: the real utility network.

predictable. However, the real dataset suggests little evidence of underlying sequences.

In the correlation analysis, both the emulated and real datasets indicate that traffic flows are intricately correlated. However, the correlations between flows seem to be less dynamic in emulated datasets.

The study of differences between emulated and real datasets ought to be a precondition for intrusion detection research, especially learning-based anomaly detection systems. The results in this paper show that traffic attributes that exist in emulated datasets may be not valid in real datasets. Therefore, it's crucial to select explainable features for anomaly detection systems when only emulated datasets are available for learning and testing. The simpler and more regular attributes can lead to overestimation of performance as well. This indicates room for improvement of emulated datasets, such as more detailed and complicated system configurations or adding random events to the process simulators.

One obvious future work is to find more attributes from different real datasets and a systematic approach to generate realistic synthetic datasets. The results in this study suggest the need to characterize the uncertainty of the selected features. Another way is to make sanitized real datasets openly available by applying traffic anonymization methods.

Acknowledgement. This work was completed within RICS: the research centre on Resilient Information and Control Systems (www.rics.se) financed by Swedish Civil Contingencies Agency (MSB). The authors would like to thank Swedish Defence Research Agency (FOI) for collaboration on RICS-el, our collaborators at Royal Institute of Technology (KTH), and our industrial partners for data collection.

References

1. Almgren, M., et al.: RICS-el: building a national testbed for research and training on SCADA security (Short Paper). In: Luijff, E., Žutautaitė, I., Hämmerli, B.M. (eds.) CRITIS 2018. LNCS, vol. 11260, pp. 219–225. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05849-4_17
2. Aoudi, W., Iturbe, M., Almgren, M.: Truth will out: departure-based process-level detection of stealthy attacks on control systems. In: Proceedings of the Conference on Computer and Communications Security. ACM (2018)
3. Barbosa, R.R.R., Sadre, R., Pras, A.: Difficulties in modeling SCADA traffic: a comparative analysis. In: Taft, N., Ricciato, F. (eds.) PAM 2012. LNCS, vol. 7192, pp. 126–135. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28537-0_13
4. Barbosa, R.R.R., Sadre, R., Pras, A.: A first look into SCADA network traffic. In: Proceedings of Network Operations and Management Symposium (NOMS). IEEE (2012)
5. Barbosa, R.R.R., Sadre, R., Pras, A.: Exploiting traffic periodicity in industrial control networks. *Int. J. Crit. Infrastruct. Protect.* **13**, 52–62 (2016)
6. Bencsáth, B., Pék, G., Buttyán, L., Félegyházi, M.: Duqu: a stuxnet-like malware found in the wild. Technical report Laboratory of Cryptography and System Security (CrySyS Lab), Budapest University of Technology and Economics Department of Telecommunications (2011)
7. Clarke, G., Reynders, D.: Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Newnes (2004)
8. Falliere, N., Murchu, L.O., Chien, E.: W32.Stuxnet dossier. Technical report Symantec, Mountain View (2011)
9. Formby, D., Jung, S.S., Copeland, J., Beyah, R.: An empirical study of TCP vulnerabilities in critical power system devices. In: Proceedings of the 2nd Workshop on Smart Energy Grid Security (SEGS), pp. 39–44 (2014)
10. Formby, D., Walid, A., Beyah, R.: A case study in power substation network dynamics. In: Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 1, p. 19 (2017)
11. Goh, J., Adeptu, S., Junejo, K.N., Mathur, A.: A dataset to support research in the design of secure water treatment systems. In: Havarneanu, G., Setola, R., Nas-sopoulos, H., Wolthusen, S. (eds.) CRITIS 2016. LNCS, vol. 10242, pp. 88–99. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71368-7_8

12. Goldenberg, N., Wool, A.: Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *Int. J. Crit. Infrastruct. Prot.* **6**(2), 63–75 (2013)
13. Hodo, E., Grebeniuk, S., Ruotsalainen, H., Tavolato, P.: Anomaly detection for simulated iec-60870-5-104 traffic. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security* (2017)
14. Jung, S.S., Formby, D., Day, C., Beyah, R.: A first look at machine-to-machine power grid network traffic. In: *Proceedings of International Conference on Smart Grid Communications (SmartGridComm)*. IEEE (2015)
15. Kiss, I., Genge, B., Haller, P.: A clustering-based approach to detect cyber attacks in process control systems. In: *Proceedings of the 13th International Conference on Industrial Informatics (INDIN)*. IEEE (2015)
16. Krotofil, M., Larson, J., Gollmann, D.: The process matters: Ensuring data veracity in cyber-physical systems. In: *Proceedings of the 10th Symposium on Information, Computer and Communications Security (ASIACCS)*. ACM (2015)
17. Landis, R., Koch, G.: The measurement of observer agreement for categorical data. *Int. Biometric Soc.* **33**(1), 159–174 (1977)
18. Lee, R.M., Assante, M.J., Conway, T.: Analysis of the cyber attack on the Ukrainian power grid: Defense use case. Technical report Electricity Information Sharing and Analysis Center (E-ISAC) (2016)
19. Lin, C.Y., Nadjm-Tehrani, S.: Understanding IEC-60870-5-104 traffic patterns in SCADA networks. In: *Proceedings of the 4th ACM Cyber-Physical System Security Workshop (CPSS)*. ACM (2018)
20. Lin, C.Y., Nadjm-Tehrani, S.: Timing patterns and correlations in spontaneous SCADA traffic for anomaly detection. In: *Proceedings of 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. USENIX Association (2019)
21. Lin, C.Y., Nadjm-Tehrani, S., Asplund, M.: Timing-based anomaly detection in SCADA networks. In: D'Agostino, G., Scala, A. (eds.) *LNCS*, vol. 10707. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-99843-5_5
22. Mai, K., Qin, X., Silva, N.O., Cardenas, A.A.: IEC 60870-5-104 network characterization of a large-scale operational power grid. In: *Proceedings of Security and Privacy Workshops (SPW)* (2019)
23. Sayegh, N., Elhajj, I.H., Kayssi, A., Chehab, A.: SCADA intrusion detection system based on temporal behavior of frequent patterns. In: *Proceedings of the 17th Mediterranean Electrotechnical Conference (MELECON)*. IEEE (2014)
24. Udd, R., Asplund, M., Nadjm-Tehrani, S., Kazemtabrizi, M., Ekstedt, M.: Exploiting bro for intrusion detection in a SCADA system. In: *Proceedings of the 2nd International Workshop on Cyber-Physical System Security (CPSS)*. ACM (2016)
25. Yang, Y., Xu, H.Q., Gao, L., Yuan, Y.B., McLaughlin, K., Sezer, S.: Multidimensional intrusion detection system for IEC 61850-based SCADA networks. *IEEE Trans. Power Delivery* **32**, 1068–1078 (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Author Index

- Abie, Habtamu 87
Adir, Allon 87
Aharoni, Ehud 87
Ahmed, Chuadhry Mujeeb 123
Aiello, Maurizio 48
Angelini, Marco 171
Apolinário, Filipe 191
- Bakalos, Nikolaos 77
Baldoni, Sara 67
Battisti, Federica 67
Becue, Adrien 31
Bertone, Fabrizio 139
Besse, Adrien 3
Bimpas, Matthaios 77
Bonomi, Silvia 171
Burke, Kelly 191
- Cambiaso, Enrico 48
Canito, Alda 191
Carli, Marco 67
Carstengerdes, Nils 191
Carvalho, Olga 191
Celozzi, Giuseppe 67
Chandramouli, Krishna 107
Chasiotis, Ioannis 155
Ciccotelli, Claudio 171
- Demailly, Samantha Dauguet 31
Doulamis, Anastasios 77
Doulamis, Nikolaos 77
- Eftychidis, Georgios 155
Escravana, Nelson 191
- Fehling-Kaschek, Mirjam 191
- Gazi, Anna 155
Gazzarata, Giorgia 48
Georgiou, Eftichia 155
Gkotsis, Ilias 155
Greenberg, Lev 87
Izquierdo, Ebroul 107
- Kandasamy, Nandha Kumar 123
König, Louis 191
Köpke, Corinna 191
- Lancelin, David 31
Lin, C.-Y. 207
Lubrano, Francesco 139
- Maia, Eva 31
Manfredi, Salvatore 16
Mangini, Matteo 191
Mantzana, Vasiliki 155
Merlo, Alessio 48
Miller, Natalie 191
- Nadjm-Tehrani, Simin 207
Neises, Jürgen 3
Neri, Alessandro 67
- Palma, Alessandro 171
Papasotiriou, Kassiani 77
Petrucci, Paolo 139
Praça, Isabel 31, 191
- Ranise, Silvio 16
Reis, Bruno 31
Rouquier, Jean-Baptiste 3
- Sciarretta, Giada 16
Soceanu, Omri 87
Sousa, Orlando 31
Srivastava, Kushal 191
Stelkens-Kobsch, Tim 191
Stirano, Federico 139
- Tomasi, Alessandro 16
Troiano, Ernesto 48
- Vaccari, Ivan 48
Varavallo, Giuseppe 139
Verderame, Luca 48
Vitali, Giacomo 139
Voulodimos, Athanasios 77