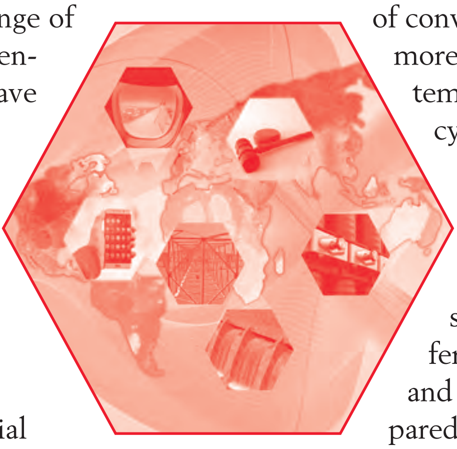
Forensic Attacks Analysis and the Cyber Security of Safety-Critical Industrial Control Systems

by Chris W. Johnson, Rob Harkness and Maria Evangelopoulou
East Kilbride and Glasgow, U.K.



Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) applications monitor and control a wide range of safety-related functions. These include energy generation, where failures could have significant, irreversible consequences. They also include the control systems that are used in the manufacture of safety-related products. In this case, "bugs" in an ICS/SCADA system could introduce flaws in the production of components; these flaws remain undetected before being incorporated into safety-related applications. Industrial Control Systems, typically, use devices and networks that are different from conventional IP-based infrastructures. These differences prevent the re-use of existing cyber-security products in ICS/SCADA environments; the architectures, file formats and process structures are all different. This paper supports the forensic analysis of industrial control systems in safety-related applications. In particular, we describe how forensic attack analysis is used to identify weaknesses in devices so that we can both protect components and determine the information that must be analyzed during the aftermath of a cyber-incident. Simulated attacks detect vulnerabilities; a risk-based approach can then be used to assess the likelihood and impact of any breach. These risk assessments are then used to justify both immediate and longer-term countermeasures.

Introduction

SCADA (Supervisory Control and Data Acquisition) and Industrial Control Systems (ICS) support a wide range of safety-related applications. They typically integrate Programmable Logic Controllers (PLCs) and a range of sensors, which are different from the devices used in more conventional networks. They use different file formats, process structures, protocols and I/O interfaces. Without some form of emulation, the run-time and programming environments are mutually incompatible between ICS/SCADA environments and those that run under Windows or Linux. In the future, these differences will diminish as SCADA/ICS migrate from specialist protocols, such as Modbus and Profibus, to process-control variants of TCP/IP [Ref. 1]. The lack of conventional network protocols and of the more common "office based" operating systems has important consequences for the cyber-security of SCADA/ICS systems [Ref. 2]. Existing tools for computer forensics cannot easily be extended to these industrial applications. They use different file formats and process structures, and there are significant differences in the run-time environments and SCADA/ICS I/O mechanisms compared to office-based systems using Windows or Linux. Ahmed *et al.* [Ref. 3] illustrate this by identifying layers in SCADA/ICS forensics. Conventional techniques work on Layers 2 and 3, but not on Layers 0 and 1:

- Layer 0 is the lowest layer and can be thought of as the individual field devices connected via a bus network.
- Layer 1 controllers receive input signals from field devices and other controllers sending output to other field devices — smart controllers sit at Layers 0 and 1.
- Layer 2 implements a supervisory network, typically using a LAN connecting the lower layers multiplexing data through a Human Machine Interface (HMI).
- Layer 3 gathers domain controllers and application servers, often isolated through a demilitarized zone (DMZ) between the enterprise LAN and the lower operational layers.

Computer forensics is the use of investigation and analysis techniques to gather and preserve evidence from a particular device in a way that is suitable for presentation in a court of law or to direct the response to any attack. The goal of computer forensics is to perform a structured investigation while maintaining a documented chain of evidence to find out exactly what happened on a computing device and who was responsible for it [Ref. 4]. Forensic investigators typically follow a standard set of procedures. An initial assessment of the system helps determine whether symptoms are related to a cyber attack rather than, for example, a hardware fault. If the initial assessment suggests a security concern, then the device must usually be isolated to prevent accidental contamination. Investigators then make a digital copy of the device's storage media, and all investigation is done on the digital copy [Ref. 5]. A variety of proprietary software can be used to examine the copy, searching hidden folders and unallocated disk space for copies of deleted, encrypted or damaged files. Any evidence found on the digital copy is carefully documented and verified against the original in preparation for legal proceedings that involve discovery, depositions or litigation. These digital investigation techniques are less mature than other branches of forensic science. For example, it is unclear how to assess the extent to which vulnerabilities contribute to a successful attack. This is important because finite investigatory resources must trace the source of an attack and understand the root causes of previous vulnerabilities. A number of organizations have published general guidance on forensic investigations [Ref. 6, 7, 8]. However, none of these documents explicitly considers forensic capture in safety-related SCADA/ICS applications. For example, it is far easier to isolate office-based systems compared to the flight data processing systems of air traffic management applications. In such contexts, we must first ensure that control has been safely transferred to secondary systems before conducting any forensic examination. Without adequate care, this transfer process may inadvertently overwrite evidence of an attack.

The New Domain of SCADA/ICS Forensics

Many SCADA/ICS components in safety-critical applications are isolated, making it difficult for malware to cross the "air gap." However, it is still possible for attacks to be launched through the laptops and other field devices that are periodically connected to programmable logic controllers (PLCs) and sensors — for diagnostic functions, for example — to install new firmware or to monitor performance. The application of a security patch or device monitoring can, therefore, increase the vulnerability of an air-gapped system, providing a mechanism for the transfer of malware. In air-gapped systems distributed across a large industrial site, there are significant logistic costs in obtaining device logs.

Many legacy ICS applications were developed without any regard for forensics. It may be hard to make any changes to introduce intrusion detection systems when source code is missing or poorly understood with legacy documentation [Ref. 1]. The companies involved in the original development of these systems may no longer be in business and the costs of any modifications may be prohibitive. Legacy systems often lack the processing power/memory/network bandwidth required to support additional logging or live monitoring tasks. These issues combine to create significant concerns that forensic analysis techniques may violate safety requirements — which guided the initial deployment of legacy systems — for instance, by increasing processing overheads or network delays.

In many cyber-attacks, it is difficult for engineers to determine whether the symptoms they identify are associated with a malicious attack, operational noise or more "benign" bugs. It is often necessary to inspect system logs or network monitoring tools to collect further evidence to confirm an initial diagnosis. These logs tend to be focused around the higher levels of any SCADA/ICS system — Layer 3 in the earlier taxonomy. In contrast, the lower levels of a SCADA/ICS network arguably cause the greatest concern. Advanced persistent threats have exploited existing vulnerabilities while these lower levels remain relatively unsupported by existing forensic tools.

A number of problems complicate the forensic analysis of Layers 1 to 3. In particular, SCADA/ICS systems tend to integrate devices from several different manufacturers with different proprietary software. At these lower levels, forensic analysis must overcome an IPR (Intellectual Property Rights) bottleneck. Manufacturers have clear commercial interests in not disclosing detailed technical information about the hardware and software that they embed within their products. Although some suppliers will cooperate with forensic investigations, many are unwilling to provide details of their memory and processing architectures and few sell specialist forensic tools. Other vendors prefer that customers return their devices so the manufacturer can conduct the forensic analysis. This is not always possible if suppliers are based in another legal jurisdiction. In consequence, the co-authors of this paper have been forced to reverse engineer many different SCADA/ICS components. This is a labor-intensive process. To identify potential malware, it is first necessary to reconstruct the memory maps and communications behavior that characterize normal operation for PLCs and sensors. The encryption and obfuscation that helps to protect the cyber-security of SCADA/ICS components can also frustrate the reverse engineering that is a prerequisite for many forensic studies. This does not simply progress on a device-by-device basis; cyber-specialists must also con-

> “Manufacturers have clear commercial interests in not disclosing detailed technical information about the hardware and software that they embed within their products. Although some suppliers will cooperate with forensic investigations, many are unwilling to provide details of their memory and processing architectures and few sell specialist forensic tools.”

sider vulnerabilities that stem from interactions between individual SCADA/ICS components.

Forensic Attack Analysis of SCADA/ICS Systems

Forensic analysis typically proceeds by comparing the known “valid” configuration or behavior of a device with that of a potentially compromised component. Given the IP bottleneck mentioned earlier, it can be hard for the operators of safety-critical SCADA/ICS systems to obtain all the details they might require to conduct a forensic analysis from the device manufacturers. Hence, it is necessary to study these components before an attack takes place. This provides multiple benefits. First, a detailed analysis of SCADA/ICS components can reveal security weaknesses, which can then be addressed — for instance, by replacing key components with more secure counterparts. Second, the analysis can help identify the nominal characteristics that are then used to inform any subsequent forensic analysis. This, however, is a costly and time-consuming exercise.

Several teams have developed similar approaches to the analysis of SCADA/ICS systems [Refs. 1 & 9], based on what we term “forensic attacks.” In other words, the first stage in preparing for any forensic analysis is to identify and exploit weaknesses in the PLC or sensor. The following sections illustrate the results of applying this approach to a number of different SCADA/ICS components. The identity of the devices is redacted for obvious reasons.

Forensic Attacks on Digital Protection Relays:

The first example of a forensic attack focuses on a class of digital relay used for current and voltage protection in electrical distribution systems. A field device/PC can be connected to the device to upload particular configuration parameters.

- **Stage One: Identify Vulnerabilities.** The first stage in forensic attack analysis is to identify vulnerabilities associated with the device. This can be done either through consultation with the manufacturer, through consultation with end-user groups or, as in this case, through ICS pen testing. This particular relay was protected against modification using a password that was set through the field device. However, the passwords were only four digits long, with no restriction on the number of attempts before the device locked them out. Passwords were unencrypted between the relay and the field device. Modbus was also used to communicate with the device; this is known to be vulnerable, for example, to man-in-the-middle attacks [Ref. 10]. These vulnerabilities were exposed using serial port monitoring tools to analyze the packets sent between the field device and the SCADA/ICS relay. This first step in a “forensic attack” helps reveal vulnerabilities that also apply to air-gapped devices.
- **Stage Two: Identify Attack Methods.** The second stage in forensic attack analysis identifies ways in which attackers might exploit a potential vulnerability. In a risk-based approach, some vulnerabilities require specialist technical knowledge before they can be exploited. Other vulnerabilities have limited consequences for the safety of application processes. In our relay, attackers can use the password to reset or disable voltage protection within a safety-critical application. Network monitoring was also used to identify the packet structure that triggered the change of password; hence, even if the password was made more secure, an attacker could spoof the transmission of the update command to the device and trigger a reset. This could be used in an iterated attack to cause denial of service by continually resetting the relay.

- **Stage Three: Implement Immediate Risk Reduction.** It is seldom possible to immediately substitute thousands of legacy devices. As a consequence, our forensic attack analysis helped identify the need for increased physical and cyber-security measures surrounding the field devices that were attached to the relays. It also motivated increased monitoring of the interconnections between SCADA/ICS components and of their configuration logs. Short-term recommendations also included dissemination activities to partner organizations and to national bodies, including CERTs.
- **Stage Four: Implement Long-Term Solutions.** Once high-risk attacks have been identified, it is important to identify long-term solutions. This can involve firmware updates — if manufacturers will provide them. In other cases, it may be necessary to identify alternate devices that provide an appropriate level of security in safety-related systems. In this case, our findings helped inform subsequent procurement decisions, even though we could not immediately replace all of the devices that were deployed with application processes.

Forensic Attacks on Security Cameras

We have also used forensic attack techniques on a range of security cameras. These play a key role in the physical security of many critical installations. They are also increasingly flexible in terms of their operation and use.

- **Stage One: Identify Vulnerabilities.** The ability to configure these devices has led to a number of inadvertent security concerns; in particular, when default settings expose potential vulnerabilities. Forensic attacks often yield different results when applied to devices "as secured" by their manufacturer and "as deployed" in a safety-critical process. One of these cameras offered multiple levels of security, including the use of SSL certification and AES encryption of the audio and video channels with 128-bit keys. However, these could only be activated using an optional site license and additional package.
- **Stage Two: Identify Attack Methods.** As delivered, the camera could be configured over a Telnet connection which, like Modbus, has several well-known vulnerabilities [Ref. 11]. User credentials can be sent unencrypted to the device. The camera transferred images to a local FTP server, again using a potentially vulnerable protocol [Ref. 12]. Both vulnerabilities were confirmed using a network monitoring application. It was possible to implement a man-in-the-middle attack against the Telnet configuration scenario and against the use of FTP image transfers. The device could also be configured to use digest authentication. Applying three hash methods to different combinations of a username, password and a server generated nonce value generates the responses. Nevertheless, digest authentication can still be vulnerable to attack. For example, attackers can auto-generate responses with different hashes and "brute force" the device. As with the protection relay, the camera does not automatically block multiple unsuccessful access requests. Many of these vulnerabilities stem from the flexibility of the camera, which runs a variant of the Linux operating system.
- **Stage Three: Implement Immediate Risk Reduction.** Unlike the relays, the security cameras provided enhanced levels of security. These were activated if the devices were correctly configured. Our simulated attacks helped to refine installation and maintenance guides so that staff were encouraged to ensure that adequate security settings were in place. In terms of forensics, the results motivated the logging of every access to reconfigure the device, and also monitoring the network traffic associated with image hosting.
- **Stage Four: Implement Long-Term Solutions.** Once the immediate measures had been implemented, we focused procurement on devices that were secure "by design" rather than "by configuration." This is harder than might be assumed. Our forensic attack analysis revealed a strong correlation between the sophistication of the protection and the flexibility of the device. In other words, security cameras that offered enhanced forms of encryption and access control tended to also support the highest degree of configuration. Very few of these cameras provided protection by default — again reiterating the need to ensure necessary levels of training for maintenance engineers.

Forensic Attacks on PLCs

Programmable Logic Controllers (PLCs) are the foundation of most ICS/SCADA systems. We were, therefore, concerned to apply our forensic attack techniques to devices from a number of different manufacturers. The following results were synthesized from these devices for reasons of confidentiality. This illustrates another aspect of the approach, given that many of the vulnerabilities and their associated attack methods were common across several of the PLCs that we examined.

- **Stage One: Identify Vulnerabilities.** Some of these suppliers already provide their own security alerts, describing DLL file injection into programming environments, similar to those used for the configuration of the relays mentioned earlier. Other security alerts describe man-in-the-middle and denial of service (DOS) attacks, for instance by forcing the PLC into a degraded mode, terminating its operation. These published vulnerabilities and the associated countermeasures provide a starting point for our simulated "forensic attacks" on PLCs.
- **Stage Two: Identify Attack Methods.** Firmware updates are a potential means of breaching air-gapped systems. Many of these updates are downloaded from the web onto a field device/PC and then uploaded to the PLC over a USB connection. This creates the possibility of a "water hole" attack if end users inadvertently download malicious code through bogus Websites. In some cases, the URL of the server that is polled for new versions of the firmware was stored in a plain-text, editable configuration file associated with the PLC programming and configuration environment. It was possible to substitute an arbitrary address and spoof the transfer of alternate files without generating a warning. As before, monitoring tools can be used to intercept and analyze both the download to the update machine and also the upload process to the PLC. In most cases, the firmware is encrypted or is hard to reconstruct without detailed knowledge of the target architecture. However, it is still possible to envisage attacks that, for example, disable USB communication between the PLC and the configuration platform through altering the USBSTOR registry values. This would be easy to detect, but annoying if extended across an organization. Other vulnerabilities relate to the "ladder logic" that is used to code PLCs. In many cases, these are protected by unencrypted checksums. Our attacks were able to perform code injection. Hash collisions prevented the detection of these changes, even when the code was uploaded to a device.
- **Stage Three: Implement Immediate Risk Reduction.** Immediate countermeasures included strict physical control over access to the PLCs and the configuration machines, as well as auditing the software installed on these devices. Forensic measures included monitoring the download and upload of firmware, as well as guidance on the preservation of volatile and non-volatile memory in the PLCs. The results of our simulated attacks also revealed scenarios where we would need external support from the manufacturer to access features on the SCADA/ICS devices that would not otherwise be exposed to the end users.
- **Stage Four: Implement Long-Term Solutions.** Too often, papers on this topic publicize successful attack methods as if this was the appropriate outcome for cyber-security research. In contrast, we stress that a successful forensic attack is the beginning — and not the end — of an investigation. In this case, procurement decisions were influenced by the results of the study. It also triggered a mutually beneficial dialogue with manufacturers.

> "There are many directions for future work. In the short term, we are developing guidance on data capture from SCADA/ICS applications. The intention is to help operators obtain sufficient information to diagnose the causes of an attack and return the plant to a safe condition/operation as quickly as possible."

Conclusion

This paper has argued that "forensic attacks" helped to identify weaknesses in the SCADA/ICS systems that control a wide range of safety-related applications. Simulated attacks detect vulnerabilities; a risk-based approach can then be used to assess the likelihood and impact of an attack. These risk assessments are used to justify both immediate and longer-term countermeasures that include physical and local access control, the use of encryption and security audits, as well as default configurations, to ensure that protection mechanisms are set correctly when devices are installed or maintained. In other cases, insecure devices must be replaced by more trustworthy components. If these options are not possible, network and device monitoring techniques can help operators identify the causes and extent of any potential breach as part of a subsequent forensic investigation.

There are many directions for future work. In the short term, we are developing guidance on data capture from SCADA/ICS applications. The intention is to help operators obtain sufficient information to diagnose the causes of an attack and return the plant to a safe condition/operation as quickly as possible. In the longer term, we are developing a canonical set of attack methods for SCADA/ICS devices that can be used to benchmark different products. These will provide an equivalent of the car manufacturers' crash-test rating. The suppliers of SCADA/ICS components might then be encouraged to provide devices that meet the higher security requirements associated with a top rating.

Finally, it is important to stress the immaturity of this field. The use of forensic attack analysis is largely guided by experience and intuition, rather than by a systematic or scientific process. The competency and expertise of the analysis is a key determinant of whether potential vulnerabilities are identified during verification and validation activities, or in response to intelligence about new attack methods.

Acknowledgments

Thanks are due to Marco Cook, Karl Drouven, Conall Ferguson, Luke McPhee, Mohammed Saleem and Athanasios Santas, who contributed to the technical work described in this paper. We also acknowledge the support of the CINIF Cyber-Security Working Group, which provided guidance and funding for aspects of this research.

About the Authors

Chris Johnson is Professor and Head of Computing Science at the University of Glasgow in Scotland. He leads a research group devoted to improving the cyber-security of safety-critical systems. He has developed forensic guidance on behalf of the U.K. civil nuclear industry and helped develop European policy for the cyber-security of aviation — including ground-based and airborne systems.

Maria Evangelopoulou is a Research Assistant working on a joint FAA/US Navy project in Glasgow University, looking at safety and security analysis of network data. She attained her MSc in Intelligence and Security Informatics from the University of Abertay and a BSc Technology Management from University of Macedonia in Greece. Maria's current research is concerned with the investigation of Cyber Situation Awareness Methods and Techniques in Cloud Networks and other kinds of systems. ⬢

Reference

1. Johnson, C.W. "Barriers to the Use of Intrusion Detection Systems in Safety-Critical Applications," *SAFECOMP 2015*, ed. F. Koorneef and C. van Gulijk, Springer Publishing Company, Heidelberg, Germany, 2015.
2. Centre for the Protection of National Infrastructure (CPNI). "Good Practice Guide – Process Control and SCADA Security," U.K., 2008. http://www.cpni.gov.uk/Documents/Publications/2008/2008031-GPG_SCADA_Security_Good_Practice.pdf, accessed February 2016.
3. Ahmed, I., S. Obermeier, M. Naedele and G. Richard. "SCADA Systems: Challenges for Forensic Investigators," *IEEE Computer*, 2012.
4. Communications-Electronics Security Group (CESG). *Forensic Readiness (Good Practice Guide 18)*, October 2015. https://www.cesg.gov.uk/guidance/forensic-readiness-good-practice-guide-18, accessed February 2016.
5. U.S. National Institute of Standards and Technology (NIST). *Computer Security Incident Handling Guide, Special Publication 800-61*, Gaithersburg, Maryland, 2012.
6. U.K. Association of Chief Police Officers. *Managers Guide: Good Practice and Advice Guide for Managers of e-Crime Investigation*, 2011.
7. European Network and Information Security Agency (ENISA). *Good Practices on Re-porting Security Incidents*, Heraklion, Greece, December 2009.
8. U.S. Department of Justice, Office of Justice Programs. *Electronic Crime Scene Investigation: A Guide for First Responders*, Second Edition, Washington D.C., 2008.
9. Wu, T., J. Disso, K. Jones and A. Campos. "Towards a SCADA forensics architecture," *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013*, pp. 12-21, British Computer Society (BCS), September 2013.
10. Mo, Y., T. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig and B. Sinopoli. "Cyber–physical Security of a Smart Grid Infrastructure," *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, 100(1), 195-209, 2012.
11. Safford, D., D.L Schales, and D. Hess. "Secure RPC Authentication (SRA) for TELNET and FTP," *USENIX Security*. 1993.
12. Horowitz, M. and S. Lunt. "FTP Security Extensions," *RFC 2228*, October, 1997.