SCADA System Forensic Analysis Within IIoT

Peter Eden, Andrew Blyth, Kevin Jones, Hugh Soulsby, Pete Burnap, Yulia Cherdantseva and Kristan Stoddart

Abstract A new wave of industrial technology has emerged in the form of Industry 4.0, which has seen a progression from electronic devices and IT (Information Technology) systems that automate production advance to a new revolution of Cyber-Physical Production Systems used for Smart Manufacturing and Smart Factories via IIoT (Industrial Internet of Things). As more and more devices are becoming connected and networked to allow for Smart Manufacturing to take place the number of data sources significantly increases as a result. Real-time Information is then becoming increasingly interlinked across multiple industries for a more efficient productivity process and a reduction in cost. Aside from Smart manufacturing and factories, Industry 4.0 has already seen huge advances in infrastructure management, energy management, transportation and building and home automation. With such industries relying so heavily on real-time data from connected sensors the security of these systems are at risk due to the reliance on low-latency and reliable communication for critical processes. The increase of interconnected networks and devices

P. Eden (✉) · A. Blyth
Information Security Research Group, Faculty of Computing, Engineering and Science, University of South Wales, Wales, UK
e-mail: peter.eden@southwales.ac.uk

A. Blyth
e-mail: andrew.blyth@southwales.ac.uk

K. Jones · H. Soulsby
Cyber Operations, Airbus Group Innovations, Cyber, UK
e-mail: kevin.jones@airbus.com

H. Soulsby
e-mail: hugh.soulsby@airbus.com

P. Burnap · Y. Cherdantseva
School of Computer Science and Informatics, Cardiff University, Cardiff, UK
e-mail: BurnapP@cardiff.ac.uk

Y. Cherdantseva
e-mail: CherdantsevaYV@cardiff.ac.uk

K. Stoddart
Department of International Politics, Aberystwyth University, Aberystwyth, UK
e-mail: kds@aber.ac.uk

L. Thames and D. Schaefer (eds.), *Cybersecurity for Industry 4.0*,
Springer Series in Advanced Manufacturing, DOI 10.1007/978-3-319-50660-9_4

across the Internet significantly increases the amount of entry points into these systems, increasing their vulnerability and allowing outsiders to take advantage of any weaknesses within them. This has already been highlighted by the events of Stuxnet, Havex, Black Energy and the German Steel Mill that targeted ICS (Industrial Control Systems) and SCADA (Supervisory Control and Data Acquisition) Systems causing catastrophic results. The use of SIEM (Security Information and Event Management) services, IDS (Intrusion Detection Systems), IPS (Intrusion Prevention Systems) and firewalls may be implemented within ICS but only operate on the perimeters of their networks or segmented networks and not at the lower operational level where critical processes rely on speed and availability simply because by doing so could introduce latency between critical processes. When events do occur, regardless of whether an incident is accidental or deliberate, an immediate incident response should take place. This chapter focusses on the forensic challenges and analysis of the physical infrastructure that underpins the systems operating within IIoT. It discusses the development of SCADA system architecture over the past few decades and how it has arrived at IIoT, creating the new generation of SCADA systems. The chapter then discusses the current available tools that exist that can help carry out a forensic investigation of a SCADA system operating within IIoT space before closing with a suggested SCADA Incident Response Model.

1 Introduction

The Industrial Internet of Things can be thought of as the next generation of SCADA systems providing the underlying infrastructure for much of the worlds critical infrastructure, such as nuclear plants, oil refineries, water treatment, manufacturing, energy and transport. These systems build on their existing infrastructure by introducing cloud based technologies into the overall network topology. A SCADA system is a hugely distributed computerised system often spanning huge geographical areas, that gathers and analyses real-time data from field devices to automate, monitor and control physical processes. SCADA systems, essentially, monitor and control a network of Programmable Logical Controllers (PLCs) and Remote Terminal Units (RTUs) that use sensors to measure performance of local operation and provide automation. A SCADA control centre collects data from field devices and allows for human interaction and supervisory control of these devices from a central location. IIoT convergence with SCADA has seen more and more control being placed in the cloud.

Originally, SCADA systems were designed to operate on closed networks, using an “air gap” to physically separate them from local networks and the Internet, and therefore minimising the risk of intrusion from the outside. Their main focus had been on making the data available but not necessarily secure or confidential. Over the years, the developments in technology have resulted in SCADA systems communicating over, TCP/IP (Transmission Control Protocol/Internet Protocol), wireless IP

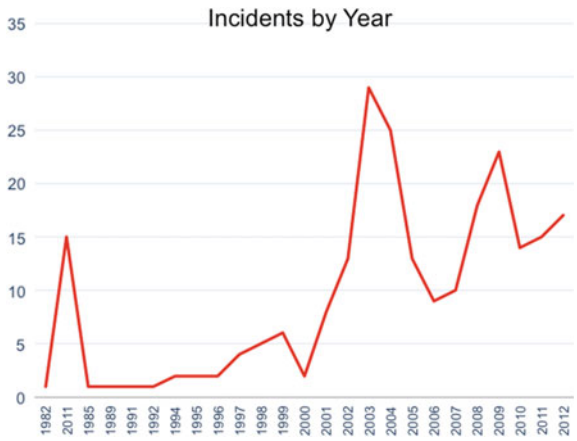

Fig. 1 RISI—no. of ICS incidents per year

and Bluetooth increasing their vulnerability to external attacks. We have seen dedicated attacks on CNI (Critical National Infrastructure) such as Stuxnet, Flame and Duqu.

Figure 1 clearly shows the number of reported incidents steadily rising from 1982 before jumping significantly in the early 2000s. This dramatic change in figures can be attributed to the fact that, around this time, more and more SCADA systems started communicating via TCP/IP and being connected to corporate LAN (Local Area Network). The figures start to decline around the mid 2000s before rising again towards the end of the decade. With IIoT bringing more and more interconnectivity through the cloud and across the internet the number of entry points into a SCADA environment increases, making them more vulnerable and therefore providing more opportunity for attackers to exploit.

When incidents occur it is vital for a forensic investigation to take place to determine the cause and those responsible, but due to the bespoke elements of SCADA systems traditional IT forensic tools and methodologies cannot be applied.

1.1 SCADA Progression and the Development of IIoT

Since the introduction of SCADA into ICS there has been some significant changes and evolutions to the SCADA system architecture that has led to the IIoT revolution.

1.1.1 Monolithic SCADA System

In its infancy SCADA architecture consisted of a centralised standalone mainframe system, with strictly no connectivity to another systems. WANs (Wide Area Networks) allowed for communication between mainframe and various RTUs, using

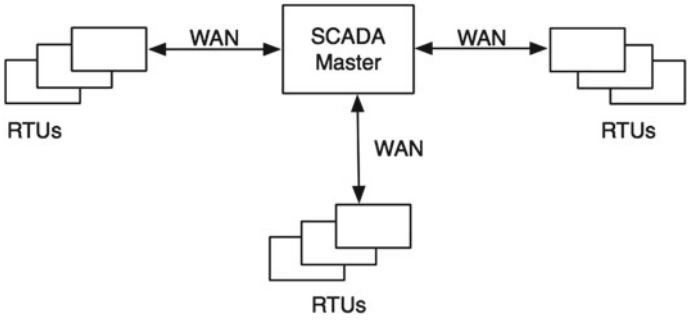

Fig. 2 Monolithic SCADA system

proprietary protocols developed by the RTUs manufacturer and supporting very limited functionality other than carrying out what was required of them. Monolithic systems also made use of a second identical mainframe system that acted as a backup in the event of any redundancy of the master (McClanahan 2003) (Fig. 2).s

1.1.2 Distributed SCADA System

With the introduction to LAN technology, within SCADA, processing could be distributed across many systems allowing for specific station functionality to communicate and share information in real-time with other stations connected to the LAN. This increased the overall processing power of the system. Rather than using mainframes for each station the SCADA architecture now utilised system miniaturisation and now implemented minicomputers at a much lesser cost (Karnouskos and Colombo 2011). Networks were limited to the local environment and the proprietary protocols used were still vendor-specific which limited the networking of different manufacturers devices (McClanahan 2003) (Fig. 3).

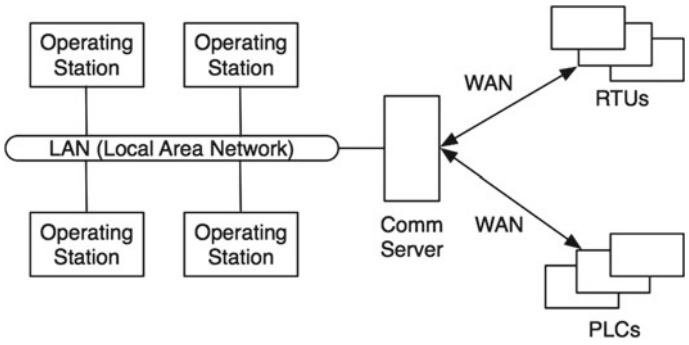

Fig. 3 Distributed SCADA system

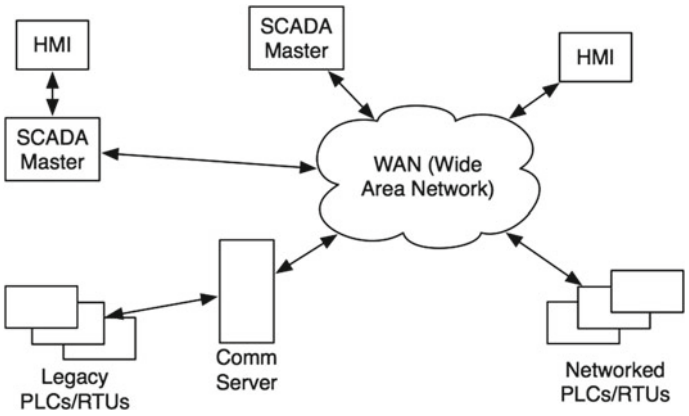

Fig. 4 Networked SCADA system

1.1.3 Networked SCADA System

The emergence of the current networked SCADA system applies the use of open architecture allowing for multi-vendor devices to be networked. It also incorporates open protocols and standards that allows for distributed SCADA functionality across the WAN Rutherford (2012). Significantly, it meant that third party peripheral devices could connect to the network and for communication between master stations ad field devices via IP (McClanahan 2003) (Fig. 4).

ICS and SCADA Information Security Principles are normally in the order of availability, integrity, confidentiality, rather than the traditional IT CIA (Confidentiality Integrity Accessibility) model, as it is deemed more of a priority to have system functionality over confidentiality of information.

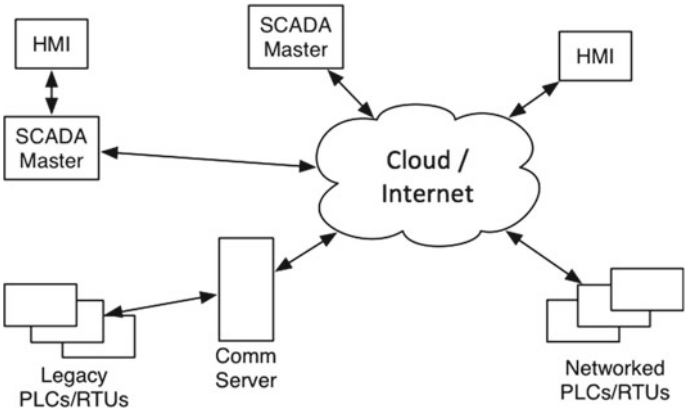

Fig. 5 SCADA system operating over IIoT

1.1.4 Industry 4.0 SCADA System

The latest breakthrough in SCADA system development arrives in the form of the Industrial Internet of Things, which in turn, accounts for a significant part of Industry 4.0. It utilises cloud computing and its commercial availability to improve productivity and reduce infrastructure costs by adopting IoT (Internet of Things) technology (Fig. 5).

2 Conceptual Architecture of a SCADA System

Modern SCADA systems comprise of a series of vital components, both hardware and software, that allow operations to be carried out successfully. These components

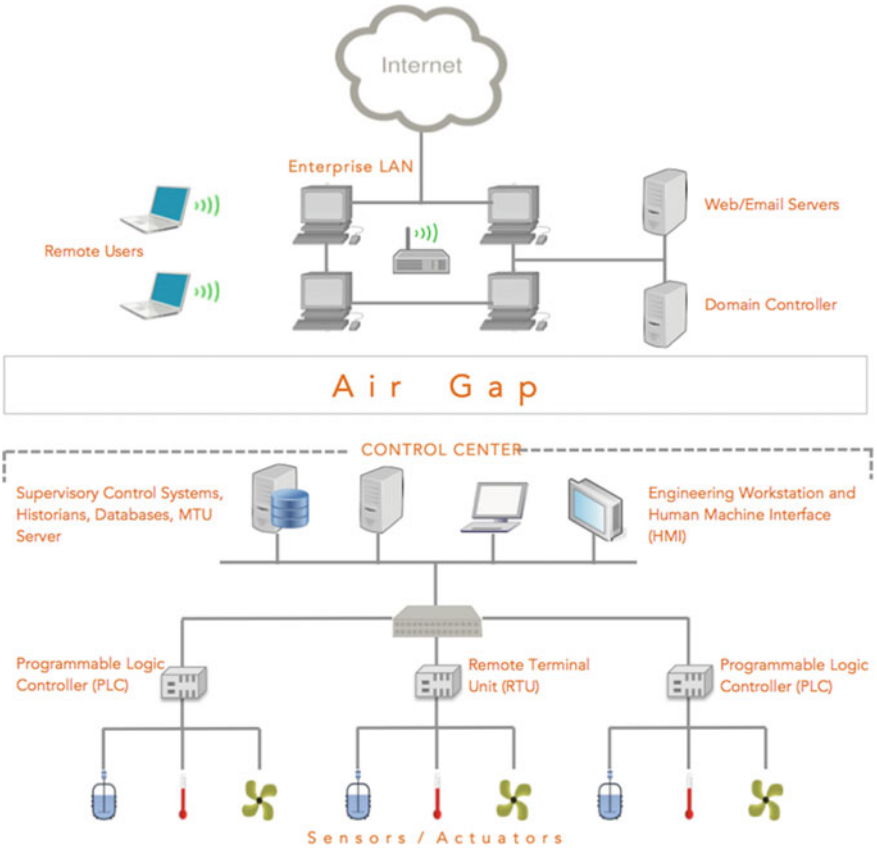

Fig. 6 Conceptual architecture of a typical SCADA system

can be divided into two main sections within a SCADA system; the control centre; and the field sites. The most common components of the control centre include a Human Machine Interface (HMI), Historian, and Master Terminal Unit (MTU). The field sites will normally of comprise of a series of Programmable Logic Controllers (PLCs) and Remote Terminal Units or Remote Telemetry Units (RTUs) (Fig. 6).

2.1 SCADA Hardware

PLC (Programmable Logic Controller): PLCs are computerised devices connected to sensors and are used to control automated processes. They consist of a CPU (Central Processing Unit), memory, power supply, and an input/output interface. They are programmed using a specific control programming language, the most common of which being ladder logic. During operation a PLC will perform an iterating cycle of operations known as a "Program Scan". Firstly, input is received via a sequential scan of the PLC's input interface, which is then stored in memory representing the status of a physical process. This is followed by an execution of the control program that uses the input to decide whether the status needs to change. Finally, the outcome of that decision is stored in an output table and is used to make a change to the operation of the physical process. One complete cycle of the controller is known as a "Scan" and the time for a cycle to complete is known as the "Scan Time". The Program Scan needs to iterate continuously so that it can react to any change in input. The shorter the scan time the faster it can react to these changes.

RTU (Remote Terminal Unit): An RTU is very similar to a PLC and performs virtually the same function in that it gathers data and transmits it back to the control centre. More recently both RTUs and PLCs have become more and more alike sharing common design features. Prior to this, the major difference between the two had been how they communicate with the control centre as well as their size and capabilities. Generally, RTUs have faster CPUs and a much larger support for communication. They also tend to be a bit more rugged and reliable in tough environments. They boast the ability for quick expansion through modularity and also provide flexibility within CPU and I/O (Input/Output) (Boyer 2004).

IED (Intelligent Electronic Device): IEDs allow for monitoring and control functionality as well as electrical protection and perform upper level communication completely independently without having to rely on any other devices.

Control Centre: A unique part of SCADA functionality is the ability to collect information about the state of its field devices and physical processes. PLCs and RTUs will continually transfer data regarding their status to a central control centre. This control centre can play a very important role in a forensic investigation when piecing together events that may have occurred. Its main components consist of an HMI (Human Machine Interface), Historian, MTU (Master Terminal Unit).

HMI (Human-Machine Interface): In order to interpret and visualise data that is transferred to the control centre SCADA systems use an HMI. The HMI not only provides a way to visually present the data that is processed but also allows for human

interaction with the system as a way of controlling its overall state. Depending on what the SCADA system is controlling will ultimately depict the size and design of the HMI interface. This can range from a large-sized, computerised control panels at a nuclear plant to a small computer or even an application on a mobile phone.

Historian: In order to carry out any forensic investigation data needs to be analysed, but first that data needs to be collected and stored ready to be made available to users for analysis and interpretation. The Historian is the Database Management System that stores and archives this data and provides audit logs for all activity across a SCADA network. The functionality of the Historian was originally to provide data trending.

MTU (Master Terminal Unit): The Master Terminal Unit, sometimes referred to as the SCADA server, is responsible for receiving and processing all the data transmitted to the control centre from the field devices as well as providing communicating with those devices. It may pre-process data before sending it to the Historian and also provides a graphical representation of the information stored in it to be transferred ands displayed on the HMI (Stouffer et al. 2008).

2.2 SCADA Software

Software found within a SCADA systems field devices will differ in its objectives depending on the devices it is programmed into but software relating to a SCADA systems HMI or servers will generally provide a level of real-time diagnosis, management control, management information, information relating to specific sensors or systems, logging and reporting (Robles and Choi 2009).

2.3 Networking

SCADA systems communicating throughout the cloud, just like any other network, rely on a network topology across its various layers of communication between its components. Over the years, the originally intended "closed" SCADA control network has not only joined with corporate networks but has also seen a huge integration with the cloud offering a broader level of control and allowing access and monitoring from outside. From a security perspective this severely increases the risk of intrusion and attacks and provides a large level of complexity for a forensic response, as will be discussed. Firstly we will identify the various networking components essential to a SCADA network.

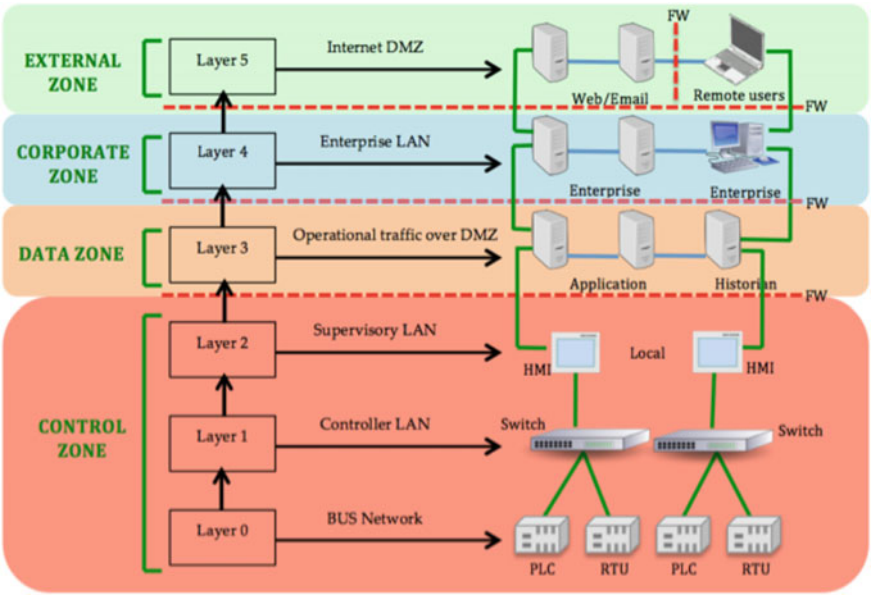

Fig. 7 Communicational zones of a SCADA system

2.3.1 Communication Zones

Modern SCADA systems operating in the IIoT space have evolved considerably since their original flat network architecture and as a result the network structure can be separated into zones. To increase security within the network, SCADA systems perform their most critical communications within the lowest most secure layer (Stouffer et al. 2011). Because connectivity within a SCADA network has multiple layers the forensic acquisition of the necessary data can often be difficult to trace (Wu et al. 2013).

According to Ahmed et al. and further developed by Pedro Taveras there is consistency when describing SCADA system forensic analysis as a 6 layer model. Figure 7 shows that model as well as the zones each layer belongs to.

Control Zone

Layer 0: At layer 0, a bus network connects up the various field device hardware, such as RTUs and PLCs.

Layer 1: Layer 1 contains the controllers that receive signals from the field devices via electrical input. Using standard networking protocols these are then decoded and signals can also be sent as outputs back to these devices as a means of control but also to layer 2 for analysis and further control.

Layer 2: As previously mentioned, layer 2 connects to layer 1 and receives information regarding the lower layers and uses this to present this data to a HMI for interpretation and control.

Data Zone
Layer 3: Layer 3 is made up of historians and application servers as well as domain controllers.

Corporate Zone
Layer 4: Layer 3 consists of all business and enterprise servers for email, DNS (Domain Name System) etc. and business workstations allowing for corporate communication (Stouffer et al. 2011).

External Zone
Layer 5: Layer 5 resides in the external zone of the SCADA network and includes connectivity to remote operations, third party vendors and business partners which ultimately defines IIoT (Knijff 2014).

2.3.2 Communication Protocols

The modern SCADA system is designed to offer real-time updates on the status of its physical processes within its network. These can sometimes cover large geographical areas and contain thousands of sensors and field devices. In order for these updates to occur, and for the successful control of the physical processes, data needs to transmit using secure communication between the field devices and the SCADA host. This is achieved through using a range of specific communication protocols that transport the information from field devices to a central control centre, whether in the cloud or locally (Fig. 6).

Vendors began developing their own communication protocols before standards organisations started developing open standards. Some manufactures even carried on creating proprietary protocols after open standards were made available (Boyer 2004). The convergence with IIoT has seen the number of varying protocols increase, but despite this large number of both proprietary and non-proprietary protocols there are some that are more common than others, such as Modbus, DNP3 (Distributed Network Protocol-3), PROFIBUS (Process Field Bus), WiMax (Worldwide Interoperability for Microwave Access), Wi-Fi, HTTP (Hypertext Transfer Protocol), CoAP (Constrained Application Protocol), AMQP (Advanced Message Queuing Protocol) (Fig. 7).

3 Examples of SCADA System Incidents Prior to IIoT

When security breaches occur within SCADA systems destruction can be life threatening. The following are examples of past system failures within SCADA environments.

3.1 Trans-Siberian Pipeline Explosion

3.2.1 Trans-Siberian Pipeline Explosion The earliest recorded incident involving cyber attacks on a SCADA system was in 1982 on the Trans-Siberian pipeline when a Trojan found its way into its SCADA system software resulting in a 3-kiloton explosion that could be seen from space. The Trojan was responsible for increasing the pressure during a pressure test on the pipeline (Miller and Rowe 2012).

3.2 Maroochy Shire Water System

The SCADA system of the Maroochy Water Sewerage Service consisted of two main computers monitoring 142 sewage pumping stations over 880 kilometres. Each of the stations consisted of SCADA field devices that would raise alarms, process instruction and communicate real-time data describing the pumps status to the control centre. In early 2000 a disgruntled ex-employee named Vitek Boden, who had previously been employed as a site supervisor, hacked into the systems over a period of several months. His actions prevented alarms from being reported to the central control centre as well as stopping communication between the control centre and the certain pumping stations, resulting in a million litres of sewage water flooding into a nearby river. He achieved this by altering the identification numbers of some of the pumping stations so that signals meant for one station would be sent to another. He used wireless equipment to gain access to the SCADA system and redirected insecure radio communications. The problem could have been avoided if the company had placed sufficient access control within its SCADA system especially regarding wireless access restrictions (Abrams and Weiss 2008) (Fig. 8).

3.3 Stuxnet

Still unsure of its architect, by September 2010, the propagation of the Stuxnet worm had infected around 45,000 computers despite appearing to be directed specifically at Iranian Industrial Control Systems running secure facilities a such as nuclear power plants or gas pipelines. By exploiting weaknesses within the Windows Operating System running Siemens Simatic STEP 7 software Stuxnet's aim was to reprogram Programmable Logic Controllers to function outside of their intended boundaries. This resulted in the plants centrifuges, responsible for separating nuclear material, spinning dangerously faster than originally intended causing damage and destruction. Despite being an isolated network the use of a removable storage device such as a USB (Universal Serial Bus) drive allowed the worm to penetrate and spread into the SCADA system of an Iranian Nuclear Power Plant. As soon as it had crossed the "air-gap" it could traverse through the network via LAN and into PLCs where it

Table 1 Stuxnet vulnerability exploits

Vulnerability	Description
MS08-67: RPC (Remote procedure call) vulnerability in server service	Allows for a remote user to gain equal rights to a local user and take control of an affected system remotely
MS10-046: LNK vulnerability in windows shell	An attacker may exploit vulnerabilities in the handling of windows shortcut files (.LNK) to insert malware remotely
MS10-061: Spool server vulnerability in print spooler service	Allows for an attacker to make a specially designed print request resulting in them taking over the server
MS10-073: Win32k.sys vulnerability in windows Kernel-Mode drivers	Allows an attacker to execute kernel privileges
CVE-2010-2772	Vulnerability in Siemens Simatic WinCC and PCS 7 SCADA system allows for attacker to use known default passwords to gain access

would infect WincCC and STEP 7 files. Documented in Critical Infrastructure Protection by NATO (North Atlantic Treaty Organisation) Advanced Research are the 5 vulnerabilities that Stuxnet exploited (Ibrahim and Faisal 2012) (Table 1).

The lack of any integrity check on messages and their sources allowed for Stuxnet to interfere with commands from the process control network without PLCs or any operators knowing.

3.4 Duqu

A year later, after Stuxnet, a new malware was discovered, resembling many of its design and structure features. It was given the name Duqu because the temporary files created by the malware's key logger all began with "DQ.." (Bencsath). Stuxnet had paved the way for targeted attacks on control systems and Duqu was just another example of the threat to CNI. Duqu was more aimed at stealing information using its key logger to obtain keystrokes, files and screen shots, a kind of industrial espionage or cyber-surveillance attack (Bencsáth et al. 2011).

3.5 Flame

Flame, also known as Flamer and sKyWlper, followed in the footsteps of Duqu as an "information stealer" and is an example of a more complex malware aimed at SCADA and industrial control systems. Like Duqu, it could steal screenshots and keystrokes but it also had the ability to activate web cams and microphones. By

disguising itself as a proxy for Windows updates it infected over a thousand systems across the Middle East and Iran (Bencsáth et al. 2011).

4 SCADA Forensics Within IIoT

New age SCADA systems that use cloud-based technology for analysis and control through IIoT ultimately rely on their physical components and sensors in the OT (Operational Technology) layer of the infrastructure to operate correctly and safely. During a forensic investigation of such a system it is these lower level devices that will hold key information that is critical to determining the cause of the incident. Having discussed some of the more commonly known, high-impact breaches of SCADA system security it is important to realise that a thorough investigation is mandatory each time an incident occurs, regardless of whether the incident was a result of malicious intent or not. A forensic investigation of a security breach or system failure aims to identify those responsible as well as the cause of the incident. When incidents occur the need for a forensic response is essential for understanding how the events happened and piecing together who was responsible. Identifying the cause of the attack will then provide a basis for patching the system to improve its security and therefore help prevent the same attack happening twice (Wu et al. 2013).

A forensic response to SCADA system failure is essential for several reasons:

- It identifies the root of an incident, and potentially those involved
- It identifies if the system is still at risk and what changes were made to the system
- It identifies the damage caused and the total probable damage
- It highlights weaknesses in SCADA systems that can be improved to reduce the risk of the incident reoccurring (Ahmed et al. 2012).

4.1 Forensic Challenges

In order to understand the key issues regarding digital forensics within SCADA systems operating within IIoT we must first understand that there is a clear distinction between IT systems and SCADA systems. It is the complexity of a SCADA environment that separates it from a traditional IT system and therefore precludes the application of standard forensic methods and tools. Current research shows a consistency in the issues and challenges faced by the forensic investigator when dealing with SCADA systems and these can be broken down into several key areas. Below is a taxonomy of the current forensic challenges existing in ICS and SCADA incident response.

4.1.1 Live Forensics

As SCADA is at the heart of all critical infrastructures it is essential that it operates continuously and is never turned off for any reason. As a result the usual standard forensic techniques, normally applied in IT, cannot be applied for data acquisition in these instances. Instead, the practice of live forensics is required. This allows for the process of data acquisition and analysis to be carried out whilst the SCADA system is running (Ahmed 2012). It is essential that during this process both volatile and non-volatile data be acquired. Non-volatile data may be stored in hard disks attached to various SCADA hardware, such as MTUs and Historians. Volatile data contains vital information describing the current state of running system and is found in the physical memory of SCADA devices. It differs from non-volatile data in that its content is constantly being overwritten and updated with newer information. This creates even further problems during an investigation.

4.1.2 Rapid Response

Data of any evidential value contained within physical memory will be at its peak just after an incident happens. From that moment on, due to the nature of volatile data, the amount of useful information will decrease as older processes and services are overwritten by newer ones (Taveras 2013). For this reason it is vital that a forensic response is carried out as quickly after an incident has occurred as possible before important information is lost. This can create another challenge when a SCADA system is spread over many thousands of square kilometres. Many of the embedded devices found in SCADA systems, such as PLCs, have a relatively small amount of memory and flash storage. As systems continue to run, data is overwritten and therefore the length of data retention is very small (Wu et al. 2013).

4.1.3 Integrity and Validity

A key part to any digital forensic investigation is to be able to obtain evidence in a forensically sound manner in order to prove its integrity and validity in a court of law. Digital evidence is normally verified by matching the hash value (calculated by applying a hashing algorithm to the data) of the original evidence against its acquired copy. This digital "fingerprint" proves that the data under examination and analysis has not been modified in any way, as any modification would cause the hash value to change and therefore not match the original. According to Ahmed the challenge within SCADA systems is that because the system remains live, and data is continuously being updated, the state of the data can change from the start of the copying process to completing a calculated hash, resulting in the hash being unusable.

4.1.4 Incident Specific Information/Logs

As some SCADA systems have avoided being updated over the years, due to the heavy risk of interference it may cause to a live system, older technologies are still present in many environments. This may be in the form of legacy or no longer supported hardware and therefore, as a result, can lead to a distinct lack of detailed logging (Fabro and Cornelius 2008). Effective logging can assist significantly in a forensic investigation and help piece together a timeline of events. According to Fabro et al., it is not uncommon for systems with logging and audit functionality to be deployed with these functions disabled. It is vital that when logging features are absent or insufficient in a system, that network traffic be logged to help understand device communication at the time of an incident.

4.1.5 SCADA Forensic Tools

Research shows a clear absence of data acquisition tools and methodologies designed specifically to incorporate SCADA systems, including their protocols and proprietary log formats (Ahmed et al. 2012). This may be, partly, due to the transparency of the effect such tools can have on live SCADA services as well as many other issues that may have prevented the production of such tools already.

4.1.6 Order of Volatility

For a tool to acquire real-time data from a live system it is inevitable that, during that process, the memory state of that system or device will change. In order to maximise the amount of data of evidential value being extracted during the data acquisition process, the tool would have to follow an order of volatility, beginning with the most volatile (Registers, Cache) and moving towards the least volatile (Archival media) (Fabro and Cornelius 2008). This lightweight approach will minimise the amount of changes memory and reduce the amount of disruption to the network (Wu et al. 2013).

4.1.7 Remote Data Acquisition

Research carried out by EADS (European Aeronautic Defence and Space) (Wu et al. 2013) emphasises the need for such a tool to be able to extract and acquire data remotely from a suspect system to an investigators machine directly via the network. They discuss the current forensic tools able to carry out this method of acquisition such as, ProDiscover and EnCase Enterprise which, when installed on a suspect system, can be used to extract forensic artefacts from an MTU or HMI etc.

4.1.8 Compatibility

In order to operate effectively the tool must be compatible with all SCADA system devices, even with those running their own exclusive operating systems or Linux variants. Many SCADA components run customised kernels in their operating systems in order to improve performance or to provide support for applications. The tool would have to be able to communicate with those operating systems in order to be able to operate successfully and exchange the relevant information (Ahmed et al. 2012).

4.2 Current Data Acquisition Methods for SCADA Systems

Although there are no specifically designed data acquisition forensic tools aimed solely at SCADA systems, there are various tools and methods that are currently being used to extract data from SCADA system components.

5 Forensic Acquisition of SCADA Artefacts

Firstly, a SCADA forensic artefact can be thought of as any data that provides explanation to the current state of a SCADA system, device or media. Data of forensic value within SCADA systems can exist in two separate streams; data that is communicated across a network; and data that is stored in a device (Knijff 2014). The latter can be further categorised as to which zone, within the SCADA architecture, that device exists. This section will aim to highlight the key tools and methods for forensically acquiring data from both the network and from the physical assets.

5.1 Network Data Acquisition

Data passing over a SCADA network can be captured in various ways and using a variety of tools. The following is a list of current tools available to perform network data acquisition within a SCADA environment.

5.1.1 In-Line Network Taps

Sniffing traffic over a network can be achieved through the use of network taps and placing them at keys points within a network, known as ‘choke-points’. Ideally, they would be placed between switches, on ethernet lines or in-between individual assets. A network tap is a device that copies network traffic passing through it to a monitor

port (Hjelmvik 2011). Implementing the use of link aggregation taps allow for both downlink and uplink traffic to be captured. Network taps can only be connected onto a SCADA network when it is safe to do so, during downtime of operations or during maintenance periods. This will eliminate any disruption to critical processes. The tap can then be connected to a separate machine dedicated for the collection of that data.

5.1.2 Port Mirroring

When network taps cannot be implemented an alternative can be to use port mirroring or SPAN (Switch Port Analyzer) to obtain SCADA network data from managed switches. By connecting a monitoring system to a managed switch, a copy of the packets sent through that switch, or separate ports on that switch, can be mirrored to a single port. That port can then be used to acquire the data. To acquire the data a monitor session must be started (CA 2015). This includes;

- the session number: to identify the monitoring session
- session source: the desired ports to mirror
- session direction: specifies the direction of the mirrored traffic, i.e. receive (RX) or transmit (TX) or both.

5.1.3 TCPdump

Much like Wireshark, but less labour intensive, TCPdump can be used as both a network monitoring tool as well as a tool to acquire network data from within a SCADA network. Data obtained via TCPdump will include timestamps, network protocol used, source IP and port, and destination IP and port (Green and Vanden-Brink 2012).

5.1.4 Wireshark

Wireshark is an open source protocol analyser and can be used to capture packets being sent across a network. Acquired data will be stored as .pcap files for later analysis or can be monitored live in real-time as data is communicated. Wireshark also supports many ICS and SCADA protocols.

5.1.5 Serial RS232 and RS485 Taps

Many devices found within SCADA networks rely on serial communication and although Wireshark also supports serial communication data there are several other tools that can be used. Much like implementing the ethernet tap an RS232 or RS485

network tap could be introduced to the network during scheduled downtime or maintenance periods to obtain serial communication data.

5.1.6 PortMon

Portmon is a utility found within Windows based systems and allows for monitoring and capturing of serial data. Simply executing the portmon.exe program file will start to capture serial communication data.

5.2 Device Data Acquisition

5.2.1 PLC:

Acquiring data from PLCs is dependent upon certain factors, such as whether the PLC needs to remain active or whether it can be powered down. The first instance poses many problems. If a PLC has to remain live for critical processing any interference to those processes may result in disastrous consequences. When this is the case sometimes the software used to program the PLC can be used to monitor and record certain vital data such as memory variable values as they alter (Wu et al. 2013). Examples of this would include using Siemens STEP7 software to record the data from any Siemens S7 PLCs, or Schneider Electric's SoMachine software to record memory address alterations in their Modicon PLC range.

Over the years there has been a distinct lack of dedicated forensic tools for PLCs and similar embedded devices (Ahmed et al. 2012) but some software tools are starting to emerge to overcome the problem. As well as using the PLCs manufacturing tools to retrieve data there are tools such as PLC Analyzer Pro and PLCLogger that perform similar functionality. PLC Analyzer Pro is a software tool designed for acquisition and analysis of recorded data on Siemens SIMATIC devices.

PLCLogger is an open source software tool and provides similar functionality to PLC Analyzer Pro with the addition of supporting any device using Modbus-TCP or Modbus-UDP.

There has been some research into the development of a solution for the security monitoring of low level SCADA devices which could potentially aid a forensic investigation within a SCADA environment. Cruz et al. (2015) suggests the use of the SSU (Shadow Security Unit) which is placed in parallel to field devices for continuous monitoring of a device. The device can check for abnormal behaviour of a PLC and through physical probing go the I/O modules can provide real-time data acquisition capabilities (Cruz et al. 2015). A similar concept is discussed by Janicke et al., implementing a run-time monitoring framework using an Ardruino Yun device, alongside a field device to ultimately capture snapshots of PLC states, i.e. values for inputs/outputs, counters and timers etc., to aid in the forensic analysis after an incident has occurred (Janicke et al. 2015).

If a PLC can be powered down for forensic analysis or is already powered off as a result of an attack then certain techniques can be used to read data from the on-board memory chips themselves through JTAG (Joint Test Action Group), chip off or ISP (In-System Programming).

JTAGging and In-System Programming are both non-invasive methods for achieving the same results. JTAGging is the process of interacting with the Test Access Points (TAPs) of the microcontroller in such a way to acquire raw data from any connected memory chips.

In-System Programming is a way to acquire data by bypassing the CPU itself and connecting directly to on-board storage chips, such as eMMC or flash storage and then pulling the raw data from them. Hardsploit is a hardware and software device designed with critical electronic and embedded devices in mind. It allows for both ISP and JTAGging to be carried out and a dump of the raw data to be obtained. The raw data can then be interpreted using a hex editor such as WinHex or HxD.

Chip off is regarded as an invasive acquisition procedure as the memory chips are physically desoldered and removed form the PLCs PCB and then read using specific chip readers to acquire the image. Chip off may be the only option if chips are already physically damaged and need to be repaired before imaging. Tools and equipment for this process would include a desoldering station to remove the chip and Hardsploit to acquire the data from it.

Once a raw image has been acquired it can then be interpreted to establish program code and ladder logic such as function.

5.2.2 HMI:

Much like a PLC the HMI typically has a fairly limited amount of on-board storage. However, the data stored on the chips could be crucial in a forensic investigation. The HMI is the interface at which a human interacts with the control devices. Decisions are made based on information passed back from field devices to the HMI. The HMI can store critical information such as event logging, alarm logging, issued commands, diagnostics and reports on the most recent status of particular field devices (Fabro and Cornelius 2008).

Performing data acquisition from HMI devices will mirror very closely the approach used with PLC devices. Vendor-specific software tools used to program the HMIs will often have monitoring and recording features which should be enabled when possible. Physical interrogation of the devices will involve ISP, JTAG and Chip-off to recover an image of the raw data, as explained in Sect. 3.2.1.

5.2.3 Engineering Workstations/General Workstations/Servers:

Workstations and Servers found at the control, data and corporate zones can all be approached in the same manner when it comes to a forensic response. Each system is going to contain different types of forensic artefact depending on the role its plays

within the SCADA environment. The underlying fundamental elements are that they will all contain data stored in both memory and on physical storage that may be vital to investigation. Therefore, different tools are generally needed for RAM (Random Access Memory) acquisition and for physical media extraction.

Disk Imaging: There is an array of disk imaging software tools that can be used to extract a forensically sound full image of internal and externally attached disks from a machine. Different tools have varying levels of capabilities and the preferred tool of choice may be dependent upon which operating system is running on the source machine.

AccessData's FTK (Forensic ToolKit) Imager is a common software tool used to create digital images of physical drives as well as the ability to obtain a full memory dump. FTK Imager Lite is a variation of the tool on USB format which eliminates the need to install any software on the source machine.

EnCase Forensic Imager can be used as an alternative to FTK Imager and ultimately performs the same functionality offering similar imaging formats and capabilities. However, a case study carried out by Muir (2015), of a comparison between EnCase version 7.10.00.103 and FTK Imager 3.3.0.5, showed that EnCase created more of a footprint than FTK when being run live on a target machine. This would be a factor to consider when acquiring a memory dump of a system as vital processes may be overwritten.

DD is a Linux command-line tool built in as standard on Linux and Unix systems and one that can also be installed on Windows machines. The dd command can be used to copy entire mounted drives both locally and remotely.

RAM Acquisition: There are also various tools that can be used to acquire memory from a device that is running such as running processes, services, drivers, registry data, network data and event logs. Tools need to be carefully selected when dealing with memory acquisition as the tools being loaded to acquire the memory will also run in memory. This could potentially overwrite vital artefacts. Running command line tools are much more advantageous then GUI tools as they use less memory space.

Dumpit, a tool created by MoonSols for Windows systems, is an open source memory acquisition tool than can be run from a USB.

Memoryze, created by Mandiant, is very similar to dumpit and is run from a USB using the command-line. It is also a free tool and allows a complete memory dump to be passed to an externally connected drive or over a network.

Mandiant Redline is capable of extracting and auditing a full memory image of a workstation in a forensically sound manner. It was designed to detect malicious activity within memory. Its IOC (indicators of Compromise) functionality allows for the identification of malicious files and processes.

LiME can be used to acquire a memory dump from a linux system. Again, this can occur locally buy installing LiME on the host machine or can be acquired over the network via TCP.

Volatility is a cross-platform tool that can also be used to extract digital artefacts from live volatile memory and also provides analysis functionality (Stirland et al. 2014).

5.3 Half-Life of Data Within a SCADA System

When an incident occurs and an investigation is undertaken a forensic investigator needs to know where data is and how long it will last there. Given the complex nature, sheer scale of possible data sources, and various interconnected networks within a typical SCADA system, calculating the half-life of data for an entire system would be impossible as it would change dramatically from one system to another and be dependant on the type of incident that has occurred and the devices running. For example, data would last a lot longer in a historian than it would in an engineers workstation, which in turn would last significantly longer than data stored in a PLC.

This implies that the half-life of data should be identified at a lower level, for each data source but even this would be individual device specific. For example, 2 identical Siemens S7 1212c PLCs that hold 25kb of volatile memory, 1kb of load memory and 2kb of retentive memory will not share the same half-life as, despite the program scan time to scan each of the inputs and outputs will be exactly the same, the list of instructions to execute in the one PLC may be significantly higher than the other meaning that data in memory is written over at a faster rate and therefore resulting in a lot lower half-life.

These characteristics should be carefully considered when prioritising devices during an incident response.

6 SCADA Forensic Process

6.1 Existing Incident Response Models

There are many models for a forensic response to normal IT systems that follow a generic model of identification and preservation, collection, examination, analysis and reporting in a forensically sound manner, but there is very limited documentation regarding ICS/SCADA forensic incident models at the low level. There are various recommended guidelines such as Homeland Security's "Developing an Industrial Control Systems Cybersecurity Incident Response Capability" (Security 2009) (and many similar) which give good guidance on incident planning, prevention and management but lack any detail of how to actually perform forensics on a SCADA system at a low level. There are, however, some effective post-incident SCADA forensic models that have been suggested such as those put forth by Kyle Wilhoit, a threat researcher from Trend Micro, (Wilhoit 2013) and Tina Wu of EADS (Wu et al. 2013) that incorporate the full SCADA System into the forensic investigation. These added elements into the forensic response model are essential for SCADA systems over normal IT systems.

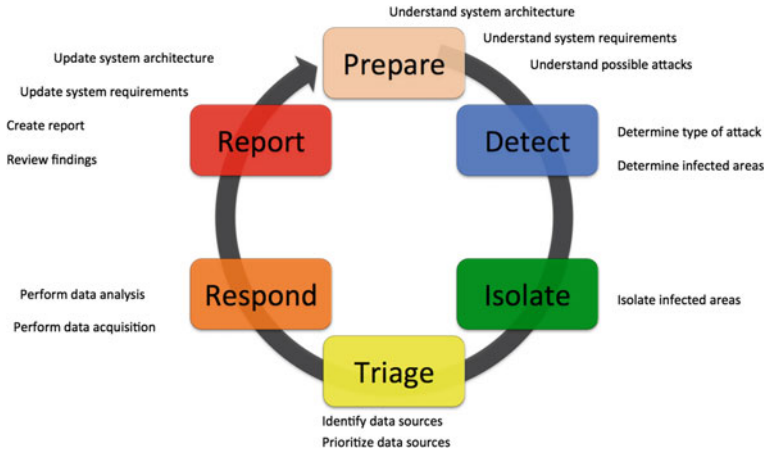

Fig. 8 SCADA forensic incident response model

6.2 Forensic Methodology for SCADA Within IIoT

A SCADA forensic response should not just take place after an incident has occurred but also before and during an incident. The more detailed information the investigator has access to regarding a SCADA system under investigation the more it will increase the level of forensic evidence recovered. Below is a proposed Forensic Incident Response Model.

SCADA forensic process models have been suggested in the past, such as that proposed by Wu et al. (2013), which adapts the traditional IT system forensics investigation process and applies it to SCADA systems. However, the incident response model proposed in this paper is an alternative, original model, first submitted to ICS-CSR (Industrial Control System-Cyber Security Research) 2015 (Eden et al. 2015), and now further developed, that treats SCADA forensics as more of an ongoing lifecycle, using the entire process to influence the next event.

Figure 8 shows the SCADA forensic incident response model consisting of six main stages; Prepare; Detect; Isolate; Triage; Respond; and Report. The final stage helps to improve the preparation for the next time an investigation is needed, therefore continuing the cycle.

6.2.1 Stage 1: PREPARE

It is vital that the preparation stage starts before an event takes place. This will involve ensuring all documentation relating to the particular SCADA system is accurate and should comprise of understanding the following areas:

- Understand system architecture: As each SCADA system will be unique in its configuration it is essential that detailed documentation regarding the system's network, hardware and software is collected and recorded. The networking information should involve network configurations, a network map and all entry points into the system. Hardware documentation should include all SCADA components, including manufacturers, makes and models. The software documentation should include all software running on each device across all zones. Accurate geographical documentation regarding locations of field devices and device half-life etc. should also be available.
- Understand System Requirements: Given the classification of certain SCADA devices it is also essential for the forensic investigator to have access to specific system requirements for the SCADA system being investigated. Documented here should be the types of systems and devices that need to remain continually running without fail, those that can be switched to a back-up, and finally those devices that can be powered down.
- Understand Potential Attacks: It is also important to gather threat intelligence and to understand the types of attacks that can occur on the system. It has already been discussed by Zhu et al. (2011) and further acknowledged by Stirland et al. (2014), that the types of possible SCADA related attacks can be divided into 3 sections. These are hardware, software and the communication stack. Detailed information relating to these types of attacks can be found at Zhu et al. (2011).

6.2.2 Stage 2: DETECT

- Determine type of attack: When an event has taken place, or is in the process of taking place, an investigator should try to determine the type of attack based on assessments of real-time data and any unusual behaviour that may have occurred.
- Determine potential infected areas: Attempt to determine potential infected areas based on assessments made from the previous step. This will help in the next stage when identifying possible data sources.

6.2.3 Stage 3: ISOLATE

- Isolate infected areas: After detecting potential infected areas an attempt can be made to isolate those networks and devices, dependant upon their system requirements within the SCADA environment and business operations.

6.2.4 Stage 4: TRIAGE

- Identify data sources: The triage stage should start by identifying possible data sources of interest for interrogation. This will be influenced by the documentation from the planning stage together with the threat intelligence of stage two and

within the isolated area. The list should include the device location within the network; device make, model and serial number; and classification status i.e. process critical.
- Prioritise data sources: The next step is to create a prioritisation list of data sources. This needs to be ordered in a way that reflects their value, volatility and accessibility in order to maximise the potential evidence available for recovery (Knijff 2014). The time taken to assemble a priority list could also have an effect on the amount of evidence recovered as certain SCADA systems in critical infrastructure can span huge geographical areas and contain hundreds of data sources.

6.2.5 Stage 5: RESPOND

- Perform data acquisition: With a priority list established the next stage involves forensically acquiring the data from the relevant data sources, which can either come from data stored in components or from data across the network (Knijff 2014).
  Data needs to be admissible in court and therefore, should be acquired using forensically sound methods. The types of data acquired at this stage should include memory dumps, disk imaging and chip imaging from across the system. Traditional IT forensic tools can be used against engineering workstations, servers and historians but for embedded devices such as PLCs and RTUs, flashing software may be required from the manufacturer to extract a raw memory dump using JTAG (Joint Test Action Group) ports and ensuring that no affect is made to the operation of the device if required to remain operational (Stirland et al. 2014). Invasive methods such as chip-off forensics may be used to extract data as a last resort but would be dependant on a component's classification. Clear guidelines would have to be established for each type of asset. The data acquisition stage should also include acquiring network data through retrieving logs data captures.
- Perform data analysis: Data analysis will involve separating the forensic artefacts from other SCADA system data. This may be carried out with the use of traditional forensic analysis tools.

6.2.6 Stage 6: REPORT

- Review findings: Based on the analysis stage relationships can be correlated between the recovered forensic artefacts to ultimately create a timeline of events and to establish the root of an incident.
- Create report: Based on the analysis of recovered artefacts a report should be compiled regarding results and findings. Inferences should be made between relationships of the gathered data, which should also include validation and integrity of data records such as chain of custody reports. It should also include any recommendations towards the development or patching of the SCADA system.

- Update system architecture: The final steps of the reporting stage should be to update the documentation relating to the SCADA system architecture, post incident. This is due to the fact that after an event has taken place the overall configuration of the SCADA environment may have changed and will need to be accurate for the next investigation.
- Update system requirements: Similar to the previous step, in the light of an incident occurring and system configurations changing, SCADA system requirements may also need to be revisited and, therefore, would need to be recorded.

6.3 SCADA Forensic Workstation

To develop and propose a SCADA forensic workstation we first must consider the software and hardware tools needed to perform data acquisition on all types of data sources as well as the analysis of recovered data. Stirland et al. suggest a similar strategy for developing a SCADA forensic toolkit in which I will adapt and add to in order to cater for physical extraction of embedded devices (Stirland et al. 2014).

6.3.1 Hardware

High Spec Machine—To efficiently process large amounts of data. Must include multiple connection ports.
Write-Blocker—To image data sources in a forensically sound manner. Can be used on all SCADA servers, Back-end databases, Engineering workstations, Historians, HMI hosts.
Memroy Imaging Tool—To acquire volatile memory on running data sources. Can be used on all SCADA servers, Back-end databases, Engineering workstations, Historians, HMI hosts that need to remain running.
JTAG Kit (incl. Screwdriver set, Multi-meter, solder iron and solder, Jtagulator, Bus Blaster/Bus pirate—Screwdriver set to disassemble device. Multi-meter to test JTAG ports. Solder and solder iron to connect wires to JTAG ports. Jtagulator to determine JTAG TAPs. Bus Blaster to extract data. Can be used on any SCADA embedded devices i.e. PLCs, RTUs, HMIs.
Storage Drives (HDDs (Hard Disk Drive)/SSDs(Solid State Drive))—To store forensic images of data sources and all acquired SCADA data. Can be used on all captured data.
Camera—To document live data acquisition processes. Can be used on all data sources where live interrogation is performed to capture steps taken by investigator.

6.3.2 Software

FTK Imager—For creating forensically sound images of data on devices. Used for all database servers, HMI hosts, engineering workstations.
AccessData FTK Toolkit/Encase—To process and analyse acquired forensic images. Can be used on all filesystem data from HMI, Engineering workstations, Servers, Historians.
Putty—For use on the forensic workstation to communicate with JTAG devices and PLCs/RTUs.
Vendor-Specific Flasher Software—To be used to acquire raw data during the JTAG process (if available) from PLCs and RTUs.

WinHex—To view acquired raw data dumps from data sources such as PLCs, RTUs, and RAM dumps.
Data Hash Generator—To generate a hash value for captured data to prove integrity. Can be used on all acquired data.
TCPDump—To capture post-incident data remnants of a network. Can be used on all network data.
Wireshark/Network Miner—To filter and acquire SCADA network protocols and traffic. Can be used on all SCADA networks.
Volatility—To perform in-depth analysis of captured live data. Can be used on all database servers, HMIs, Engineering workstations, Historians.
AlienVault ICS SIEM—SIEM and integrated forensics tool that can be used to parse acquired network data allowing the user to define specific rules for monitoring. Can be used on all network data (Stirland et al. (2014)).

7 Conclusion

IIoT provides a fundamental core for industrial control systems, including much of the world's critical national infrastructures, to operate continuously and safely without disruption or interference. As the majority of these systems that were initially SCADA systems designed to operate on closed networks are now operating through the cloud over TCP/IP the risk from outside targeted attack is already evident. Any interference to them could cause huge economical damage and even loss of life. When an incident occurs it is essential that a forensic response is undertaken, following defined procedures and methodologies and with the correct tools. Current research clearly highlights a distinct lack of dedicated SCADA certified forensic tools for incident response and an absence of a forensic triage model for carrying out investigations. As a result suggestions have been discussed to aid the forensic response to SCADA incidents. Ongoing research should be directed at developing such tools, methodologies and models to aid in IIoT forensic investigations post-incident.

8 List of Abbreviations

AMQP—Advanced Message Queuing Protocol
CIA—Confidentiality Integrity Accessibility
CNI—Critical National Infrastructure
CoAP—Constrained Application Protocol
CPU—Central Processing Unit
CRC—Cyclic Redundancy Check
DNP3— Distributed Network Protocol-3
DNS—Domain Name System
EADS—European Aeronautical Defence and Space
HDD—Hard Disk Drive
HMI—Human Machine Interface
HTTP—Hypertext Transfer Protocol
ICS—Industrial Control System
IIoT—Industrial Internet of Things
IOC—Indicator of Compromise
IoT—Internet of Things
IP—Internet Protocol
I/O—Input/Output
IT—Information Technology
JTAG—Joint Test Action Group
LAN—Local Area Network
MTU—Master Terminal Unit
NATO—North Atlantic Treaty Organisation
NIST—National Institute of Science and Technology
PLC—Programmable Logic Controller
PROFIBUS—Process Field Bus
RAM—Random Access Memory
RISI—Repository of Industrial Security Incidents
RPC—Remote Procedure Call
RTOS—Real Time Operating System
RTU—Remote Terminal Unit
SCADA—Supervisory Control and Data Acquisition
SPAN—Switch Port Analyzer
SSD—Solid State Drive
SSU—Shadow Security Unit
TAP—Test Access Point
TCP/IP—Transmission Control Protocol/Internet Protocol
USB—Universal Serial Bus
WAN—Wide Area Network
WiMAX—Worldwide Interoperability for Microwave Access

Acknowledgements This work is funded by the Airbus Group Endeavr Wales scheme under the SCADA Cyber Security Lifecycle (SCADA-CSL) programme with the ultimate goal of improving the forensic handling and incident response process for SCADA systems.

References

Abrams M, Weiss J (2008) Malicious control system cyber security attack case study-maroochy water services. Australia, Technical report, NIST

Ahmed I, Obermeier S, Naedele M (2012) Scada systems: challenges for forensic investigators. Computer 450(12):44–51

Bencsáth B, Pék G, Buttyán L, Félegyházi M (2011) Duqu: a stuxnet-like malware found in the wild. Technical report, laboratory of cryptography and system security (CrySyS)

Boyer S (2004) Scada. ISA-the instrumentation, systems, and automation society, Research triangle park, NC

CA (2015) Data acquisition: best practices guide. Technical report, CA technologies

Cruz T, Barrigas J, Proenca J, Graziano A, Panzieri S, Lev L, Simões P (2015) Improving network security monitoring for industrial control systems. In: 14th IFIP/IEEE international symposium on integrated management (IM 2015)

Eden P, Blyth A, Burnap P, Cherdantseva Y, Jones K, Soulsby H, Stoddart K (2015) A forensic taxonomy of scada systems and approach to incident response. In: 3rd international symposium for ICS and SCADA cyber security research 2015

Fabro M, Cornelius E (2008) Recommended practice: recommended practice: creating cyber forensics plans for control systems. Technical report, department of homeland security

Green T, VandenBrink R (2012) Analyzing network traffic with basic linux tools. Technical report, SANS Institute InfoSec Reading Room

Hjelmvik E (2011) Intercepting network traffic. NETRESEC (Network forensics and network security monitoring). http://www.netresec.com/?page=Blogandmonth=2011-03andpost=Sniffing-Tutorial-part-1---Intercepting-Network-Traffic

Ibrahim M, Faisal M (2012) Stuxnet, duqu and stuxnet, duqu and beyond. Int J Sci Int J Sci Eng Invest 1(2):75–78

Janicke H, Nicholson A, Webber S, Cau A (2015) Runtime-monitoring for industrial control systems. Electronics 40(4):995–1017

Karnouskos S, Colombo AW (2011) Architecting the next generation of service-based scada/dcs system of systems. In: IECON 2011—37th annual conference on ieee industrial electronics society, pp 359–364

McClanahan R (2003) Scada and ip: is network convergence really here? IEEE Industry Appl Mag 90(2):29–36

Miller B, Rowe D (2012) A survey of scada and critical infrastructure incidents. Proceedings of the 1st Annual conference on research in information technology. New York, NY, USA. ACM, pp 51–56

Muir B (2015) Encase imager versus ftk imager. http://bsmuir.kinja.com/encase-imager-vs-ftk-imager-1677906594. Accessed 21st June 2016

Robles R, Choi M (2009) Assessment of the vulnerabilities of scada, control systems and critical infrastructure systems. Int J Grid, Distrib Comput 2

Rutherford D (2012) Make the most of your energy ethernet for scada systems. Technical report, Schneider electric telemetry and remote SCADA solutions

Homeland Security (2009) Recomended practice: developing an industrial control systems cybersecurity incident response capability. Technical report, Homeland security

Stirland J, Jones K, Janicke H, Wu T (2014) Developing cyber forensics for scada industrial control systems. In: Proceedings of the international conference on information security and cyber forensics. SDIWC Digital Library

Stouffer K, Falco J, Kent K (2008) Guide to industrial control systems (ics) security. gaithersburg, md: U.s. department of commerce, national institute of standards and technology. Technical report, NIST (National institute of standards and technology)

Stouffer K, Falco J, Scarfone K (2011) Recommendations of the national institute of standards and technology. NIST

Taveras P (2013) Scada live forensics: real time data acquisition process to detect, prevent or evaluate critical situations. Eur Sci J

van der Knijff RM (2014) Control systems/scada forensics, what's the difference? Digit Invest 110(3):160–174

Wilhoit K (2013) The scada that didn't cry wolf. Technical report, Trend Micro

Wu T, Disso J, Ferdinand P, Jones K, Campos A (2013) Towards a scada forensics architecture. In: Proceedings of the 1st international symposium for ICS and SCADA cyber security research

Zhu B, Joseph A, Sastry S (2011) A taxonomy of cyber attacks on scada systems. In: Proceedings of the 2011 international conference on internet of things and 4th international conference on cyber, physical and social computing, pp 380–388