EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems

Research Article ICST.ORG

Attribution of Cyber Attacks on Industrial Control Systems

Allan Cook[1], Andrew Nicholson[2], Helge Janicke[1], Leandros Maglaras[1,*], Richard Smith[1]

[1]Cyber Security Centre, De Montfort University, Leicester, LE1 9BH, UK
[2]Cyber Security Centre, WMG, University of Warwick, Coventry CV4 7AL, UK

Abstract

In order to deter or prosecute for cyber attacks on industrial control systems it is necessary to assign attribution to the attacker and define the type of attack so that international law enforcement agencies or national governments can decide on appropriate recourse. In this paper we identify the current state of the art of attribution in industrial control systems. We highlight the critical differences between attribution in enterprise networks and attribution in industrial networks. In doing so we provide a roadmap for future research.

,FZXPSETBUUSJCVUJPOTDBEBJOEVTUSJBMDPOUSPMTZTUFNTTVSWFZDZCFSBUUBDLT

Received on 8 January, 2016; accepted on 10 February, 2016; published on 21 April, 2016

doi: 10.4108/_____

1. Introduction

Industrial Control Systems (ICS) are increasingly becoming the subject of computer network attacks [53]. These systems provide essential services for sovereign nations critical infrastructure, and as such these attacks represent a significant threat to the continued security of these countries [39]. ICS have performance and reliability requirements that may be considered unconventional by contemporary IT professionals. These requirements include the management of processes that, if not executed correctly, pose a significant risk to the health and safety of human lives, serious damage to the environment, as well as serious financial issues such as production losses that may have a negative impact on a nation's economy [72].

2. Contribution

We present the first survey of technical attribution techniques specifically in relation to ICS. Previous attack taxonomies used for contemporary attribution do not accommodate methods to integrate data from these cyber-physical systems (CPS). Our paper collates research into one self-containing attribution resource that is useful for new researchers to the field. We identify promising areas for future work, particularly that a combination of techniques offers the potential to build a probabilistic model that may improve overall attack attribution.

*Corresponding author. Email: leandros.maglaras@dmu.ac.uk

3. Motivations for Attribution in Critical National Infrastructure

In this paper we discuss fundamental aspects of attribution of cyber attacks when considering industrial control systems. To the best of our knowledge the information is disparate and not self-contained, hence providing motivation for our paper. We consider technical and non-technical issues including current legal precedent and standards that will shape future direction of this field.

The subject of attribution of cyber attacks targeted at industrial control systems (ICS) is an emerging issue. Zhu et al (2011) [84] describe how these systems are "deeply ingrained in the fabric of critical infrastructure", but subject to disruption or damage by cyber effects. They describe in particular, the potential for cyber-physical attacks, where the impact of cyber attacks can result in outcomes in the physical world. Miller and Rowe (2012) [47] described examples of these physical outcomes in a range of incidents between 1982 and 2012.

In order to prosecute in response to cyber attacks on industrial control systems it will be necessary to assign attribution to the attacker, and define the type of attack, so that international law enforcement agencies or national governments can decide on appropriate recourse. Attribution serves to act as a deterrent to future attacks, can provide the basis for interrupting attacks in progress and can support overall improvements to defensive techniques [32].

3.1. Defining Attribution

Attribution of cyber attacks lacks a universally accepted definition. Proposed definitions have often been limited in their approach, confining each to subsets of attribution. For example, definitions offered by Hunker et al [32] limits attribution to "any attribution technique that begins with the defending computer and recursively steps backward in the attack path towards the attacker". Wheeler et al [79] defined attribution as "determining the identity or location of an attacker or an attacker's intermediary".

3.2. Legal Requirements for Attribution

Before considering the techniques available for attack attribution it is necessary to understand the legal requirements for the prosecution of a cyber attack and the role that attribution plays.

Brenner [10] described the legal requirements for attribution as answering two fundamental questions; firstly, who carried out the attack and, secondly, what kind of an attack was it? The former assigns responsibility for committing an act, the latter assigns responsibility for responding to an attack. With regard to the responsibility for committing an act, Keyser [36] highlighted the adoption of the Council of Europe Convention on Cybercrime as the de facto standard for transnational cyber crime prosecution framework for Western European counties and the North Americas by harmonising local laws. He discussed Article 5 of the treaty, relating to "system interference"and its aim to prevent the intentional "hindering"of the functioning of a computer system by interfering with, or manipulating, computer data without right. It continued with a discussion of violations under the article and the requirement for a demonstration of *mens rea* (guilty mind), although he cites that the definition of intentional action remains an unresolved issue and has been treated differently in signatory countries. Therefore, simply tracing an attack on an ICS to its source will not necessarily result in sufficient evidence for a prosecution. Any technical facts must be supported by a motive or intent.

With regard to responsibility for responding to an attack, Brenner [10] touched upon national jurisdictions and the transnational nature of cyber crime. Keyser [36] described how cyber criminals and malicious actors either base their operations in counties outside of legal frameworks such as the Convention on Cybercrime, or route their traffic through such countries. Kohl [40] discussed how a response to a cyber attack then becomes a question of which country or law enforcement agency has the responsibility and authority to investigate, under which legal framework the perpetrators can be prosecuted, and which laws apply.

This transnational issue was explored more recently in the Tallinn Manual on the International Law Applicable to Cyber Warfare [64] when discussing the acts of a nation-state. Rule 6 described how a nation-state "bears international legal responsibility for a cyber operation attributable to it", but recognised that the location from which the attack took place does not necessarily define whether that nation-state is responsible. It described a scenario in which Nation-State A, under the instructions of Nation-State D created a botnet in Nation-State B to attack targets in Nation-State C (as illustrated in Figure 1). Under these conditions the Tallinn Manual defined that Nation-State B could not be held responsible for the attack, and that Nation-State D, from which the intent was derived, was attributable for the actions. The involvement of Nation-State A was discussed as a less well-defined area as it could not be presumed responsible based on the fact that the attack traffic originated from there. It again, required a measure of mens rea to determine legal responsibility.

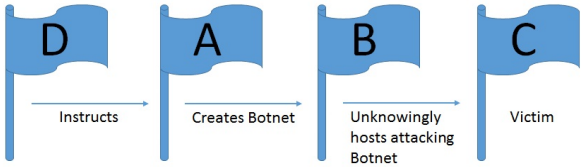

Figure 1. An illustration of the complexity of nation-state responsibility in attribution

It therefore becomes apparent that, in light of the ambiguities in international laws, methods of attributing the execution of an attack must include not only the technical reconstruction of the attack path to the source device, but also a means by which the intent of the perpetrator is elicited. Similarly, in order to support the decision of who should respond to an incident, a taxonomy of types of attack is required to allow an international understanding of the nature and impact of cyber effects. The significance of this assessment rises in priority if the industrial control system attack results in loss of life or significant impact on a nation-state.

In light of this risk, it is perhaps more practical to focus on deterrence rather than prosecution. Libicki [44], discussing cyberattacks in the context of cyberwarfare, argued "cyberattacks can be launched from literally anywhere, including cybercafes, open Wi-Fi nodes, and suborned third-party computers. They do not require rare or expensive machinery. They leave no physical trace. Thus, attribution is often guesswork. True, ironclad attribution is not necessary for deterrence as long as attackers can be persuaded that their actions may provoke retaliation. Yet some

proof may be necessary given (1) that the attacker may believe it can shake the retaliator's belief that it got attribution right by doing nothing different ("who, me?") in response to retaliation, (2) that mistaken attribution makes new enemies, and (3) that neutral observers may need to be convinced that retaliation is not aggression"[44].

While reporting of security breaches is on the rise [33], little data is available to identify the sources of such attacks. The problem of attribution of cyber effects in general is a well documented issue, yet little has emerged from academic or industry research to satisfy the legal requirements for accuracy to support prosecution of the attack originators [15]. The techniques available to attackers to obfuscate their location and route to target introduce too much uncertainty in a court of law, or at the least to act as a deterrent [27].

4. Challenges of ICS Attribution

Attribution of cyber attacks in ICS environments is a significant challenges when compared with attribution of cyber attacks in enterprise environments. In this section we begin by identifying those differences and what this means for attribution.

4.1. Attribution and Architecture

ICS differ from traditional IT architectures in that they are generally not all IP-enabled, and incorporate a number of proprietary or industry-specific protocols based upon serial or bus communications. Even when IP is used, the performance requirements necessitated the use of modified IP stacks or optimised routers that limit the level of auditing and inspection available. These protocols are deployed at differing layers of the architecture and often require gateways for interoperability [24]. This heterogeneous communications environment services a number of measurement and control devices, and are often in service for 10-20 years [49] [4] running the same operating systems, and operate with limited computing capacity, designed for performance and reliability rather than security [12] [83].

4.2. Enterprises and Industrial Control Systems

ICS is a general term that encompasses a family of process automation technologies, including Supervisory Control and Data Acquisition (SCADA) systems and Distributed Control Systems (DCS). These control systems use Programmable Logic Controllers (PLC) or similar Remote Terminal Units (RTU) and Intelligent Electronic Devices (IED) to manage electromechanical equipment in either local or distributed environments. Their application covers a range of industrial sectors and critical infrastructures such as electricity generation and distribution, water treatment and supply, oil refining, food production and logistics [51]. These control systems provide automation and process control of the systems that provide the reliable flow of products and services necessary for the security and operations of industrialised nation-states [48].

Table 1. PLC vs. General-Purpose Computer [60] [20]

PLC	Computer
Ruggedised design for industrial environments	Designed mainly for data processing and calculation
Ability to operate in high temperatures and humidity	Limited environmental range
High immunity to signal noise	Optimised for speed
Integrated proprietary command interpreter	Support for multiple development environments
Limited memory	Significant and expandable memory
Optimised for single-thread processing	Multitasking capability

As an example of the differences between ICS and conventional enterprise IT, Table 1 compares a PLC to a generalised IT computer.

When considering the diversity of industrial control systems it is helpful to have a common framework in which to model the common aspects of such systems, and the levels of process hierarchy that exists. Williams [81] described the Purdue model, a reference architecture for control hierarchy that has become the standard within ICS [84]. It described six levels within an organisation managing an industrial control system, as illustrated in Figure 2.

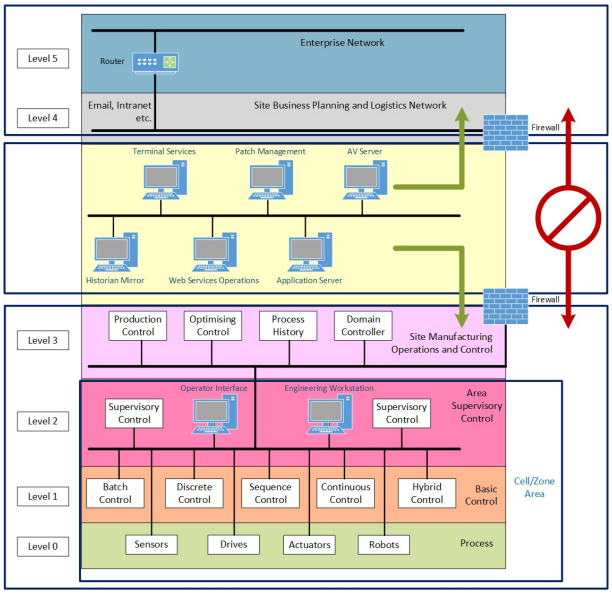

Figure 2. Purdue Model for Control Hierarchy[18]

Level 5 describes the corporate or enterprise network of an organisation running its business management applications and services. Internet access exists within this layer. **Level 4** shows the services to manage the planning, scheduling and logistics of the operations. **Level 3** encompasses the management of the day-to-day industrial operations of the facility, including production scheduling, quality assurance, process optimisation etc. **Level 2** provides supervisory control of the equipment involved in the overall industrial process. **Level 1** encapsulates the control of individual devices and equipment involved in discrete elements of the overall process (PLC, RTU, IED etc.) **Level 0** includes the devices, sensors and associated equipment performing the industrial process.

Whilst the Purdue reference model is not used to govern ICS implementations, it reflects the general architectural principles adopted whereby the control of industrial equipment is managed in a layered hierarchy that is logically, if not physically, separated from the management of the industrial facility and its business processes. Importantly, it defines the areas of an industrial control architecture where IP-based protocols transition to legacy serial communications.

4.3. Importance of ICS Artifacts for Attribution

In the case of a cyber attack on an ICS it is likely that there will be some real-world physical manifestation of the misuse. In the worst cases this could result in damage, injury, environmental impact or loss of life. In these instances, where it is probable that some legal or regulatory investigation would be required, the importance of attribution artefacts increases. It would be necessary to identify whether the behaviour of the ICS was caused by an error in facility operations, a failure of a safety device, or whether the processes and devices were maliciously manipulated to achieve the end result.

These artefacts, ideally, should be in a form whereby their authenticity can be guaranteed and track traffic and ICS commands through the entire operational process, ensuring end-to-end integrity. This assurance should include the logs of the devices and components that controlled the industrial equipment involved.

5. Review of Attribution Taxonomies

Studies into the spectrum of attribution techniques for attacks on ICS are limited at present. A taxonomy of attribution techniques for cyber attacks by Nicholson et al [52] provides an overview of the technical options available and classifies their attributed and practicalities. An initial investigation into attribution in SCADA systems, also by Nicholson et al [51], investigates five known attribution techniques and discusses their viability within an ICS environment.

Researchers have surveyed individual technical approaches to attribution, including; traceback - where the traffic from a target device is recursively stepped-back through its routing path to its originating source, and honeypots - where vulnerable software and services are hosted in order to allow activities to be monitored. Kuznetsov et al [42] evaluated four traceback approaches. Their criteria evaluated the number of packets required, complexity, robustness and ease of deployment. They found that current approaches are at a disadvantage as they need a large number of attack packets and require changes to Internet infrastructure. Kuznetsov et al [42] concluded that a better solution would be to embed traceback functionality within key Internet devices. Belenky and Ansari [6] proposed a framework for evaluating IP traceback systems. Their criteria for traceback qualities includes effects of partial deployment, processing and bandwidth overhead, memory requirements and scalability. They find that no traceback scheme is able to meet each criteria. Strayer [73] produced a taxonomy of stepping stone detection techniques. [26] evaluated five traceback techniques; Probabilistic Packet Marking (PPM), ICMP Traceback (iTrace), Deterministic Packet Marking(DPM), Source-Path Isolation Engine (SPIE) and a hybrid CenterTrack approach, classifying by factors such as computational overhead and robustness. Hamadeh and Kesidis [30] authored a taxonomy of Internet Traceback techniques, separating the problem into IP traceback, traceback across stepping stones, and worm traceback. Vincent and Raja [76] published a survey of IP traceback mechanisms, specifically looking at overcoming DoS attacks by using two types of IP traceback techniques; Packet Marking and Packet Logging, with an exploration of a hybrid of them both.

These taxonomies and surveys focus on specific families of techniques, which are each one category within the field of technical attribution, and therefore are limited in their approach when considering attribution holistically, these surveys miss important techniques when not accounting for all of them. Wheeler and Larsen [79] were the first to classify the landscape of technical attribution techniques and thus critique their combined merits. A number of taxonomies followed this approach. For example,Thing et al [74] reviewed a number of attribution techniques in the context of adaptive responses to DoS attacks. Blakely also considered traceback as a mechanism to identify cyber attackers by feature analysis [9].

The intent of an attack was explored by Duggan [19] and included an assessment of attacker capability. The research proposed a set of six generic threat actor profiles and their level of proficiency over seven characteristics, those being; available funding, determination, stealth, physical access to the target, software development skills, the perceived time it

would take to develop an effect, and finally the size of the organisation required to develop the effect. Barnum [5] described a more detailed model of capability. In this taxonomy the seven characteristics essentially decomposed to a lower level of granularity. Barnum also included an impact severity. Miller and Rowe [47], in a survey of SCADA and critical infrastructure incidents, included an impact of the attack, citing the outcomes of cyber effects on ICS as including disruption, distortion, destruction, disclosure or death. Fleury et al [23] do not cover motivation or intent, but proposed a framework based on an "attack-vulnerability-damage"(AVD) model.

Zhu et al (2011) [82] described a taxonomy of cyber attacks on SCADA systems and introduced the esoteric nature of ICS to the various methods of attack that these systems face. The research explained the focus on data integrity and availability within such systems, with a perceived reduced need (at least in the past) for confidentiality. It went on to offer examples of various attack surfaces and vectors, but did not offer a repeatable model for categorising and analysing attacks.

These taxonomies and surveys focus on specific families of techniques, which are each one category within the field of technical attribution, and therefore are limited in their approach when considering attribution holistically, these surveys miss important techniques when not accounting for all of them. Wheeler and Larsen [79] were the first to classify the landscape of technical attribution techniques and thus critique their combined merits. A number of taxonomies followed this approach. For example,Thing et al [74] reviewed a number of attribution techniques, as did Blakely [9].

None of the individual taxonomies reviewed offer the level of detail required in order to adequately define the intent, capability, level of exploitation and impact of an attack on an ICS, but there is merit in considering a fusion of various elements of them all in order to create a model of sufficient robustness to support an international definition of the outcome of an attack to allow agreement on ways and means to allocate responsibility and resources to investigate.

5.1. ICS and Attribution Problem Catalogue

As none of the taxonomies reviewed offer the level of detail required, it is therefore necessary to review individual tecniques. In order to assess the usefulness of an attribution technique to ICS we require a set of criteria by which the technique's effectiveness can be judged. The characteristics below have been used to measure the effectiveness of each attribution method in the context of an industrial system.

Performance: The ability to provide attribution functions without degrading ICS performance.

Reliability: The ability to provide attribution functions without adversely affecting the operating and safety processes of the ICS facility.

Extent: The ability to monitor traffic from originating source to the final end ICS device, including all protocol transformations en route, to provide a full picture of network behaviour.

Coherence: The ability to cross-reference traffic with ICS device behaviours through the synchronisation of device logs to permit inspection of command execution.

Identification: The ability to identify the attacker from behaviours or technical signatures.

Intent: The ability to determine the purpose of the attack, whether successful or otherwise, to provide suitable evidence and mens rea in order to support a prosecution.

5.2. Review of Attribution Techniques

Traceback. Traceback is a class of methods that encompasses techniques by which the traffic from a target device is recursively stepped-back through its routing path to its originating source device [63]. Figure 3 shows three paths that represent possible attack paths from suspected attackers. Traceback creates an attack graph showing the intermediate devices that the attack passed through.

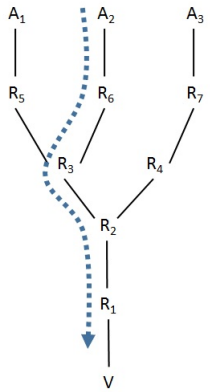

Figure 3. Theoretical Attack Graph [63]

Kuznetsov et al [42] aggregated the significant research in this area into three distinct approaches and evaluated their practicality. The first category included manual methods of traffic tracing, and required the routing device to support input debugging as well as constraining the period of analysis to the duration of

the attack itself. The second category spanned logging techniques, whereby routers persist information about the traffic they have encountered. These were described as impractical due to the storage requirements of such a mechanism. One variant, however, the Source Path Isolation Engine (SPIE) techniques of Snoeren et al [68], capable of tracing the route of a single packet through SPIE-compliant routers, could trace the addressed storage issues by only collecting hashes of the packets. While this reduced the storage overhead, Gao and Ansari [26] highlight that the computational requirements increased as a consequence. The third category included the various methods of probabilistic packet marking (PPM), and ICMP traceback (iTrace).

PPM, originally proposed by Savage et al [63], and extended by Song and Perrig [70] and Belinky and Ansari [6], used packet marking to sample a number of packets with path data so that should a target device receive a sufficient volume of such packets it could reconstruct the entire path back to the source. Marking information is stored in unused or infrequently used packet header fields, such as the 16-bit Identification field. Savage et al [63] suggest that 75 packets would be sufficient when the path length is 10 and the number of attackers is small. When the number of attackers is large, this technique becomes ineffective; thousands of packets are required and convergence time increases.

Song and Perrig [70] proposed Advanced and Authenticated Marking Schemes (AMS and AMS II). AMS advances upon the work of Savage et al by compressing entire traceback data into the Identification field. AMS II introduced authentication so that each router used a unique secret key to mark packets. Despite these modifications, the technique still remained weak against distributed denial of service attacks (DDoS) and spoofing. Goodrich [29] proposed Randomize-and-Link which aimed to counter these weaknesses. This technique used large checksums to link packets across a wide spectrum meaning an attacker's chance of spoofing was minimised. Finally, Belenky and Ansari [6] proposed Deterministic Packet Marking (DPM), which aimed to stop spoofing and allow low packet quantity.

iTrace, first proffered by Bellovin et al [7], generated out-of-band ICMP messages containing the same IP destination address as well as the IP header of the traced packet. It also included the IP address of the incoming and outgoing interfaces. As long as the target victim device received enough of these messages it could reconstruct the attack path, although this was reliant on the proper handling of ICMP traffic at all stages of the traffic route. Kim et al [37] highlighted the dependency on this method on correct BGP routing paths, and the inadequacies of BGP authentication and monitoring of changes. They proposed an augmented iTrace method whereby AS-PATH and link connectivity data was also included in the message in order to facilitate correct validation of routing through autonomous systems.

Traceback methods typically require a modification to the network infrastructure over which they will operate, and it is questionable how cost-effective this would be given the scale of the modern Internet. More important, however, is that all traceback techniques, including a hybrid model proposed by Korkmaz et al [41], fail to address the nature of contemporary, multi-stage attacks described by Clark and Landau [14] whereby intermediary devices are coerced by malware to infiltrate one computer to use as a platform to attack a second etc., in an ongoing process of originator obfuscation. At best, they will only attribute the attack to a coerced device.

Traceback techniques suffer from a number of problems that mean deployment in the Internet environment is unlikely. Traceback techniques provide direct artefacts such as the source IP address, however since IP addresses may be associated with compromised machines, this is of little use. It is only useful if the owner of the IP address endpoint is willing to fully cooperate and allow forensic investigation of their machine(s). Traceback is intrusive, requiring infrastructure changes for deployment and packet/router modifications, additional traffic or additional storage and processing requirements. Furthermore, the onus of who should manage these aspects is unclear. Finally, traceback techniques may introduce new attack vectors. For example, packet logging produces additional traffic and could cause a DDoS attack in itself.

Traceback is considered against our assessment criteria below:

Performance: Traceback functions require mechanisms to capture and analyse traffic, sometimes including router modification. This would introduce a level of latency that is likely to be unacceptable to an ICS operator.

Reliability: Traceback functions would require he introduction of new elements in the ICS safety chain that would require certification, and would probably not be certified to operate within known boundaries.

Extent: Currently traceback mechanisms only support IP traffic. ICS include a number of non-IP-based protocols that existing approaches would not support.

Coherence: Currently traceback only focuses on traffic, not associated log analysis.

Identification: Without modification of the infrastructure of the internet, traceback would not provide a means by which end-to-end traffic could be accurately monitored.

Intent: Whilst the purpose of the traffic could be assessed, traceback does not currently provide a suitable evidence chain for a prosecution.

Honeypots. Honeypots approach the issue of attribution of attacks differently to Traceback methods, by observing an attack in situ. A honeypot is a system, or set of systems, where vulnerable software and services are hosted in order to allow activities to be monitored and logged.

Franz and Pothamsetty (2004) [57], created the first publicly acknowledged SCADA honeypot. Their goal was to determine the feasibility of building a software framework to simulate a variety of industrial networks and devices. They found that there was a general lack of information relating to SCADA vulnerabilities and attacks. A technical deliverable was produced, a SCADA honeypot based on a low interaction honeypot, Honeyd. Honeyd simulated many network protocols such as HTTP, SMTP and FTP. Honeyd could be extended to simulate more network protocols using simple scripts. Franz and Pothamsetty created scripts to simulate the SCADA functionality of a Modicon Quantum device with HTTP, FTP, Telnet and Modbus services. They also created a Java applet, "StatusApplet.java", which could be accessed via a web server and simulated the status of a SCADA field device. The technical implementation of this honeynet was primitive and at a proof of concept stage. Subsequently little effort was placed on concealing the honeypot status. For example, the action event on the HTML forms reads 'action="honeyd-feedback.py"', an indicator that the SCADA system is actually a honeypot.

Researchers at Digital Bond expanded upon the work of Franz and Pothamsetty when they released two VMWare virtual images [77]. One image contained a SCADA honeypot based on Franz and Pothamsetty's work and another image contained Honeywall to monitor activity, collect data and prevent outbound connections from compromised honeypots. Digital Bond also included their Quickdraw rules, a collection of Snort intrusion detection system (IDS) preprocessors and plugins specifically for SCADA protocols. What made this work unique was that the Honeywall image could be placed in front of either the SCADA honeypot or a real non-production PLC. The latter configuration was important because it enabled a physical SCADA device to be used as a honeypot.

In the following year Rrushi and Campbell [61] continued the trend of using real devices when they proposed "reactor mirage theory". Their proposal aimed to use deception to detect intrusions against the nuclear power sector. Their prototype made active decisions to draw adversaries towards a honeypot which used real industrial devices as honeypots. By populating the environment with deceptive systems they increased the possibility of an adversary targeting a non-production system. Similarly, by using real devices as deception systems and creating simulated activity, Modbus protocol traffic, they increased realism and decreased the possibility of an adversary discovering that they are interacting with a honeypot. Despite these benefits, the costs associated with deploying many real devices for deception purposes is high and any increase of network traffic directly or indirectly on a production SCADA network should be approached with close scrutiny.

In another academic proposal in 2009, Valli [75] described a SCADA forensics framework which combined the Snort IDS with two low interaction honeypots; Honeyd and Nepenthes. The idea was to replay known SCADA exploits in a controlled lab to create network IDS rules which would then influence configurations for the two honeypots. However, it is unclear if this initial proposal received further attention.

Dacier et al [17] considered attribution in their study of low-interaction honeypots to differ from that of traceback, i.e. "determining the identity or location of an attacker or an attacker's intermediary". Instead they approached the issue in terms of defining a series of "attack events"that were observed to model the attackers'modus operandi. Attack events comprised a series micro attack events that occurred during observed periods of time, which were then analysed to attempt to establish connections between them in order to form an aggregate of activity into a "Misbehaving Cloud"(MC). The paper illustrated various means to correlate the observed activities into such MCs at an attack level, thus demonstrating the attack techniques, but did not apply this to a wider analysis of the data advertised on the honeypots and the correlation between the attack method and the ultimate aim of the attacker. Pouget and Dacier [58] attempted to address this in a later paper that used clustering algorithms to analyse the data captured by a honeypot and presented methods to identify the root causes of attacks, stating that "identifying the root causes is a prerequisite for a better understanding of malicious activity". The results however, did not propose a framework in which to attempt to assess the attacker's intent.

Spitzner [71] highlighted the limitations of honeypots due to the narrow field of view available to them, and that it only allows a focus on attacks against specific targets (i.e. the honeypot). He highlighted that while the data capture can be very rich, it does not encompass all of the surrounding behaviour that may occur outside of the honeypot that may indicate the wider events associated with an attack.

In order to try and address this issue, Wagener et al [78] adopted high-interaction, self-adapting honeypots that introduced simulated failures into the interactions to repeatedly attract attackers and lure them into revealing as much information about themselves as

possible. The study noted that attackers showed a level of determination to achieve their perceived objectives and that "we assume that attackers are rational and follow a specific goal during attacks", implying a further focus on modus operandi.

The SCADA honeypot proposals discussed so far focused on the assumption that attacks are network borne. In 2012 students at Bonn University in Germany, led by Sebastian Poeplau, created Ghost USB, a honeypot which emulates USB devices to counter the threat of malware that propagated by removable media [56]. Currently, Ghost USB supports 32bit Windows XP/7 and is supported by the Honeynet Project. The project has obvious uses for SCADA systems; Stuxnet was presented as a use case for Ghost USB, since the malware propagated by USB devices connected to SCADA engineer workstations. This tool could be deployed on production systems with little cost and is readily available.

Wilhoit [80] attempted to address the attribution of attacks specifically against ICS by employing a set of honeypots that advertised themselves as an operational system with PLCs attached. The honeypot architecture ran the BeEF framework (Browser Exploitation Framework Project) [3] to embed a script into web pages that was executed every time an attacker compromised the site authentication. The script determined the geographical location of the attacker, as well as capturing statistical information. The study managed to identify the locations of the attackers, and made a high-level assessment of the perceived intent of the attacks, stating that if "an attack was targeted in nature, for instance, but did not compromise the operation of a target ICS device, the attackers' motivation could be espionage or information gathering. If an attack, however, compromised the operation of a target ICS device, depending on how badly it was affected then the motivation could be considered destructive in nature". There would be merit in considering this research in the context of the Miller and Rowe [47] taxonomy.

Most recently the Honeynet Project has announced and released Conpot, which aims to simplify the process of setting up SCADA honeypots [59]. Conpot currently supports Modbus and SNMP, however the developers intend to add support for other protocols. The tool simulates the Siemens S7-200 PLC. Conpot feeds into HPFeeds, a data sharing platform which is used by a number of the Honeynet Project honeypots.

Performance: Honeypots can be deployed so that they do not introduce additional links in a process chain, and if properly located, can avoid impact on existing operations.

Reliability: Honeypots can be deployed away from critical systems and networks to avoid impacting operational processes.

Extent: Honeypots could simulate serial or bus connected devices, or perhaps have some physically attached, but they would not necessarily provide a representative architecture in a complex environment.

Coherence: Honeypots only provide a view of attack behaviour from the device itself, and by its nature encourages the attacker to that device. Wider situational awareness of other malicious activities is not maintained.

Identification: As the attacker is encouraged towards particular devices, repeat activities can be identified and recorded, and mechanisms deployed to increase the level of confidence of attribution.

Intent: The purpose of the attack on they honeypot device will become apparent, however the legal issues of entrapment are yet to be addressed in this matter.

Digital Forensics Techniques. Digital Forensics is a broad subject which involves the recovery, acquisition and investigation of digital evidence. In traditional IT domains commercial tools such as EnCase [69] and FTK [1] and open source tools such as Sleuthkit and Autopsy [67] are used to acquire, analyse, and report on digital evidence. These tools tend to be specific to x86 and x64 processor architectures and targeted towards file systems, such as FAT, NTFS and popular operating systems, such as Windows and Linux.

Forensics in a SCADA environment could identify attribution data to identify perpetrators. In the SCADA network segment and field device segments there are a broad range of devices which may store a wealth of digital evidence. However, SCADA systems come with a unique set of challenges for forensic analysis. For example, the standard forensic procedure for taking a bit-for-bit disk acquisition involves switching off a system, connecting the hard disk to a write blocker and acquisition system and then waiting for the acquisition to complete. Switching off a SCADA system which monitors and controls critical infrastructure is unlikely to be an option. One way to mitigate this issue is to have fail over systems. However, this is costly and if the fail-over system is a duplicate of the original system, it might be infected in exactly the same way.

The diversity of devices that a forensic investigator can encounter in the SCADA environment is far wider than that of the traditional IT domain. Traditional IT systems have a lifespan of a couple of years, perhaps 10 at most, whilst ICS will typically remain in service for 20 years [49] [4]. However, as PLCs and other SCADA devices continue to move towards commercial off-the-shelf hardware and software, the forensic analysis of SCADA systems becomes standardised and therefore simpler.

Among the diverse devices found in SCADA environments is the Historian, which is essentially a database management system (DBMS). It collects a wealth of data to enable auditing, trend analysis and anomaly detection. As a DBMS, traditional database forensics techniques should be suitable for these devices. However, unlike the historian, many of these devices encountered are unlikely to have persistent memory. It is true that "most process control systems were not built to track their processes, but merely to control them"[50]. For example, the Siemens S7-300 PLC uses a micro memory card (MMC) for storage [66] which ranges from 64KB to 8MB, while integrated CPU memory for this device ranges from 32KB to 2MB.

With this absence of persistent memory, researchers have proposed the use of another technique from the traditional IT domain to SCADA; live forensics. In live forensics data acquisition takes place while the system is operational. In traditional IT systems, tools are used to capture running processes, RAM memory, browsing history and more, in the order of volatility. Performing live forensics on an operational machine in a SCADA environment prompts significant challenges; accidentally causing the machine to crash could be catastrophic. Ahmed et al [2] discusses this issue and suggests using fail over systems to allow for live forensic analysis to take place. Another challenge is that post-incident the investigator is competing with recovery efforts which will most likely destroy evidence. There is also clearly a logistics concern when performing SCADA forensics. Field devices could be located many miles away, perhaps on different continents, or perhaps in difficult to reach places, such as on the ocean floor. Physically reaching these devices may not be possible.

Forensics is primarily a practitioner-led field with research taking place as and when it is required. In a recent effort to outline a research agenda for this field, SCADA forensics was identified as a predominant theme [50]. The following points were identified as near future research for forensics in SCADA systems:

1. Collection of evidence in the absence of persistent memory.
2. Hardware-based capture devices for control systems network audit trails.
3. Honeypots for control systems as part of the investigatory process.
4. Radio frequency forensics.
5. Intrusion detection systems for control systems.

Digital Forensics techniques are considered for applicability against our criteria as below:

Performance: Forensic techniques require mechanisms to record behaviour which may degrade performance.

Reliability: The requirement for additional monitoring will have a performance impact on devices not scaled to support facilities beyond their original scope. Additionally, the process may also require compromised devices to be taken out of service in order to facilitate an analysis.

Extent: The lack of support for proprietary devices and operating systems limits the scope of its applicability.

Coherence: If all elements of an ICS could be deployed with forensic tools, the possibility for an end-to-end analysis of attack behaviour increases.

Identification: The ability to identify the attacker from behaviours or technical signatures would depend on the capabilities of the overall suite of forensic tools deployed, their ability to integrate to a coherent time source and operate in proprietary environments. This currently limits the scope of its deployment.

Intent: Forensic tools would support the identification of targeted devices, and may add weight to other evidential means, but by themselves they do not provide attribution suitable for taking forward a prosecution.

Network Forensics. Another field of forensics used in traditional IT systems is network forensics. This field primarily involves two stages: collecting network messages and analysing network messages. Existing infrastructure such as switches and routers can be configured to collect messages, or extra equipment can be deployed, such as a network tap device. By logging messages to files, analysis can take place during an attack or post-attack. During analysis of network traffic, attribution data can be found, such as connection source, time of connection, commands that were sent and payload data.

Collection of data is relatively straightforward. An organisation must identify points in the network where they wish to collect network data. Mahmood et al [45] describes traditional network analysis problems and network sniffer deployment in a SCADA environment. An area that will require further consideration is when traditional communication channels other than Ethernet are used, such as RS232 and radio link. Specialist sniffers will be required in this instance. Traffic should be stored in known network capture formats, such as PCAP. Wireshark, a popular network sniffer and packet analyser tool, already has dissectors for some SCADA protocols, including Modbus [34],

DNP3 [16] and FINS [55], a proprietary protocol, however there are many SCADA protocols that are not supported.

Full packet capture in a traditional IT system can cause problems due to the high volume and large packet size. In a SCADA environment traffic volume is generally much lower and message sizes are much smaller. Message content is likely to be significantly less diverse, as content is machine generated and not user generated. This results in network collection devices requiring less storage and processing power, meaning that organisations can make savings or deploy more devices.

Of course, similar to traceback, network forensics will only be able to identify attacks that use network communications as a vehicle for attack. Those that use removable media will not be visible.

Network Forensics techniques are assessed against our criteria as follows:

Performance: Many ICS protocols are optimised for performance. The introduction of devices into the communications chain will require significant testing to prove it will not increase latency.

Reliability: Significant testing will be required to ensure that the introduction of devices will not impact the boundary conditions of the system.

Extent: The lack of support for proprietary protocols and non-IP bearers limits the scope of its applicability.

Coherence: Network forensics, offer the ability to capture wider attack behaviour as it extends beyond individual devices. However, the lack of support for non-IP protocols and bearers limits its utility.

Identification: Behaviours across the network would allow for attack tools and techniques to be analysed for commonalities.

Intent: Network forensics would, if deployed across all elements of the ICS, provide a means to ascertain the intent of the attack.

Malware Analysis. Malware, in its various forms; virus, worm, trojan, adware, spyware, back doors and rootkits, may be analysed to identify characteristics which could be used as an attribution data source. Malware analysis in the traditional IT domain can be split into two areas: behavioural analysis and code analysis.

Behavioural analysis examines the way that malware interacts with the environment. Malware might make changes to the registry, create new processes, hide files, execute other binaries, contact command-and-control servers, cover tracks by deleting evidence of its modifications (as Stuxnet did), disable security protections, record user interaction (e.g. keylogging), harvest sensitive data, exfiltrate data, attempt to update, pivot to other systems, establish back doors and more. A controlled sandbox environment is usually created to examine this behaviour. Virtual machines are commonly used for this task as they can be quickly reset with snapshot/roll-back functionality. A wide range of tools are available to analyse malware behaviour in the traditional IT domain, such as the Microsoft Windows SysInternals suite [46]. The investigator can change the sandbox environment to illicit a response from malware. Examples of change include:

1. Introducing new services, files and removable media.
2. Introducing Internet connectivity.
3. Browsing websites, sending and receiving e-mail.
4. Inputting passwords and other sensitive information.

The response, or lack of response, helps to identify what the malware does. The process of behavioural analysis can be automated with tools such as CWSandbox [62] which monitors Windows system calls made by malware. Behaviour analysis tools and environments are fairly limited to operating systems used in traditional IT environments e.g. Windows and Linux; they do not support the firmware found on SCADA PLCs and RTUs. Ahmed et al. (2012) [2] identified that SCADA simulation environments should be created, possibly by Universities and industry partners, and this would certainly help to rectify this issue.

Code analysis is concerned with examining the code that makes up the malware. Source code for malware might be available, although it is unlikely. If by chance it is then source code analysis can take place. Otherwise, reverse engineering and debugging take place. Reverse engineering involves restoring the malware's binary machine code to human-readable assembly code, using tools such as IDA Pro [31] and OllyDbg [54]. These tools are particularly effective at reversing binaries compiled for x86, x64 and ARM CPU architectures. The code can then be executed in a debugger to step through the instructions, inspect register contents, identify embedded strings and set breakpoints to determine the malware's functionality.

Practitioners used reverse engineering against the Stuxnet malware [21]. They identified clues in the code, such as binary compile times, suspicious variable names, registry keys that appear to be dates and directory names that might be biblical names. Some or all of these clues could have been false flags; data that was purposely crafted to implicate another entity as the malware authors. Symantec consulted the expertise of established SCADA practitioners in order

to understand the effects that the Stuxnet malware had on the Siemens PLCs. This again highlights the diverse skill sets required for the SCADA environment and the necessity for security professionals to work closely with SCADA engineers. Code analysis was also used to identify re-use of code and libraries; Stuxnet, Flame and Duqu were identified as having shared code.

Malware Analysis techniques are reviewed against our standard criteria below:

Performance: As malware analysis occurs after an infection has been discovered, and takes place in an environment away from operational systems, the process has no impact on ICS performance.

Reliability: Similarly, the offline analysis has no impact on safety processes.

Extent: The propagation of the malware can be determined if it leaves a persistent footprint, although it does not necessarily provide evidence of a targeted progression through systems and devices.

Coherence: Malware analysis provides limited opportunities to cross-reference traffic and behaviours.

Identification: The reverse-engineering of malware may highlight commonalities in coding techniques, naming conventions and other identifiable features.

Intent: The functionality of the malware can be assessed as to its purpose, and from that its likely targets.

Intelligence-led Attribution. A number of non-technical investigatory techniques may offer alternative or complementary approaches to assigning attribution to a cyber attack. For the purposes of this survey these have been categorised as âĂŸintelligence-ledâĂŹ techniques.

As technical attribution techniques offer limited and varying degrees of actionable data. Carr [11] proposed that the "one thing you can count on is that someone has to pay for the necessities of virtual combat. Therefore, one sound strategy in any cyber investigation is to follow the money trail created by the necessary logistics of organizing a cyber attack – domain registration, hosting services, acquisition of software, bandwidth, and so on."He highlighted that although false identities are often used when registering and acquiring services, the increased use of social media and the increasing size of individual and corporate digital footprints allows for a forensic examination of online presence and identity may reveal such deceptions. Gantz et al [25] estimated that approximately 45GB of data existed for every person on the planet. They also discussed the analysis of âĂŸdigital shadowsâĂŹ, that ambient content data created by traffic cameras, use of ATMs, online transactions etc.

An analysis of alleged Chinese computer attack behaviour [13] resulting from a reported seven years of covert observation offered an insight into the scale and complexity of attacks on ICS. Targets included transportation, navigation, engineering, food and agriculture, chemicals, energy, aerospace and mining - all areas where industrial control systems were likely to be used. Its attribution of the observed attack behaviour to China was based upon a mix of technical measures and intelligence data gathering and analysis. In particular, the report focused on commonalities between attack methods, consistencies in naming conventions and comparative analysis of malware.

Both Fireeye [22] and Shivraj [65] described the consistency in attack behaviour observed from common sources. Fireye leveraged their position as a supplier of commercial security products to gather and analyse APT callback traffic and events in order to establish patterns of behaviour and command and control traffic. Shivraj [65] defined the stages of contemporary APT behaviour with a focus on SCADA attacks and illustrated how common malware approaches can be applied to ICS targets with limited alteration required, at least at the early stages of an attack. The combined findings of both papers could be potentially combined to provide an indication of attack attribution and a tangible assessment of where the target is in the attack cycle, and therefore what preventative measures may be appropriate as a consequence.

Langner [43], in his investigation of the Stuxnet malware, was unable to provide any substantive evidence to attribute the originator of the code, but did find significant indicators as to the evolution of the software and its intended effects. In particular, he highlighted the level of industrial process and control system knowledge required to develop the malware, and speculated as to the high level of testing that would have been required to prove the payload prior to its release. The necessity for the malware to traverse the traditional IT layers of the target environment before its compromise of the industrial control system to damage the physical elements of the system under control gave rise to a complex piece of software. Langner [43] believed that to develop Stuxnet required nation-state resources. Although he provided no irrefutable evidence for this, he presented a compelling argument based on the complexity of the development undertaken. Knake [38], took a more pragmatic and empirical view when testifying to the US House of Representatives on the cyber threat, stating that at the uppermost level of threat, that of a nation-state, the issue of attribution is simplified as "there are a limited number of actors capable of carrying

out such attacks."It is perhaps worth considering the requisite capability of an actor when attempting to assign attribution to a covert attack.

This concept of nation-state capability was extended by Geers et al [28] in an attempt to characterise the motivations and nature of state-sponsored cyber attacks. In a discussion of cyberwarfare, the paper proposed that "[a] cyber attack is best understood not as an end in itself, but as a potentially powerful means to a wide variety of political, military, and economic goals."In this context, an analysis of the intent of an attack would perhaps elucidate which nation-state(s) would benefit from the outcome of the attack, and from this we could derive motive. While not an attribution method in itself, it would allow for an investigation into attribution to be focused on likely perpetrators.

In a post-Stuxnet analysis, Bencsáth et al [8] undertook comparative analyses of malware in their investigation of the Duqu, Flame and Gauss executables. The report highlighted that Duqu shared "striking similarities with Stuxnet"and proposed that there were indications that the three malware tools were part of the same family, suggesting at least a partial common source.

Accepting that there are inherent problems with absolute attribution of cyber attacks, Kalutarage et al [35] proposed a probabilistic approach based on Bayesian methods. The methodology divided the problem into two smaller domains; evidence fusion and aggregation (described as "accumulation"), and the subsequent analysis (described as the "anomaly definition"). The accumulation allowed for the incorporation and use of many Bayesian approaches and prepared the anomaly definition to allow the analysis of attacker activity patterns within a series of node profiles. The data used in the experimentation came from a series of logging techniques and appeared to be entirely IP-based. However, there appeared to be nothing in the methodology that would preclude the use of serial data or historian records from an ICS. In the context of intelligence-led attribution analyses, there may be some valuable research to be undertaken in the field of probabilistic attribution.

Performance: As intelligence-led analysis requires no specific hardware or software to be deployed into the ICS, it has no impact.

Reliability: As above, no changes to the operational systems are required.

Extent: Without technical means, the level of penetration of an attack cannot be determined.

Coherence: The process does not determine how the attack was achieved.

Identification: An analysis of tools, techniques and methods of known malicious actors can be used to determine a subset of possible attack originators.

Intent: A broad analysis of the attack can allow non-technical impacts of the attack to be considered, including financial losses, reputational impact etc., and an assessment of who would gain as a result of the attack.

6. Summary of Attribution Techniques

Table 2 summarises the review of the attribution techniques by assigning a value of low, medium or high that refers to the techniques ability to support the chosen assessment criteria. A value of 1, 2 or 3 is assigned respectively, allowing an overall assessment to be produced (out of a possible total of 18).

7. Conclusions

This study has identified few publications on the subject of the attribution of attacks on industrial control systems, and none where the problem has been explored to any significant depth. Technical research on the related subject of IP traceback has highlighted that while the research areas are maturing, the techniques do not address the multi-stage nature of contemporary cyber attacks and only serve to identify the device from which the attack was launched. Honeypots offer a potentially richer dataset from which to analyse the source of an attack, and begin to look for repeated patterns of behaviour, but relies on an organisation being prepared to leave devices open to exploitation by malicious actors in order to obtain this information. Few of the publications reviewed provided a detailed analysis of the nature of industrial protocols, particularly those not based on IP, and the need to integrate logging and monitoring data into any attribution mechanism in order to assess the entire attack chain.

The international legal frameworks for dealing with cyber attacks appear fragmented and do not lend themselves to addressing transnational malicious activities. In order to prosecute for such behaviours it is necessary to identify the human, or group of humans, responsible for the attack. Technical attribution cannot achieve this. In order to prosecute, or in the extreme case of cyberwarfare, to retaliate, it is necessary to determine mens rea. This intent cannot be defined by technical means alone, nor can it be determined absolutely. Alongside the technical means that can be applied during or after an attack, there are a number of intelligence-led investigatory methods and techniques than can be adopted to determine the motivations and capability of an attacker, along with their previous

Table 2. Summary of Attribution Technique Assessment.

	Traceback	Honeypots	Digital Forensics	Network Forensics	Malware Analysis	Intelligence Led
Performance	Low	High	Low	Low	High	High
Reliability	Low	High	Low	Low	High	High
Extent	Low	Medium	Low	Medium	Medium	Low
Coherence	Low	Medium	Medium	Medium	Low	Low
Identification	Low	Medium	Medium	High	Medium	High
Intent	Low	Low	Low	High	High	High
TOTAL	**6**	**13**	**8**	**12**	**14**	**14**

modus operandi, in order to present a probabilistic picture of the originator of the attack.

8. Future Research Opportunities

This study suggests that further research into the end-to-end chain of attacks on industrial control systems, covering all elements of their architecture, is required to allow comprehensive attack taxonomies to be defined and applied. This study also suggests there is merit in research into a methodology that encompasses both technical and non-technical techniques to form a probabilistic model of attribution.

References

[1] AccessData. Forensic toolkit (ftk). http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk, Accessed 01/01/2016.

[2] Irfan Ahmed, Sebastian Obermeier, Martin Naedele, and Golden G Richard III. Scada systems: Challenges for forensic investigators. *Computer*, (12):44–51, 2012.

[3] W Alcorn. Browser exploitation framework (beef). *Online at http://beefproject. com*, 2013.

[4] Rafael Ramos Regis Barbosa. *Anomaly detection in SCADA systems: a network based approach*. University of Twente, 2014.

[5] S Barnum. Common attack pattern enumeration and classification (capec) schema description. *Cigital Inc, http://capec. mitre. org/documents/documentation/CAPEC_Schema_Descr iption_v1*, 3, 2008.

[6] Andrey Belenky and Nirwan Ansari. Ip traceback with deterministic packet marking. *IEEE communications letters*, 7(4):162–164, 2003.

[7] Steven Michael Bellovin, Marcus Leech, and Tom Taylor. Icmp traceback messages. 2003.

[8] Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, and Mark Felegyhazi. The cousins of stuxnet: Duqu, flame, and gauss. *Future Internet*, 4(4):971–1003, 2012.

[9] Benjamin A Blakely. Cyberprints: Identifying cyber attackers by feature analysis. 2012.

[10] Susan W Brenner. "At Light Speed": Attribution and response to cybercrime/terrorism/warfare. *The Journal of Criminal Law and Criminology*, pages 379–475, 2007.

[11] Jeffrey Carr. *Inside cyber warfare: Mapping the cyber underworld*. " O'Reilly Media, Inc.", 2011.

[12] Nicholas B Carr. *Development of a tailored methodology and forensic toolkit for industrial control systems incident response*. PhD thesis, Monterey, California: Naval Postgraduate School, 2014.

[13] Mandiant Intelligence Center. Apt1: Exposing one of chinaâĂŹs cyber espionage units. *Mandian. com*, 2013.

[14] David D Clark and Susan Landau. The problem isn't attribution: it's multi-stage attacks. In *Proceedings of the Re-architecting the Internet Workshop*, page 11. ACM, 2010.

[15] Paul Cornish, David Livingstone, Dave Clemente, and Claire Yorke. *Cyber security and the UK's critical national infrastructure*. Chatham House, 2011.

[16] Ken Curtis. A dnp3 protocol primer. *DNP User Group*, pages 1–8, 2005.

[17] Marc Dacier, Van-Hau Pham, and Olivier Thonnard. The wombat attack attribution method: some results. In *Information Systems Security*, pages 19–37. Springer, 2009.

[18] Paul Didier, Fernando Macias, James Harstad, Rick Antholine, Scott A Johnston, Sabina Piyevsky, Mark Schillace, Gregory Wilcox, Dan Zaniewski, and S Zuponcic. Converged plantwide ethernet (cpwe) design and implementation guide. *CISCO Systems and Rockwell Automation*, pages 252–253, 2011.

[19] David Patrick Duggan. *Generic threat profiles*. United States. Department of Energy, 2005.

[20] K. Erickson. Programmable logic controllers: Hardware, software architecture. https://www.isa.org/standards-publications/isa-publications/intech-magazine/2010/december/automation-basics-programmable-logic-controllers-hardware-software-architecture, 2010.

[21] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5, 2011.

[22] FireEye. The advanced cyber attack landscape. https://www2.fireeye.com/WEB2013ATLReport.html, 2013.

[23] Terry Fleury, Himanshu Khurana, and Von Welch. Towards a taxonomy of attacks against energy control systems. In *Critical Infrastructure Protection*, pages 71–85. Springer, 2008.

[24] Brendan Galloway and Gerhard P Hancke. Introduction to industrial control networks. *Communications Surveys & Tutorials, IEEE*, 15(2):860–880, 2013.

[25] John F Gantz and Christopher Chute. The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011. IDC, 2008.
[26] Zhiqiang Gao and Nirwan Ansari. Tracing cyber attacks from the practical perspective. *Communications Magazine, IEEE*, 43(5):123–131, 2005.
[27] Kenneth Geers. The challenge of cyber attack deterrence. *Computer Law & Security Review*, 26(3):298–303, 2010.
[28] Kenneth Geers, Darien Kindlund, Ned Moran, and Rob Rachwald. World war c: Understanding nation-state motives behind todayâĂŹs advanced cyber attacks. Technical report, Technical report, FireEye, 2014.
[29] Michael T Goodrich. Efficient packet marking for large-scale ip traceback. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 117–126. ACM, 2002.
[30] Ihab Hamadeh and George Kesidis. A taxonomy of internet traceback. *International Journal of Security and Networks*, 1(1-2):54–61, 2006.
[31] Hex-Rays. What is ida all about? https://www.hex-rays.com/products/ida/, Accessed 01/01/2016.
[32] Jeffrey Hunker, Bob Hutchinson, and Jonathan Margulies. Role and challenges for sufficient cyber-attack attribution. *Institute for Information Infrastructure Protection*, 2008.
[33] ICS-CERT. Ics-cert year in review - 2012. Online, 2012.
[34] National Instruments Inc. The modbus protocol in-depth. http://www.ni.com/white-paper/52134/en/, Accessed 13/12/2014.
[35] Harsha K Kalutarage, Siraj Shaikh, Qin Zhou, Anne E James, et al. Sensing for suspicion at scale: A bayesian approach for cyber conflict attribution and reasoning. In *Cyber conflict (CYCON), 2012 4th international conference on*, pages 1–19. IEEE, 2012.
[36] Mike Keyser. Council of europe convention on cybercrime, the. *J. Transnat'l L. & Pol'y*, 12:287, 2002.
[37] Eunjong Kim, Dan Massey, and Indrajit Ray. Global internet routing forensics. In *Advances in Digital Forensics*, pages 165–176. Springer, 2005.
[38] Robert K Knake. Untangling attribution: Moving to accountability in cyberspace. *Prepared Statement before the Subcommittee on Technology and Innovation, Committee on Science and Technology, Hearing: Planning for the Future of Cyber Attack*, 2010.
[39] Eric D Knapp and Joel Thomas Langill. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.
[40] Uta Kohl. Eggs, jurisdiction, and the internet. *International and comparative law quarterly*, 51(03):556–582, 2002.
[41] Turgay Korkmaz, Chao Gong, Kamil Sarac, and Sandra G Dykes. Single packet ip traceback in as-level partial deployment scenario. *International Journal of Security and Networks*, 2(1-2):95–108, 2007.
[42] Vadim Kuznetsov, Helena Sandstrom, and Andrei Simkin. An evaluation of different ip traceback approaches. In *Information and Communications Security*, pages 37–48. Springer, 2002.
[43] Ralph Langner. To kill a centrifuge. *Langner Group*, 2013.
[44] Martin C Libicki. *Cyberdeterrence and cyberwar*. Rand Corporation, 2009.
[45] Abdun Naser Mahmood, Christopher Leckie, Jiankun Hu, Zahir Tari, and Mohammed Atiquzzaman. Network traffic analysis and scada security. In *Handbook of Information and Communication Security*, pages 383–405. Springer, 2010.
[46] Microsoft. Windows sysinternals. https://technet.microsoft.com/en-gb/sysinternals/bb545021.aspx, Accessed 01/01/2016.
[47] Bill Miller and Dale Rowe. A survey scada of and critical infrastructure incidents. In *Proceedings of the 1st Annual conference on Research in information technology*, pages 51–56. ACM, 2012.
[48] John Moteff and Paul Parfomak. Critical infrastructure and key assets: definition and identification. DTIC Document, 2004.
[49] Martin Naedele. Addressing it security for critical control systems. In *40th Annual Hawaii International Conference on System Sciences*. IEEE, 2007.
[50] Kara Nance, Brian Hay, and Martin Bishop. Digital forensics: defining a research agenda. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–6. IEEE, 2009.
[51] Andrew Nicholson, Helge Janicke, and Tim Watson. An initial investigation into attribution in scada systems. In *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013*, pages 56–65. BCS, 2013.
[52] Andrew Nicholson, Tim Watson, Peter Norris, Alistair Duffy, and Roy Isbell. A taxonomy of technical attribution techniques for cyber attacks. In *European Conference on Information Warfare and Security*, page 188. Academic Conferences International Limited, 2012.
[53] Cabinet Office. The uk cyber security strategy–protecting and promoting the uk in a digital world, 2011.
[54] OllyDbg.de. Ollydbg. http://www.ollydbg.de/, Accessed 01/01/2016.
[55] Omron. Fins command technical guide. http://downloads.omron.us/, 2012.
[56] Sebastian Poeplau and Jan Gassen. A honeypot for arbitrary malware on usb storage devices. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*, pages 1–8. IEEE, 2012.
[57] Venkat Pothamsetty and Matthew Franz. Scada honeynet project: Building honeypots for industrial networks, 2008.
[58] Fabien Pouget, Marc Dacier, et al. Honeypot-based forensics. In *AusCERT Asia Pacific Information Technology Security Conference*, 2004.
[59] The Honeynet Project. Conpot. https://www.honeynet.org/node/1047, November 2013.
[60] F. Rios-Gutierrez. Overview of programmable logic controllers. http://www.d.umn.edu/ snorr/ece4951s7/Lect4.pdf,, 2007.
[61] Julian Rrushi and Roy Campbell. Detecting cyber attacks on nuclear power plants. In *Critical Infrastructure Protection II*, pages 41–54. Springer, 2008.

[62] sandbox.org. Understanding the sandbox concept of malware identification. www.cwsandbox.org, Accessed 01/01/2016.

[63] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for ip traceback. In *ACM SIGCOMM Computer Communication Review*, volume 30, pages 295–306. ACM, 2000.

[64] Michael N Schmitt. *Tallinn manual on the international law applicable to cyber warfare*. Cambridge University Press, 2013.

[65] Anant Shivraj. Cyber threat evolution with a focus on scada attacks. Online, May 2011.

[66] Siemens. Which memory cards can you use with an s7-300 cpu? https://support.industry.siemens.com/cs/document/19102565/which-memory-cards-can-you-use-with-an-s7-300-cpu?dti=0&lc=en-WW, July 2013.

[67] sleuthkit.org. Open source digital forensics. http://www.sleuthkit.org/, Accessed 01/01/2016.

[68] Alex C Snoeren, Craig Partridge, Luis A Sanchez, Christine E Jones, Fabrice Tchakountio, Stephen T Kent, and W Timothy Strayer. Hash-based ip traceback. In *ACM SIGCOMM Computer Communication Review*, volume 31, pages 3–14. ACM, 2001.

[69] Guidance Software. Encase forensic v7 overview. https://www2.guidancesoftware.com/products/Pages/encase-forensic/overview.aspx, Accessed 01/01/2016.

[70] Dawn Xiaodong Song and Adrian Perrig. Advanced and authenticated marking schemes for ip traceback. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 878–886. IEEE, 2001.

[71] Lance Spitzner. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading, 2003.

[72] Keith Stouffer, Joe Falco, and Karen Scarfone. Guide to industrial control systems (ics) security. *NIST special publication*, pages 800–82, 2011.

[73] W Timothy Strayer, Christine E Jones, Isidro Castineyra, Joel B Levin, and Regina Rosales Hain. An integrated architecture for attack attribution. *BBN Technologies*, 10, 2003.

[74] Vrizlynn LL Thing, Morris Sloman, and Naranker Dulay. Adaptive response system for distributed denial-of-service attacks. In *Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on*, pages 809–814. IEEE, 2009.

[75] Craig Valli. Scada forensics with snort ids. 2009.

[76] Shweta Vincent and J Immanuel John Raja. A survey of ip traceback mechanisms to overcome denial-of-service attacks. In *Proceedings of the 12th international conference on Networking, VLSI and signal processing*, pages 20–22, 2010.

[77] Susan Marie Wade. Scada honeynets: The attractiveness of honeypots as critical infrastructure security tools for the detection and analysis of advanced threats. 2011.

[78] Gérard Wagener, Alexandre Dulaunoy, Thomas Engel, et al. Self adaptive high interaction honeypots driven by game theory. In *Stabilization, Safety, and Security of Distributed Systems*, pages 741–755. Springer, 2009.

[79] David A Wheeler and Gregory N Larsen. Techniques for cyber attack attribution. Technical report, DTIC Document, 2003.

[80] Kyle Wilhoit. The scada that didnâĂŹt cry wolf. *Trend Micro Inc., White Paper*, 2013.

[81] Theodore J Williams. The purdue enterprise reference architecture. *Computers in industry*, 24(2):141–158, 1994.

[82] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. A taxonomy of cyber attacks on scada systems. In *Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing*, pages 380–388. IEEE, 2011.

[83] Bonnie X Zhu. Resilient control and intrusion detection for scada systems. Technical report, DTIC Document, 2014.

[84] Quanyan Zhu, Craig Rieger, and Tamer Bacsar. A hierarchical security architecture for cyber-physical systems. In *Resilient Control Systems (ISRCS), 2011 4th International Symposium on*, pages 15–20. IEEE, 2011.