# Developing a State of the Art Methodology & Toolkit for ICS SCADA Forensics

Molly Betts[1], Joseph Stirland[1], Funminiyi Olajide[1], Kevin Jones[1], Helge Janicke[2]
[1]*Airbus Group*
[2]*De Montfort University*

## Abstract

*Supervisory Control and Data Acquisition (SCADA) systems are used in different Critical National Infrastructure (CNI), including Electric Power, Oil & Gas, Manufacturing, Utility, Transportation services and others. The underpinning control systems have unique characteristics such as being real-time and safety critical. Therefore interference and disruption of the services from cyber attack poses a significant risk to; the environment, properties, economies and human lives. Responding to such events in not trivial, and recovering the required forensic evidence to understand the cause and consequence of such an event is key. Further, developing a suitable incident response methodology to identify evidential artefacts of the causes of disruption is crucial, should security mechanisms fail. In this paper we present the state of the art methodology forensic toolkit for cyber incident response on Industrial Control System (ICS) environment of SCADA plus evaluate the applicability of current IT forensic tools and the requirements of an 'ICS forensic toolbag'. The research work presents an experimental case study of a malware USB device based attack, a man in the middle attack and a remote access attack.*

## 1. Introduction

Given the current pace at which technology is progressing, the maintenance/update cycles of SCADA, and the rapid evolution of threats against SCADA systems, it is unlikely that the limitations found within SCADA forensics will abate, ultimately affecting incident response to cyber-attack. Technology is increasing in the development of sophisticated tools for SCADA incident response. This monitoring system can analyse cyber incidence in an environments that contain both legacy and up to date systems. Forensic toolkits and methodologies for traditional ICT infrastructure are well understood, as they are mostly based around standard IT systems and are supported by forensic tools, however such tools lack SCADA specific capabilities.

The digital forensics investigation process involves the recovery, analysis, and presentation of digital evidence found in any seized electronic devices. These often act as part of a criminal and corporate investigation, on the improvement of cyber defence to identify the questions of who, what, why and how evidence is found on the allocated memory of computer systems. This process is undertaken in a way that is legally admissible, reliable and with the ability to follow a best practice methodology [1], [2]. Forensic artefacts include systems, devices, and documents that comprise the control systems and traffic packets (data in transit) [3]. When considering SCADA forensic investigations and incident response, it is important to consider the types of systems that are used within SCADA infrastructure, including the devices information that are commonly found on the memory, with their associated data that is relevant to a forensic investigation.

SCADA systems can monitor and control hundreds of hundreds, and thousands of input/output points [4], including HMIs, PLCs and Data Historians. SCADA forensics involves developing and testing a methodology that draws on both incident response and cyber forensics models. Stirland et al [5] and Wu et al [6], proposed the application of a state of the art toolkit. This article provides an evaluation of different methods using the results of experimental case study of cyber-attacks on SCADA devices. A pre-prepared collection of state of the art forensic software and hardware are used for the digital investigations of SCADA architectures. A paper of [7], presented that to test the validity of forensically sound evidence, the evidence gathered are to be coherently and materially unaltered.

The detailed contributions of this article include the evaluation, analysis of components including limitation of SCADA systems from forensics perspective. The existing methods and current forensic toolkits robustly, lack readiness for active response on varied cyber-attack incidence on SCADA systems, resulting to research gap analysis of SCADA forensic tools. Therefore, in this paper, we present the state of the art methodology and toolkit for SCADA systems. Not only the technical part of critical control systems were investigated, also, the investigators perspective, timeline analysis, research goals and mind-set awareness are described. The research ensures coverage of all SCADA components whilst maintaining forensic soundness of digital artefacts.

## 2. Current Forensics Process

SCADA systems were initially designed and developed to be physically isolated from corporate networks and external Internet connections. This is known as an "air gap", encouraging the belief that there was little need for extra network security mechanisms, as the only way to transfer information would be via removable devices on site. Due to the unique aspects of SCADA, security was mostly obtained through obscurity [8], and the focus was on the functionality and availability of services rather than the confidentiality and integrity aspects [5]. Air gapped systems are a progressively uncommon approach, as there is increasing connectivity between SCADA and IT infrastructures [9]. SCADA architectures have progressed to utilising Internet-based communications in order to seamlessly integrate SCADA information and external information [1], as well as enhancing system reliability and enabling remote system recovery.

These connections have expanded SCADA systems' attack surface for external threats originating from the internet [10] as SCADA proprietary protocols as an extension of traditional proprietary control protocols such as Modbus-TCP and DNP3 which were initially designed without security mechanisms given the expectation for an isolated system. When these bespoke protocols are transported over open standards and communications protocols, it could lead to an attack ultimately, exposing the unprotected SCADA protocol, allowing for unauthorised access and control of the SCADA network [11].

Belief in the protection and isolation provided by air gaps [12] becomes an issue when deployed in SCADA operations as the distribution of security updates or patches are not considered once the connection has been cut. Whilst some attacks may not propagate without an internet connection, it can easily be transferred via a storage medium such as a USB device through the actions of a third party contractor, as highlighted by Paganini [13]. Severing a network connection with an air gap simply creates new pathways that may contain remote devices, which are more difficult to manage yet just as easy to infect [12].

Nevertheless, it could be argued that while an air gap is not perfect, it provides an extra layer for adversaries to overcome whilst conducting a cyber-attack and will likely form one part of a 'defence in depth' strategy of CNI operators.

Defence of cyber assets can be difficult as a cyber-defender must protect the entire network, whereas an attacker only has to find one successful attack vector [14]. This calls attention to the requirement for security mechanisms to be placed on different layers of a network, rather than using a single security mechanism around the perimeter [15]. This defence in depth approach also increases the availability of monitoring and log data that is vital to forensic investigations.

As a SCADA infrastructure consists of varying devices and integrated networks, the number of attack vectors become vast and diverse in methods

To develop and apply the methodology of forensic toolkit, for an appropriate integration into future SCADA forensic processes, the components and challenges within SCADA forensics must be addressed. Evidence types that may be present during investigations and case studies of past incidents must also be accounted for, in order to tailor the experiments correctly to the SCADA test bed.

## 3. Methodology Application

A structured and consistent approach is vital to the development of SCADA forensic investigations, and keeping focus on further research and development are required [16]. SCADA forensic methodology requires specialised processes including acquiring data from embedded systems such as PLCs and RTUs [6].

The methodology proposed by Stirland et al [5], in Figure 4 highlights the steps taken during a forensics investigation of a SCADA system, which is supported by an applicable toolkit discussed further in Section IV. The process draws on the incident response and cyber forensics models [5], ensuring coverage of all components, whilst maintaining forensic soundness of digital artefacts. When acquiring data to be presented as evidence it is important to conduct a forensically sound capture method, otherwise it may result in invalid evidence.

Process of forensic investigation followed the common practice of digital forensic and the flowchat of this is described in Figure 1.These were described in phases of the incident response and forensic process of a typical SCADA environment.

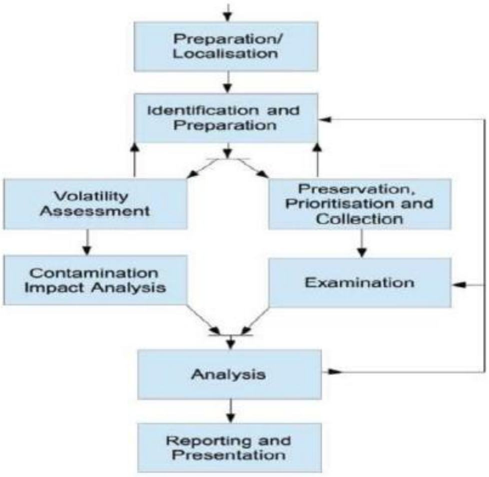

Preparation/ Localisation
Identification and Preparation
Volatility Assessment
Preservation, Prioritisation and Collection
Contamination Impact Analysis
Examination
Analysis
Reporting and Presentation

Figure 1. SCADA Incident Response and Forensic Process

Phase 1: Preparation & Localisation

During the initial phase, potential sources of evidence such as systems, networks and connected devices are identified. Any other components such as servers, routers and access terminals that could have a relationship with the SCADA system will also be examined.

Phase 2: Identification & Preparation

Identification of the types of systems to be investigated is undertaken including operating system, manufacturer, serial numbers and model of PLCs and network design and implementation. This phase should also include the geo-location identification of devices as required considering carefully the scale and distributed nature of the control environment.

Phase 3: Volatility Assessment, Contamination Impact Analysis, & Preservation, Prioritising and Collection

This phase involves three steps of the methodology and re-quires assessing the volatility of resources immediately after they are identified. This is to aid progression of the priority list used within Preservation, Prioritisation and Collection and Volatility Assessment. Level of volatility is to be documented along with impact on the reproducibility of the investigation results. Contamination Impact Analysis: The impact of volatile data capture should be assessed against the safety and operation of the system.

Identification of the impact on the volatility during the collection and analysis of volatile data items with lower priority of the SCADA system should also be undertaken. Preservation, Prioritising and Collection: Any highly volatile data is to be forensically captured and stored to maintain integrity for analysis. All potential evidence from the systems that are suspected to be a part of the affected SCADA system being investigated should be collected. Network traffic is also captured to discover any anomalies in the data. Volatile and dynamic information across network cards and controller units are to be prioritised to prevent the loss of any data.

Phases 4: Examination & Phase 5: Analysis

The analysis phase consists of finding relationships between the recovered forensic artefacts and piecing the evidential data together. This is performed in order to develop a timeline of the incident and its impact on the control environments. As SCADA is relatively unique, the examination process should also include engineering representatives who are familiar with the operation of the system.

Phase 6: Reporting and Presentation & Phase 7: Reviewing Results

The investigator is to compile their findings and analysis into report(s). This should include recommendations for engineers and to carefully consider the requirements and operation of a SCADA system. For clarity, the results and findings should be reviewed to ensure validation and that the forensic chain of custody for information has been met and is forensically sound.

Each step within the methodology proposed by Stirland et al [5] is adhered to during the investigation of all used cases and in a controlled environment. The methodology is applied to other post-attack in order to make clear details of the investigation process, as this will aid in distinguishing the devices affected in each attack including, the tools required for the investigation, and the evidential data discovered, as the attack delivery of each experiment differs. It will allow for evaluation on whether the evidence gathered is admissible and valuable, thus providing results which support the theoretical methodology and toolkit proposed by Stirland et al [5].

The data captured from the components mentioned will allow an insight into what traces were left behind by the experiments, the extent to which traces can be found on varying devices and systems, operations that are affected and traces of how the incident occurred. In order to do this, SCADA specific tools and currently available forensic tools must be considered.

## 4. State of the Art SCADA Forensic Toolkit

As “state of the art” is not defined within digital forensics, it is best described as the most recent stage in the development of forensics products, incorporating the newest ideas and features [17]. A forensic toolkit is a pre-prepared collection of open source or commercially available forensic tools installed on a forensic computer prior to conducting an investigation. It includes tools for acquiring data, undertaking analysis, and compiling the findings into reports [5]. A state of the art cyber forensic investigation consists of known methodologies and challenges within forensic investigations and technology, both current and future, including the existing forensic tools. In Figure 2, this paper described a typical test environment of SCADA Production Line Testbed. In test environment, the purpose of the experiments is to replicate common attack methods on sub-scale systems of establishments which utilise SCADA systems, such as production or power generation facilities, as they are commonly targeted. Further example sectors include:

- Critical National Infrastructure services such as gas pipelines, water treatment and distribution, electricity providers, energy and oil.
- Manufacturing such as production lines.
- Transportation including train signals, dispatching and traffic lights.
- Communication such as telephone communications.

Generally, SCADA systems are also used by establishments including government, emergency and health services, finance, telecommunications, hazards, and the food and beverage industry [18]. Attacks are directed at these establishments to disrupt and degrade services in order to stop production, gain information or cause damage. Figure 2, illustrate the SCADA production line testbed used for the experimental attacks. These systems can provide evidential data including communications over the network such as issued commands, and may provide user information, volatile data and system logs. This is in relation to devices attached. Logs and evidential information depend on the delivery method of the attacks conducted, such as the different components affected by the malware USB device, the man in the middle network attack and possibly remote access attack.

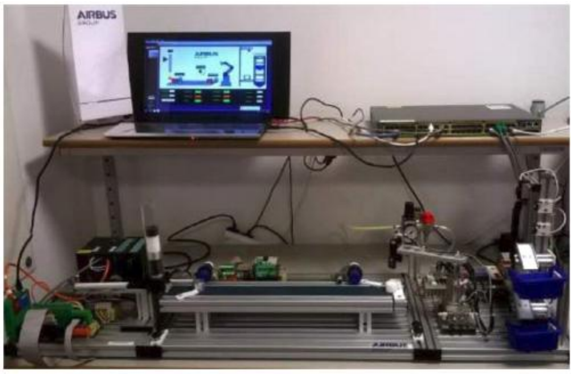


Figure 2. SCADA Production Line Testbed

It could be argued that launching specific attacks is biased due to familiarity of the test devices; however a testbed which acts as a microcosm of real world vulnerable environments is a reasonable approximation of critical industries providing that the complexity and interoperability of the system remains representative such is the case in the design of this experimental environment. Testbeds that include a selection of devices used in real systems are preferable for evaluation of new methodologies and tools, as they are representative extrapolations however failures are contained and do not risk lives. Variations of these devices are a typical system configuration in a SCADA environment. Alongside relevant and plausible attacks, it is required to get the most from testing the forensic methodology and toolkit proposed for efficiency, and to discover onto what extent evidence can be identified and examined.

## 5. Experiments

Common and commercially available attack tools are used within the experiments to show how attacks can be undertaken with tools which are easy to obtain. For example, USB Rubber Ducky [19] is a Human Interface Device (HID) and acts as a keyboard using simple Ducky scripts and is a cross-platform tool. This tool is utilised particularly for social engineering and penetration testing with plenty of resources online which are mostly open source, along with a wide community of support. Metasploit Framework [20] and Kali Linux [21] are well known tools and are used for penetration testing. They are regularly updated with a wide support community. It is unlikely that skilled adversaries use simple GUI based tools against critical infrastructure, however, graphical details of simple attack methods are provided, and traces can be identified using the methodology and toolkit.

During the experiments, it is assumed that methods of reconnaissance and social engineering have been undertaken beforehand. Varied methods of attack have been chosen to highlight the alternative routes that can be taken to gain access to a SCADA system, which resulted in varying types of evidential data. As the attacks take place, a SCADA based methodology and forensics toolkit is used for the acquisition of data, information of relevant networks and devices found in each used cases.

The data acquired is used for analysis to determine if a breach has occurred. The extents to which the system is compromised with what functional operations and assets are affected including, how the breach of incident occurred were evaluated towards attribution [5]. Each of the experiments are simple attacks that are described step by step, so that the evidence found after applying a methodology and toolkit can be referenced to each different methods within the attack vectors. This paper therefore, presents on three experiments carried out on a typical SCADA system environment. These includes, experiment on infected Malware USB attack, Man in the Middle attack and Remote Access attack.

### 5.1. Malware USB Attack

This attack emulates the human element in an incident when (inadvertently or maliciously) a user plugs USB device infected with malware into a machine such as an engineer’s workstation, as was the case in Flame [22,23,24] and Stuxnet [25,26] attacks. USB devices are extremely common and do not require Internet access, circumventing protection provided by air gaps. Overview is shown in Figure 3 and Figure 5 below. Experiment steps are as follows:

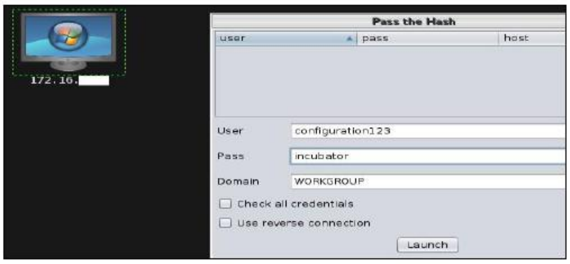
**Step 1 - Social Engineering**: In a real world instance an operator can be manipulated into utilising a USB device, even if it is just by leaving the device within the vicinity, such as a desk or car park as it is likely to be picked up and plugged in [22]. Elements of social engineering are a common platform used to launch attacks.

**Step 2 & 3 - Attaching the Rogue USB and Hidden Account Creation**: The USB Rubber Ducky Device contains a script which creates a hidden user account with full administration privileges. This device is disguised as a typical USB device which is attached to the HMI, and uses the Ducky script to configure the account.

**Step 4 & 5 - Payload and Exploitation**: Figure 3 shows the adversary machine once it has scanned the victim machine with the credentials readily input to connect via a reverse shell.

**Step 6 – Screenshot**: Once the machine is exploited, the adversary can take screenshots or capture images through the webcam with the option of setting zoom preferences, in order to understand more about the device and its purpose. Clicking Watch (10s) automatically snaps an image every ten seconds.

**Step 7 - SCADA Manipulation**: Once the victim machine has been exploited it is free for manipulation. In this example malicious PLC logic is uploaded to the HMI and executed in order to disrupt the operations of the production line attached to the system. For simple disruption, the device can be remotely restarted or shut down. There are also methods for data exfiltration allowing an attacker to download, delete and upload files, modify timestamps on a file or directory, as well as enabling keylogging.



| user | pass | host |
|---|---|---|

User configuration123

Pass incubator

Domain WORKGROUP

□ Check all credentials

□ Use reverse connection

Figure 3.Entering adversary generated credentials during attack

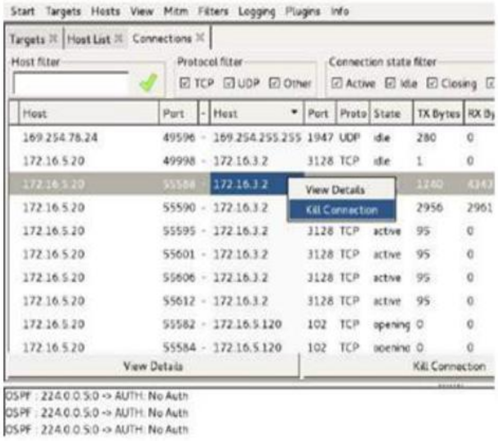
## 5.2. Man in the Middle Network Attack

Typical methods include packet sniffing and modifying data transmitted across a system, focusing on network devices and protocols. This consists of infiltrating transmitted information and data and modifying it in turn, by destroying the integrity and confidentiality of information, and the availability of the systems depending on the modification. Stuxnet [27] utilised an example of a complex man in the middle attack [14]. This simple attack utilises Ettercap 0.8.2 on Kali, the services provided by the HMI are simply being disrupted to provide evidence of an attack which can be identified using forensics tools and the methodology. Figure 5 described the steps taken in the experiment as follows:

**Step 1 - Targeting Hosts**: Assuming the attacker has undertaken reconnaissance of a system; the attacking machine scans the subnet and adds the HMI and PLC IP addresses as target hosts.

**Step 2 – Poisoning**: The attacking machine then ARP poisons the selected targets allowing the attacker to view and sniff the connections for statistics. As a result, an adversary can disrupt the traffic by killing the connections or modifying the data. In Figure 4, connections to the TCP port which is the connection used by the HMI and PLC to communicate are killed using ettercap.

**Step 3 - Denial of Service (DOS)**: eEttercap's Dos plugin can also be used to completely disable the services simply by using a target IP address and an unused IP address. The unused address then floods the target with SYN packets, consuming resources.



Host filter

Protocol filter ☑ TCP ☑ UDP ☑ Other

Connection state filter ☑ Active ☑ Idle ☑ Closing ☑

| Host | Port | - | Host | Port | Proto | State | TX Bytes | RX By |
|---|---|---|---|---|---|---|---|---|
| 169.254.78.24 | 49596 | - | 169.254.255.255 | 1947 | UDP | idle | 280 | 0 |
| 172.16.5.20 | 49998 | - | 172.16.3.2 | 3128 | TCP | idle | 1 | 0 |
| 172.16.5.20 | 55588 | - | 172.16.3.2 |  |  |  | 1240 | 4343 |
| 172.16.5.20 | 55590 | - | 172.16.3.2 |  |  |  | 2956 | 2961 |
| 172.16.5.20 | 55595 | - | 172.16.3.2 | 3128 | TCP | active | 95 | 0 |
| 172.16.5.20 | 55601 | - | 172.16.3.2 | 3128 | TCP | active | 95 | 0 |
| 172.16.5.20 | 55606 | - | 172.16.3.2 | 3128 | TCP | active | 95 | 0 |
| 172.16.5.20 | 55612 | - | 172.16.3.2 | 3128 | TCP | active | 95 | 0 |
| 172.16.5.20 | 55582 | - | 172.16.5.120 | 102 | TCP | opening | 0 | 0 |
| 172.16.5.20 | 55584 | - | 172.16.5.120 | 102 | TCP | opening | 0 | 0 |

```
OSPF : 224.0.0.5:0 -> AUTH: No Auth
OSPF : 224.0.0.5:0 -> AUTH: No Auth
OSPF : 224.0.0.5:0 -> AUTH: No Auth
```

Figure 4. Killing target connections using Ettercap

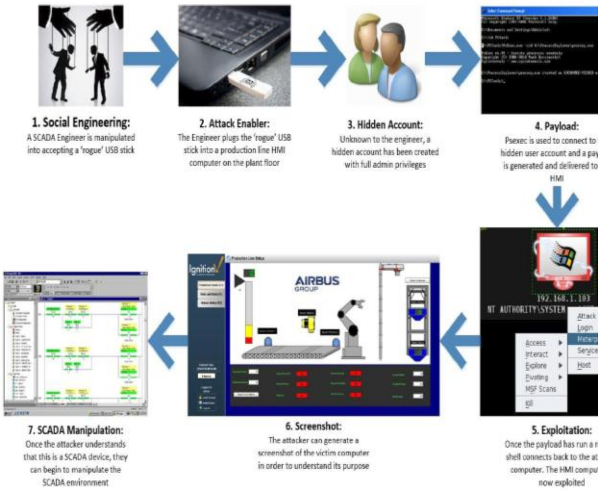
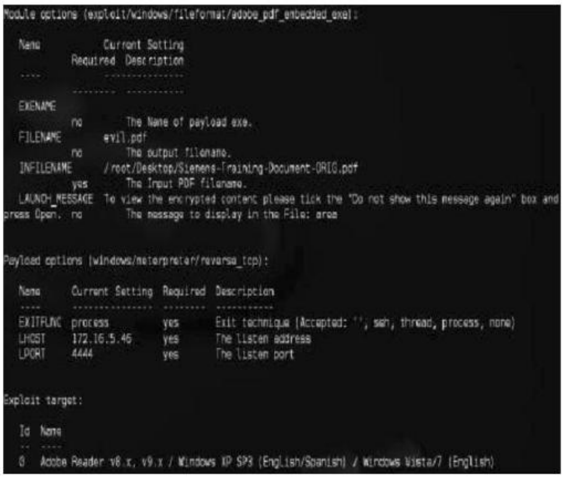

1. Social Engineering:
A SCADA Engineer is manipulated into accepting a 'rogue' USB stick
2. Attack Enabler:
The Engineer plugs the 'rogue' USB stick into a production line HMI computer on the plant floor
3. Hidden Account:
Unknown to the engineer, a hidden account has been created with full admin privileges
4. Payload:
Psexec is used to connect to hidden user account and a pay is generated and delivered to HMI
192.168.1.103
NT AUTHORITY\SYSTEM
Attack
Login
Meterp
Service
Host
Access
Interact
Explore
Pivoting
MSF Scans
Kill
5. Exploitation:
Once the payload has run a r shell connects back to the at computer. The HMI compu now exploited
AIRBUS GROUP
Ignition
6. Screenshot:
The attacker can generate a screenshot of the victim computer in order to understand its purpose
7. SCADA Manipulation:
Once the attacker understands that this is a SCADA device, they can begin to manipulate the SCADA environment

Figure 5.The attack flow scenario utilising a USB device

## 5.3. Remote Access Attack

A common client side exploit includes distribution of PDFs or other attachments via email which contain embedded executables. This is a simple example of a spear phishing attack, which could also easily be converted to a watering hole attack if customers download the PDF from a compromised website. This attack documented by Offensive Security [28] exploits a HMI via an Adobe Reader vulnerability using Metasploit. In this experiment, a typical step of a remote access attack carried out is described as follows:

**Step 1 - File Creation**: Information gathered is used to create an executable which is embedded in an existing PDF documenting training for the Siemens TIA Portal and is attached to a tailored phishing email for the victim. Figure 6 displays the options used to create the malicious PDF.



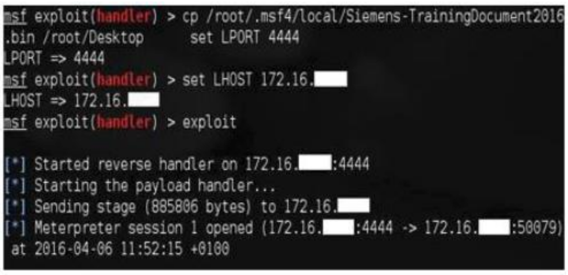
```
Module options (exploit/windows/fileformat/adobe_pdf_embedded_exe):

   Name           Current Setting
           Required  Description
   ----           ---------------
           --------  -----------
   EXENAME
           no        The Name of payload exe.
   FILENAME       evil.pdf
           no        The output filename.
   INFILENAME     /root/Desktop/Siemens-Training-Document-ORIG.pdf
           yes       The Input PDF filename.
   LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this message again" box and
press Open.  no      The message to display in the File: area


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.16.5.46      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)
```

Figure 6. Displaying the options input for malicious PDF creation within Meterpreter

**Step 2 - Multi Handler Listener**: Before sending the email, a multi handler listener is started in Kali to capture the reverse connection. Then using the send Email script, any SMTP server, attaching the malicious PDF and any convincing senders email address, the email is ready to be sent. When the user opens the PDF, a reverse TCP connection is created back to the attacking machine, shown in Figure 10. The attacker then moves the shell to a different process to avoid losing it when the user kills the Adobe Reader process.

**Step 3 - Post Exploitation**: A keylogger can be started to further monitor and exploit the network by gathering credentials and understanding the purpose of the machine. As there is a shell on the system, the adversary is free to conduct any post exploitation work relevant to their goal, whether it be data exfiltration or disruption of services. The experimental attacks chosen are common methods [5], used against both enterprise systems and are increasingly targeting SCADA systems, as highlighted by statistics provided by multiple incident response teams and previous infamous SCADA attacks such as unprecedented hack of Ukraine's Power Grid [29]. According to Dell's 2015 Annual Threat Security Reviews [30], SCADA attacks increases compares to the previous years and it was found that buffer overflows continue to be the primary attack method, accounting for about 25% of the attacks. The threat Report discusses key SCADA attack methods and found that the attack total had doubled in 2014 compared to 2013. ICS-CERT Monitor September 2014 to February 2015 [31] emphasised that unauthorised network scanning is one of the most common attempted attack methods against SCADA systems.

This means that from November 2015 ICS-CERT Monitor [32], there has been a 20% increase over 2014 regarding SCADA directed attacks. In another example, Spear phishing [33] continues to be the most common attack vector, followed by unknown methods and the abuse of authorised access on critical national infrastructure of ICS/SCADA such like manufacturing, energy, water treatment, and transportation systems which are found to be the most targeted sectors respectively.



```
msf exploit(handler) > cp /root/.msf4/local/Siemens-TrainingDocument2016
.bin /root/Desktop      set LPORT 4444
LPORT => 4444
msf exploit(handler) > set LHOST 172.16.[redacted]
LHOST => 172.16.[redacted]
msf exploit(handler) > exploit

[*] Started reverse handler on 172.16.[redacted]:4444
[*] Starting the payload handler...
[*] Sending stage (885806 bytes) to 172.16.[redacted]
[*] Meterpreter session 1 opened (172.16.[redacted]:4444 -> 172.16.[redacted]:50079)
 at 2016-04-06 11:52:15 +0100
```

Figure 7. The adversary machine once the PDF has been opened and a session has start

The purpose of these experiments is to emulate a real world attack environment for which a forensics methodology and toolkit can be applied and evaluated, whilst also being specific to SCADA environments. After these tests, the relevant forensic tools are applied throughout each phase of the methodology in order to ensure that the process is forensically sound, and all possible sources of evidence are accounted for.

Other common attack vectors include back doors and holes in the network perimeter, vulnerabilities in common protocols, attacks on field devices, database attacks, communications hijacking [34], and Cinderella attacks on time provision and synchronisation [35].

Elements of infamous attacks are considered whilst developing test attack methods to give the application of the methodology and toolkit more integrity, and to display that they can be applied in real world scenarios. Therefore, the elements can include the USB delivery methods as used in Flame [36] and Stuxnet, a man in the middle method attack, similar to the method utilised in Stuxnet [1], [30], [37], [38], [39] though network based, an attack utilising spear phishing methods as found in Duqu [1], [31], [40], [41] and the recent Ukrainian Power Plant attack [42], remote access methods such as in the Maroochy Shire Water Services incident [10],[43], and a DoS attack as found in the Davis-Besse Nuclear Facility incident[44].

## 6. Results

The experiments conducted use different attack vectors requiring various forensic tools to be used in order to acquire the multitude of evidential sources and data produced. Locations of data will slightly differ between OS types and release versions, and also for various SCADA devices from different vendors and generations. As a result, it is crucial that tools must be compatible for differing systems, legacy devices and for various vendors. Table 1 above displays the evidential sources provided by SCADA devices within the testbed using various recovery tools within each experiment. It is important to note that the OS used on the SCADA testbed is Windows 7 Ultimate therefore locations of evidence in Table 1 and the forensic tools used relate to this OS version. Throughout the steps of each experiment, a number of different evidence types can be found. This is not a comprehensive list of all evidential sources, but an overview of most relevant information found from using enterprise tools. In relation to Figure 3 of the methodology in Section III, each phase is discussed with reference to each attack and evidence types that were presented and as shown in phases:

**Phase 1: Preparation & Localisation and Phase 2: Identification & Preparation**

During the first two steps of the methodology the results are similar for all three attacks. The devices found in the SCADA network are identified and made note of, including devices such as HMIs, PLCs and network devices, the brand and models, OS, serial numbers, whether they are live or have been shut down. Siemens S7-300 PLC – Live Workstation/HMI – Asus laptop running Windows 7 Ultimate and Siemens Ignition Software – Live Network devices, including routers, switches and firewalls etc. Live. The USB device used to distribute malware may also be left behind.

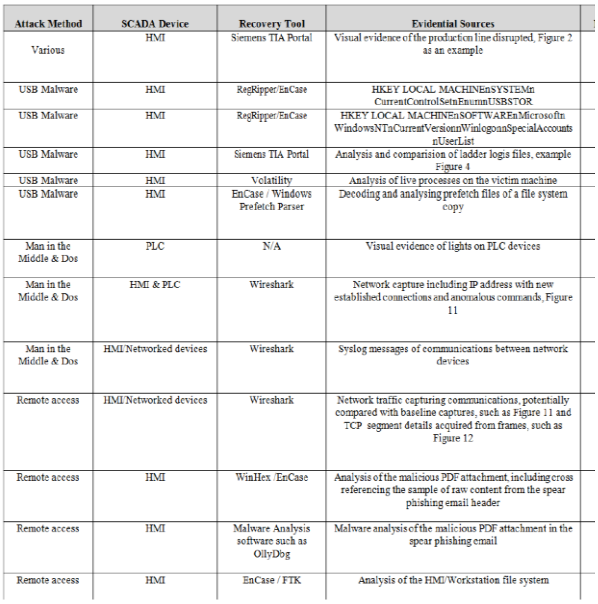
**Phase 3: Volatility Assessment, Contamination Impact Analysis, and Preservation, Prioritising and Collection**

Referring to the prioritisation list in the previous step, work from highest priority capture to lowest e.g. data which is most volatile and vulnerable to loss. This will more than likely be PLC or RTU devices, followed by network devices, and workstations. Critical systems and components of the system cannot be shut down to avoid disrupting operations; therefore live forensic investigations will take place. Location of devices relevant to the investigation, and using the forensic acquisition tools in the toolkit, to capture the data using hashing and verification techniques. Devices include engineering workstations, HMI devices, network devices producing traffic and any other attached devices, such as the USB device used to distribute malware. Ensure that the data is stored securely using storage HDD's, and logged in the evidential notes within text editor software, such as Notepad++. A packet capture is created on the network using Wireshark and the data is treated as evidential data, which can be hashed and verified via the use of software in order to verify its credibility.

**Phase 4: Examination & Phase 5: Analysis**

The analysis phase consists of finding relationships between the recovered forensic artefacts and piecing the evidential data together. This is performed in order to develop a timeline of the incident and its impact on the control environments. As SCADA is relatively unique, the examination process should also include engineering representatives who are familiar with the operation of the system. Evidence artefacts found within the investigations are listed for each of the case study attacks.

Table 1. SCADA Devices and Data Examples



| Attack Method | SCADA Device | Recovery Tool | Evidential Sources |
|---|---|---|---|
| Various | HMI | Siemens TIA Portal | Visual evidence of the production line disrupted, Figure 2 as an example |
| USB Malware | HMI | RegRipper/EnCase | HKEY LOCAL MACHINEnSYSTEMn CurrentControlSetnEnumnUSBSTOR |
| USB Malware | HMI | RegRipper/EnCase | HKEY LOCAL MACHINEnSOFTWAREnMicrosoftn WindowsNTnCurrentVersionnWinlogonnSpecialAccounts nUserList |
| USB Malware | HMI | Siemens TIA Portal | Analysis and comparision of ladder logis files, example Figure 4 |
| USB Malware | HMI | Volatility | Analysis of live processes on the victim machine |
| USB Malware | HMI | EnCase / Windows Prefetch Parser | Decoding and analysing prefetch files of a file system copy |
| Man in the Middle & Dos | PLC | N/A | Visual evidence of lights on PLC devices |
| Man in the Middle & Dos | HMI & PLC | Wireshark | Network capture including IP address with new established connections and anomalous commands, Figure 11 |
| Man in the Middle & Dos | HMI/Networked devices | Wireshark | Syslog messages of communications between network devices |
| Remote access | HMI/Networked devices | Wireshark | Network traffic capturing communications, potentially compared with baseline captures, such as Figure 11 and TCP segment details acquired from frames, such as Figure 12 |
| Remote access | HMI | WinHex /EnCase | Analysis of the malicious PDF attachment, including cross referencing the sample of raw content from the spear phishing email header |
| Remote access | HMI | Malware Analysis software such as OllyDbg | Malware analysis of the malicious PDF attachment in the spear phishing email |
| Remote access | HMI | EnCase / FTK | Analysis of the HMI/Workstation file system |

## 6.1. USB Malware Attack

Social Engineering: Depending on the method of social engineering, such as spear phishing or physical access, a number of traces which can be found may likely be tangible such as the USB device or CCTV footage. Acquiring a bit-for-bit copy of a file system from a workstation and performing examination through package software such as Encase will allow an investigator to find browsing history, email artefacts and other evidential data relating to a spear phishing attack.

USB Device traces: Once the USB device is attached, changes are made to the HMI, including the USBSTOR located in the SYSTEM registry hive and file system changes. Various USB devices which have been attached to the machine can be identified, and the information available includes Vendor ID (VID) and Product ID (PID), the serial number of the device that can be used to match the mounted drive letter, user and the first and last connected times of the device [45]. Other locations of interest are: MountedDevices, MountPoints, the USB key in the SYSTEM hive and the setupapi log. However, there are scenarios when the USB does not interact with the system this way, which is where devices using the Media Transfer Protocol (MTP) are introduced [45], and is explained in further detail by Ibrahim [46], [47]. Analysis of USB devices can be parsed by tools such as Internet Evidence Finder (IEF), or RegRipper [47]. This information is useful within a SCADA environment if the system is air gapped, and can provide information to support a chain of events.

Hidden User Account traces: The account created by the adversary does not appear on the login user list or in the Control Panel list; thus hidden from an operator. However, it will appear in the UserList in the registry which can be parsed by RegRipper, providing more evidential data on "who" made the attack.

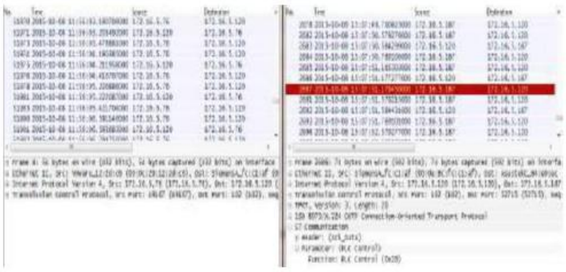
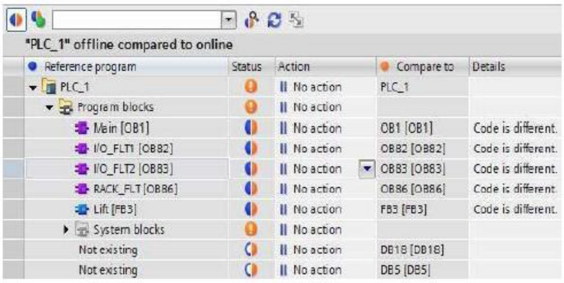


Figure 8.Two network traffic captures from a HMI, in Wireshark

Network Traffic: It would be difficult to obtain network traffic during the event unless monitoring and logging is in use and is available for analysis. A Wireshark capture of network data is taken during the experiment in order to display the types of traffic which may be available and can be found utilising this tool during an attack. Figure 8 displays baseline network traffic on the left compared to abnormal traffic highlighted on the right using Wireshark. In this example, timestamps are displayed alongside source and destination IP addresses and ports, and the protocol used which provides an indication of which devices are communicating. On the right side of the image, the abnormal network traffic highlights the adversary machine communicating with the PLC, utilising the S7 proprietary protocol, ISO 8073/x.224 Connection-Oriented Transport Protocol (COTP) and TPKT to issue commands to the PLC. Baseline information is incredibly useful; if an investigator is not familiar with the network, it allows them to identify any further resources of evidential data which may have been missed.

Screenshots: The screenshots captured of the victim ma-chine cannot be seen on network traffic analysed by Wireshark, as Meterpreter pipes all information through an SSL/TLS tunnel and is fully encrypted [48]. Only the initial payload can be seen in the capture. Evidence of screenshots could be found in the file system of the adversary machine by using EnCase or FTK if the images have been stored and access is available.

Logic Comparisons: One of the most SCADA specific features which may be found during an investigation is logic block comparisons within vendor software. Figure 12 displays an example in the Siemens TIA Portal of two pieces of logic compared in multiple block segments, which have been identified as having differing code. This indication allows an investigator to analyse these code blocks for any malicious modifications, and to pinpoint any changes made to the system and devices. Liaising with specialist SCADA operators during analysis of this evidence type is highly useful. The logic block comparison feature may be available in most SCADA vendor software, and detailed comparisons can be printed out which provide logic diagrams and breakdown of the code. This information requires further evidential artefacts for support, such as from Wireshark or EnCase, as this software is not forensic based and does not provide sufficient evidence alone.



"PLC_1" offline compared to online

| Reference program | Status | Action | Compare to | Details |
|---|---|---|---|---|
| PLC_1 | | No action | PLC_1 | |
| Program blocks | | No action | | |
| Main [OB1] | | No action | OB1 [OB1] | Code is different. |
| I/O_FLT1 [OB82] | | No action | OB82 [OB82] | Code is different. |
| I/O_FLT2 [OB83] | | No action | OB83 [OB83] | Code is different. |
| RACK_FLT [OB86] | | No action | OB86 [OB86] | Code is different. |
| Lift [FB3] | | No action | FB3 [FB3] | Code is different. |
| System blocks | | No action | | |
| Not existing | | No action | DB18 [DB18] | |
| Not existing | | No action | DB5 [DB5] | |

Figure 9. Comparisons of logic blocks within Siemens Totally Integrated Automation Portal

Further tools can be used for the capture of volatile information, such as a hex viewer like WinHex to examine raw PLC data dumps that possibly contain ladder logic extracts. Command line tools may also be used if they are preferred by the investigator to display and configure basic network information. Data hashing tools are also used to preserve integrity and validity of evidence which most forensic tools have as a feature.

## 6.2. Man in the Middle Attack

Physical SCADA Device Indicators: Physical evidence may be present after attacks, one example includes lights on PLCs flashing abnormally. This may be interpreted as a faulty device so it is usually helpful to keep network logs for examination in case an operator removes or resets the device before an incident is discovered.

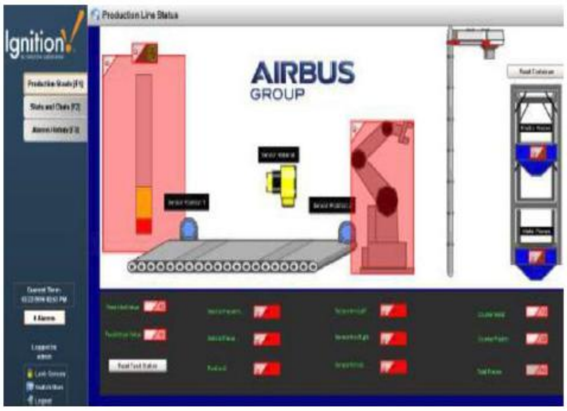
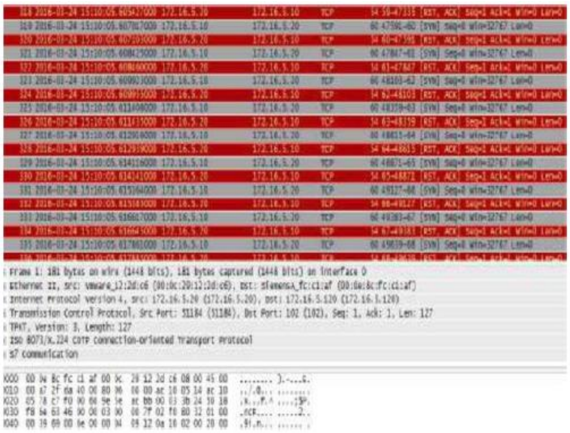


Figure 10. Visual indicator that the production line HMI has been disrupted

This attack mainly provides network evidence which can be found using Wireshark and looking for new established connections or anomalous commands sent to the PLC. Figure 10 shows the vendor software highlighted in red which may be displayed on a workstation after the attacks if the logic is changed and operations are disrupted.

Network Traffic: Comparisons of baseline and attack captures are useful during the investigation in this experiment. Before the attack, the HMI and PLC will communicate by sending ARP broadcasts to find the MAC address of the other device. During the attack, the adversary machine sends ARP packets telling the victim that the IP addresses belonging to other devices is associated to its own MAC addresses instead of the actual IP. Display and colour filters in Wireshark can be used to distinguish types of transmissions occurring over the network, including specific plugins for SCADA based protocols as shown in Figure 11.

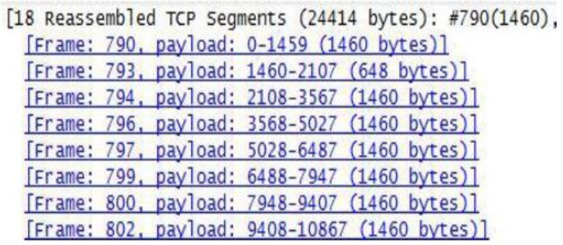


```
318 2016-03-24 15:10:05.605427000 172.16.5.20  172.16.5.10  TCP  54 59→47335 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
319 2016-03-24 15:10:05.607017000 172.16.5.10  172.16.5.20  TCP  60 47591→60 [SYN] Seq=0 Win=32767 Len=0
320 2016-03-24 15:10:05.607203000 172.16.5.20  172.16.5.10  TCP  54 60→47591 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
321 2016-03-24 15:10:05.608425000 172.16.5.10  172.16.5.20  TCP  60 47847→61 [SYN] Seq=0 Win=32767 Len=0
322 2016-03-24 15:10:05.608660000 172.16.5.20  172.16.5.10  TCP  54 61→47847 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
323 2016-03-24 15:10:05.609903000 172.16.5.10  172.16.5.20  TCP  60 48103→62 [SYN] Seq=0 Win=32767 Len=0
324 2016-03-24 15:10:05.609915000 172.16.5.20  172.16.5.10  TCP  54 62→48103 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
325 2016-03-24 15:10:05.611404000 172.16.5.10  172.16.5.20  TCP  60 48359→63 [SYN] Seq=0 Win=32767 Len=0
326 2016-03-24 15:10:05.611415000 172.16.5.20  172.16.5.10  TCP  54 63→48359 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
327 2016-03-24 15:10:05.612916000 172.16.5.10  172.16.5.20  TCP  60 48615→64 [SYN] Seq=0 Win=32767 Len=0
328 2016-03-24 15:10:05.612939000 172.16.5.20  172.16.5.10  TCP  54 64→48615 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
329 2016-03-24 15:10:05.614116000 172.16.5.10  172.16.5.20  TCP  60 48871→65 [SYN] Seq=0 Win=32767 Len=0
330 2016-03-24 15:10:05.614141000 172.16.5.20  172.16.5.10  TCP  54 65→48871 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
331 2016-03-24 15:10:05.615364000 172.16.5.10  172.16.5.20  TCP  60 49127→66 [SYN] Seq=0 Win=32767 Len=0
332 2016-03-24 15:10:05.615383000 172.16.5.20  172.16.5.10  TCP  54 66→49127 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
333 2016-03-24 15:10:05.616607000 172.16.5.10  172.16.5.20  TCP  60 49383→67 [SYN] Seq=0 Win=32767 Len=0
334 2016-03-24 15:10:05.616645000 172.16.5.20  172.16.5.10  TCP  54 67→49383 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
335 2016-03-24 15:10:05.617861000 172.16.5.10  172.16.5.20  TCP  60 49639→68 [SYN] Seq=0 Win=32767 Len=0

Frame 1: 181 bytes on wire (1448 bits), 181 bytes captured (1448 bits) on interface 0
Ethernet II, Src: Vmware_12:2d:c6 (00:0c:29:12:2d:c6), Dst: SiemensA_fc:c1:af (00:0e:8c:fc:c1:af)
Internet Protocol Version 4, Src: 172.16.5.20 (172.16.5.20), Dst: 172.16.5.120 (172.16.5.120)
Transmission Control Protocol, Src Port: 51184 (51184), Dst Port: 102 (102), Seq: 1, Ack: 1, Len: 127
TPKT, Version: 3, Length: 127
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
S7 Communication

0000  00 0e 8c fc c1 af 00 0c  29 12 2d c6 08 00 45 00   ........ ).-...E.
0010  00 a7 2f da 40 00 80 06  00 00 ac 10 05 14 ac 10   ../.@... ........
0020  05 78 c7 f0 00 66 9e 5e  ac bb 00 03 3b 24 50 18   .x...f.^ ....;$P.
0030  f8 64 63 46 00 00 03 00  00 7f 02 f0 80 32 01 00   .dcF.... .....2..
0040  00 19 69 00 00 6e 00 00  04 09 12 0a 10 02 00 20 00   ..i..n.. .......
```

Figure 11. Traffic between HMI and adversary machine during launched DoS attack

## 6.3. Remote Access Attack

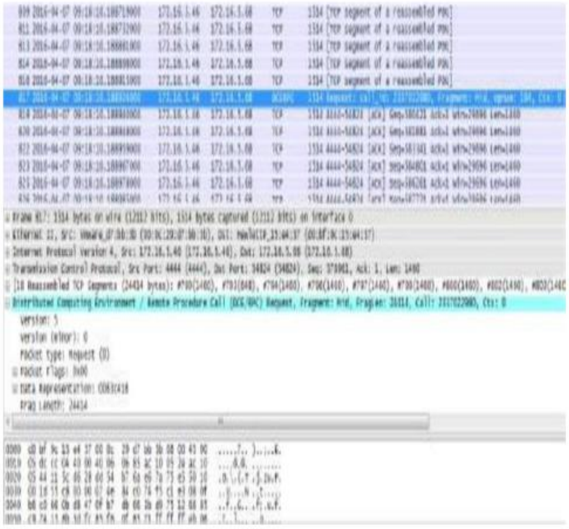
Email Traces: Presence of the spear phishing email with the attached malicious file can be found on browser history using a copy of the HMI file system analysed in EnCase or FTK. Evidence may include looking at the raw content of the email for cross referencing between other evidence. Raw content including the email header will provide information such as the sender's IP address, protocols used, timestamps and a sample of the malicious PDF's encoded format. Dates and times from the email itself can start to provide a rough outline for a chain of events. Attached File Analysis: This is using WinHex or EnCase for analysis of the PDF itself once a copy of it has been obtained. Malware analysis of the file, such as using OllyDbg5, can also be utilised in order to reverse engineer any hidden executables.



```
[18 Reassembled TCP Segments (24414 bytes): #790(1460),
  [Frame: 790, payload: 0-1459 (1460 bytes)]
  [Frame: 793, payload: 1460-2107 (648 bytes)]
  [Frame: 794, payload: 2108-3567 (1460 bytes)]
  [Frame: 796, payload: 3568-5027 (1460 bytes)]
  [Frame: 797, payload: 5028-6487 (1460 bytes)]
  [Frame: 799, payload: 6488-7947 (1460 bytes)]
  [Frame: 800, payload: 7948-9407 (1460 bytes)]
  [Frame: 802, payload: 9408-10867 (1460 bytes)]
```

Figure 12. Reassembled TCP segment details from frame 817

Network Traces: Anomalous communication to the victim machine will be present within the network which may be recorded if logging is enabled. If not, a network capture may be possible although it depends on the type of attack and the amount of time after the incident has occurred. Figures 12 and 13 show traces of network traffic provided information around the attack.



```
Frame 817: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: Vmware_07:bb:3b (00:0c:29:07:bb:3b), Dst: ...
Internet Protocol Version 4, Src: 172.16.1.46 (172.16.1.46), Dst: 172.16.1.68 (172.16.1.68)
Transmission Control Protocol, Src Port: 4444 (4444), Dst Port: 54824 (54824), Seq: 37993, Ack: 1, Len: 1460
[18 Reassembled TCP Segments (24414 bytes): #790(1460), #793(648), #794(1460), #796(1460), #797(1460), #799(1460), #800(1460), #802(1460), #803(146
Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fragment: Mid, FragLen: 31014, Call: 2317022080, Ctx: 0
  Version: 5
  Version (minor): 0
  Packet type: Request (0)
  Packet Flags: 0x00
  Data Representation: 00010418
  Frag Length: 24414
```

Figure 13. Connections from the adversary machine

## 7. Evaluation

The methodology and toolkit have both been independently reviewed against common attack scenarios. Testing the methodology as defined by Stirland et al [5], consists of deploying common attack scenarios against a SCADA production line testbed, and using available forensic tools to gather evidence. With reference to the experiments in Section V, by following the phases in the SCADA methodology and using the toolkit, examples of relevant evidence have been acquired, therefore supporting that the methodology and toolkit are both suitable for forensic investigations. By utilising attack scenarios, evidence has been found to support the forensic question of who, what, why and how elements of when an incident occurs as seen in Section VI, supported by Table 1. However, as proven by the results, evidential acquisitions relate to the SCADA methodology, which is inclusive of all bespoke, live, and shutdown devices.

The methodology covers all aspects of a SCADA environment compared to standard forensic models and previous SCADA methodologies which include as little as three phases [1]. However, to be effective during incident response, the methodology will need SCADA specific tools to recover as much information as possible.

The SCADA forensic toolkit requires further tailoring to bespoke devices in order to support the methodology properly. One example of data that could not be utilised includes data extracted from PLCs directly, which provide rich sources of evidence to support SCADA incidents.

The toolkit needs SCADA specific tools to recover as much information as possible from bespoke SCADA and legacy devices in order to further develop the methodology. Further work is to be conducted around SCADA data PLC, and developing bespoke software that the defined commercial toolkit does not feature, for example it is not possible to capture memory from PLCs.

## 8. Conclusion

Development of a methodology and toolkit is crucial in order to support the protection of Critical National Infrastructure and other core services, such as communication and manufacturing environments, from incidents which hinder production or cause a large scale societal impact.

The challenges and threats that exist within SCADA environments and unique requirements of the systems, such as critical availability, have been addressed in Section II along with the importance of distinguishing traditional and bespoke enterprise devices and systems, which need to be considered within state of the art research in order to prepare forensic investigators.

Past threats and case studies, such as Stuxnet and Flame, have been considered to create attacks against SCADA testbeds in order to create a realistic scenario for the methodology and toolkit to be applied against. Developing a methodology is important in order to keep the digital forensics process up to date with new laws, technologies and methods of attack, especially as there is increasing connectivity between SCADA and enterprise networks.

The methodology proposed in Section III provides phases specifically considering the requirements and components within SCADA systems for forensic investigations, including workstations and PLCs. In this paper, we have described the implementation of a SCADA methodology and toolkit after common cyber-attacks have been conducted. The evidence from each attack has been referenced to each tool used and as related to methodology steps taken. The data captured provides an insight into the traces that are left behind by common attacks, the extent to which traces can be found on varying devices and systems, the operations that are affected and traces of how the incident occurred.

Table 1 in Section V provides an overview of what types of evidence were able to be retrieved using the toolkit, and supporting that the methodology is efficient for SCADA infrastructures. By using a consistent approach, testable and repeatable results are produced providing identification of critical issues and requiring further development, such as the lack of forensic based SCADA tools. A number of commercially available and open source forensic tools can be applied to a SCADA system in order to acquire evidential artefacts. Further evidence may be discoverable which cannot be found by using current forensic tools. Testing existing tools provides an insight into the current lack of SCADA specific forensic tools, and the challenges which the development of new tools must consider, such as critical availability of systems, the integrity of volatile data, and acquiring data from bespoke devices.

Applying the methodology and toolkit throughout the representative experimental use cases contributes to the development of incident response which can be used within industrial environments, including production and testing facilities. As a result, the methodology aids future preparation within forensic practice, as well as the identification of the defence against SCADA based cyber attacks. Testbeds which include a selection of devices used in real systems are also preferable for evaluation of new methodologies and tools, as failures are contained and do not risk lives. Whilst security is a crucial issue within SCADA environments, it will never completely mitigate malicious attacks undertaken by adversaries, including external and internal entities.

Therefore, developing understanding of how an incident initially occurred will enable the forensic community to support security front lines in order to compete with attackers. There is not just one technological solution to SCADA vulnerabilities, but rules and guidelines, including training and software procedures to follow when SCADA security has been compromised. Sharing knowledge of new threats and ensuring product vendors and users response promptly to such issues is key to managing the risk in future.

## 9. Future Work

Future work includes developing SCADA specific forensic tools, and further research into the challenges within incident response such as the increase of devices, Big Data, the Internet of Things, increasing storage space, connectivity, and use of mobile technology. It is crucial that these issues are addressed primarily due to the utilisation of legacy devices, systems, and protocols which are currently in operation and work in cooperation with increasingly sophisticated enterprise tools and devices.

## 10. References

[1] I. Ahmed, S. Obermeier, M. Naedele, and G. G. R. III, “Scada systems: Challenges for forensic investigators,” Computer, vol. 45, pp. 44 – 51, December 2012.

[2] “Introduction to computer forensics.” Forensic Control, 2014.

[3] R. Hunt and S. Zeadally, “Network forensics: An analysis of techniques, tools, and trends,” 2012.

[4] M. Hentea, “Improving security for SCADA control systems,” Interdisciplinary Journal of Information, Know ledge, and Management, vol. 3,2008.

[5] J. Stirland, K. Jones, H. Janicke, and T. Wu, “Developing cyber forensics for SCADA industrial control systems,” in International Conference on Information Security and Cyber Forensics, (Proceedings of the International Conference on Information Security and Cyber Forensics, Kuala Terengganu, Malaysia), 2014.

[6] T. Wu, J. F. P. Disso, K. Jones, and A. Campos, “Towards a SCADA forensics architecture,” in Proceedings of the 1st International Symposium for ICS & SCADA Cyber Security Research, (EADS Innovation Works Quadrant House Celtic Springs, Coedkernew, Newport), EADS, BCS Learning and Development Ltd, 2013.

[7] “Forensic soundness.” Illurity Blog, August 2009.

[8] D. Fisher, “State of SCADA security ’laughable’, researchers say.”Threatpost Website, February 2012.

[9] A. Nicholson, S. Webber, S. Dyer, and H. Janicke, “SCADA security in the light of cyber-warfare,” in Computers & Security, no. 31, pp. 418 – 436, Elsevier, February 2012.

[10] J. Slay and M. Miller, “Lessons learned from the maroochy water breach,” in Critical Infrastructure Protection (E. Goetz and S. Shenoi, eds.), vol. 253, ch. 6, pp. 73 – 82, Boston: Springer, 2008.

[11] T. Kilpatrick, J. Gonzalez, M. P. R. Chamdia, and S. Shenoi, “An architecture for SCADA network forensics,” in Advances in Digital Forensics II (S. S. Olivier M, ed.), vol. 222, ch. 22, pp. 273 – 285, Springer, 2006.

[12] E. Byres, “No.1 ICS and SCADA security myth: Protection by air gap.” Tofino Security Blog, July 2012.

[13] P. Paganini, “New attacks against SCADA, old vulnerabilities, very old issues.” Security Affairs, January 2013.

[14] "The history of STUXNET: Key takeaways for cyber decision makers." Cyber Conflict Studies Association, June 2012. http://www.afcea.org/committees/cyber /documents/TheHistoryofSTUXNET.pdf.

[15] E. Byres, "Defense in depth is key to SCADA security - part 1 of 2." Tofino Security Blog, February 2012.

[16] I. O. Ademu, C. O. Imafidon, and D. S. Preston, "A new approach of digital forensic model for digital forensic investigation," in (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 2, 2011.

[17] "Definition: State of the art." http://www.oxforddictionaries.com/.

[18] S. Mustard, "Security of distributed control systems: the concern increases," IEE Computing and Engineering, vol. 16, pp. 19 – 24, January 2006.

[19] "USB rubber ducky."http://hakshop.myshopify.com /products/usbrubber-ducky-deluxe.

[20]"Metasploit penetration testing software." http://www.metasploit.com/.

[21] "Kali Linux." https://www.kali.org/.

[22]. M. Faisal, M. Ibrahim, STUXNET, DUQU and Beyond, International Journal of Science and Engineering Investigations, Vol. 1, Issue 2, March 2012, pp. 75 – 78.

[23]. M. Combs, Impact of the STUXNET Virus on Industrial Control Systems, XIII International forum Modern information society formation - problems, perspectives, innovation approaches5 - 10 September, 2012, St.-Petersburg.

[24]. P. Kerr, J. Rollins, C. Theohary, The STUXNET Computer Worm: Harbinger of an Emerging Warfare Capability, Congressional Research Service, Dec 9, 2010, pp. 1-9.

[25] T. Miyachi, H. Narita, H. Yamada, H. Furuta, Myth and Reality on Control System Security Revealed by STUXNET, SICE Annual Conference, 13-18 September, 2011, Tokyo, Japan, pp. 1537-1540.

[26] N. Falliere, L. Murchu, E. Chien, W32.STUXNET Dossier, Symantec Security Response, pp. 1-68.

[27]. Schouwenberg, R. (n.d.). STUXNET and Flame – burning ring of fire. Retrieved from Kaspersky Lab: https://tweakimg.net/files/upload/Suxnet_and_Flame.pdf

[28] "Client side exploits." Offensive Security.

[29]. ICS SANS. (2016, March 18). Analysis of the Cyber Attack on theUkrainian Power Grid - Defense Use Case. Retrieved from E-ISAC Electronic Information Sharing And Analysis Center: http://www.nerc.com/pa/CI/ ESISAC/Documents/EISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[30]. Dell. (2015). 2015 Dell Security Annual Threat Report. Von Dell SonicWALL Threat Research Team – SCADA Cyber attack threat finding: https://software.dell .com/docs/2015-dell-security-annual-threat-report-white-paper-15657.pdf abgerufen.

[31] "ICS-Cert monitor," Homeland Security, November/December 2015.

[32]. Schouwenberg, R. (n.d.). STUXNET and Flame – burning ring of fire. Retrieved from Kaspersky-Lab: https://tweakimg.net/files/upload/Suxnet_and_Flame.pdf

[33]. FireEye. (01. May 2016). Spear-phishing attacks-why they are successful and how to stop them. Von FireEye-Targeted Emain-based phising campaigns: https://www. google.co.uk/?gws_rd=ssl#q=Dell%E2%80%99s+2015+Annual +Security++ abgerufen

[34] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," tech. rep., University of California at Berkley, CA, Department of Electrical Engineering and Computer Sciences, 2011.

[35] A. Krennmair, "Cinderella: A prototype for a specification-based nids," August 2003.

[36] E. Byres, "Flame malware and SCADA security: What are the impacts?."Tofino Security Blog, May 2012.

[37] E. Byres, "Summing up STUXNET in 4 easy sections." Tofino Security Blog, March 2011.

[38] H. Dalziel, "The four amigos: STUXNET, flame, gauss and duqu." Concise Courses Blog, February 2013.

[39] R. Langner, "Cracking STUXNET, a 21st-century cyber weapon." TED, March 2011.

[40] "The mystery of duqu 2.0: a sophisticated cyberespionage actor returns." Global Research and Analysis Team, June 2015.

[41] M.-B. Samekh, "Lessons learned from flame, three years later." Securelist Blog, May 2015.

[42] R. Lipovsky and A. Cherepanov, "Blackenergy trojan strikes again: Attacks ukrainian electric power industry." We Live Security Website, January 2016.

[43] M. Abrams and J. Weiss, "Malicious control system cyber security attack case study - maroochy water services, australia," tech. rep., Mitre, July 2008.

[44] B. Miller and D. C. Rowe, "A survey of SCADA and critical infrastructure incidents," October 2012.

[45] J. McQuaid, "How to analyse USB device history in windows." Magnet Forensics, July 2014.

[46] N. Ibrahim, "Part 1: USB device research," September 2013. https://nicoleibrahim.com.

[47] N. Ibrahim, "USB devices and media transfer protocol: Identifying the existence of data exfiltration

artifacts." SANS DFIR Summit Presentation, G-C Partners, LLC.

[48] "Packet sniffing." Offensive Security Website.