SCADA LIVE FORENSICS: REAL TIME DATA ACQUISITION PROCESS TO DETECT, PREVENT OR EVALUATE CRITICAL SITUATIONS

Pedro Taveras N., MSC.
Pontificia Universidad Católica Madre y Maestra, Dominican Republic

Abstract:

SCADA (Supervisory Control and Data Acquisition System) systems were originally created to be deployed in non-networked environments. Therefore they lack of adequate security against Internet-based threats and cyber-related forensics. In recent years, SCADA systems have undergone a series of changes that might increase the risks to which they are exposed. Among these risks it can be observed that its increased connectivity may permit remote controls over the Internet, or the incorporation of general purpose tools, thus incorporating already known vulnerabilities of these. Any cyber-attack against SCADA systems demands forensic investigation to understand the cause and effects of the intrusion or disruption on such systems. However, a SCADA system has a critical requirement of being continuously operational and therefore a forensic investigator cannot turn off the SCADA system for data acquisition and analysis. This paper leads to the creation of a high level software application capable of detecting critical situations like abnormal changes of sensor reads, illegal penetrations, failures, physical memory content and abnormal traffic over the communication channel. One of the main challenges is to achieve the development of a tool that has minimal impact over the SCADA resources, during the data acquisition process.

Key Words: Cyber security, SCADA, Live Forensics, Intrusion Detection

Introduction

The security of SCADA systems is especially relevant in the field of Critical Infrastructure. A failure of critical infrastructure could have direct impact for society to the extent of affecting entire nations and their environment.

Any government network infrastructure or industrial based SCADA (Supervisory Control and Data Acquisition) or DCS (Distributed Control Systems), designed to automate, monitor and control critical physical processes, including manufacturing and testing, electric transmission, fuel and water transport, is subject to potential attacks.

Supervisory Control and Data Acquisition comprise all application solutions that collect measurements and operational data from locally and remotely controlled equipment. The data is processed to determine if the values are within tolerance levels and, if necessary, take corrective action to maintain stability and control. Its basic architecture comprises a centralized server or server farm, RTU (Remote Terminal Units) or PLC (Programmable Logic Controller) to manage devices; consoles from which operators monitor and control equipment and machinery.

SCADA systems were originally created to be deployed in non-networked environments. Therefore they lack of adequate security against Internet-based threats and cyber-related forensics.

Most industrial plants now employ networked process historian servers for storing process data and other possible business and process interfaces. The adoption of Ethernet and transmission control protocol/ Internet protocol TCP/IP for process control networks and wireless technologies such as IEEE 802.x and Bluetooth has further reduced the isolation of SCADA networks (Zhu, Anthony & Sastry, 2011).

In recent years, SCADA systems have undergone a series of changes that might increase the risks to which they are exposed. Among these risks it can be observed that its increased connectivity may permit remote controls over the Internet, or the incorporation of general purpose tools, thus incorporating already known vulnerabilities of these.

SCADA systems, in particular, perform vital functions in national critical infrastructures, such as electric power distribution, oil and natural gas distribution, water and waste-water treatment, and transportation systems. They are also at the core of health-care devices, weapons systems, and transportation management. The disruption of these control systems could have a significant impact on public health, safety and lead to large economic losses (Cardenas, Amin, Huang, Lin & Sastry 2011).

As a consequence, there is an increasing interest in the security/forensic research community on SCADA systems. This is mostly due to the heightened focus of governments worldwide on protecting their critical infrastructures, including SCADA systems (Ahmed, Obermeier & Naedele, David, Chaugule & Campbell, 2012).

Securing SCADA systems is a critical aspect of Smartgrid security. As sophisticated attacks continue to target industrial systems, the focus should be on planning and developing new security techniques that will adapt to the SCADA environment and protocols (Rodrigues, Best & Pendse, 2011).

Immediate needs identified in this area include the collection of evidence in the absence of persistent memory, hardware-based capture devices for control systems network audit trails, honeypots for control systems as part of the investigatory process, radio frequency forensics and intrusion detection systems for SCADA control systems (Nance, Hay & Bishop, 2009). However, post-mortem analysis tools require the investigator to shut down the system to inspect the contents of disks and identify artifacts of interest. This process breaks network connections and unmounts encrypted disks causing significant loss of potential evidence and possible disruption of critical systems (Chan & Venkataraman, 2010).

Computer forensics relies on log events for searching evidence of a security incident. However, the massive amounts of generated events along a lack of standardize logs complicate the analyst tasks (Herrerias & Gomez, 2007).

Digital forensics investigators are experiencing an increase in both the number and complexity of cases that require their attention. Most current digital forensic tools are designed to run on a single workstation, with the investigator issuing queries against copies of the acquired data evidence. With current generation tools, the single workstation models works reasonably well and allows tolerable case turnaround times for small forensic targets (for example < 40GB). For much larger targets, these tools are too slow to provide acceptable turnaround times (Richard & Roussev, 2006).

The challenge, however, is to mitigate the vulnerabilities that occur once a networked device becomes accessible from the internet. Attacks ranging from DDoS to backdoor intrusion are possible on industrial networks and power and SCADA systems. Although network firewalls can stop a significant amount of malicious traffic, there are several techniques hackers can use to bypass these security devices. The complexity of the infrastructure can make it difficult to detect malicious behavior (Rodrigues *et al.,* 2011).

Research Problem

Any cyber-attack against SCADA systems demands forensic investigation to understand the cause and effects of the intrusion or disruption on such systems. However, a SCADA system has a critical requirement of being continuously operational and therefore a forensic investigator cannot turn off the SCADA system for data acquisition and analysis. Current forensic tools are limited by their inability to preserve the hardware and software state of a system during investigation.

Research Goal and Target

Process control systems (SCADA Systems) generated much discussion as an area that the security community recognizes as a security threat, but not yet perceived by industry to be as much of a threat. As a result, this field lags behind most technical fields in the area of security (Nance, Hay & Bishop, 2009).

Study and research security vulnerabilities related to networked Supervisory Control and Data Acquisition (SCADA) systems, in order *to develop a forensic computing model to support incident response and digital evidence collection process*. Forensic investigation can play a vital role in a protection strategy for SCADA systems and may assist in the prosecution of attackers, but also in a deep analysis of the underlying SCADA IT system, for example, in the case of non-malicious events such as malfunctioning hard disks or other hardware. However the critical nature of SCADA systems and the 24/7 availability requirement entails forensic investigators spending as little time on a live SCADA system as possible, necessarily performing live data acquisition and then subsequent offline analysis of the acquired data (Ahmed *et al*. 2010).

Relevance and Significance

In the last years there has been an increasing interest in the security of process control and SCADA systems. Furthermore, recent computer attacks such as the Stuxnet worm, have shown there are parties with the motivation and resources to effectively attack control systems (Cardenas *et al.*, 2011)

SCADA systems are deeply ingrained in the fabric of critical infrastructure sectors. These computerized real-time process control systems, over geographically dispersed continuous distribution operations, are increasingly subject to serious damage and disruption by cyber means due to their standardization and connectivity to other networks (Zhu & Anthony, 2011).

In recent times it has been noticed that hackers implement newer techniques to launch attacks that can evade traditional security devices. It is therefore important to secure the SCADA systems from process related threats (Rodrigues *et al.*, 2011).

Compromising such a system with intrusion attacks can lead not only to high financial loses but, more importantly, to the endangerment of public safety. The danger is even higher considering that critical infrastructures are not immune to these threats and that they may be potentially more vulnerable than common information technology systems. Hence intrusion protection for critical infrastructures is an obvious need (Linda, Vollmer & Manic, 2009).

Reliability of many SCADA systems is not only dependent on safety, but also on security. Recent attacks against SCADA systems, by sophisticated malware, demands forensic investigation to understand the cause and effects of the intrusion on such systems so that their cyber defense can be improved.

A SCADA system has a critical requirement of being continuously operational and therefore a forensic investigator cannot turn off the SCADA system for data acquisition and analysis. In this case, live forensics is a viable solution for digital investigation in SCADA systems (Ahmed *et al.*, 2012).

In real life, logs are rarely processed by stakeholders due to 1) the large number of entries generated daily by systems and 2) a general lack of security skills and resources (time) (Hadziosmanovic *et al.*,2011). However, the use of the classical post-mortem analysis approach is becoming problematic especially for large-scale investigations involving a network of computers. In addition, the amount of time available for processing this data is often limited (Su & Wang, 2011).

Review of Literature

A substantial body of research exists in the area of forensics models for live acquisition over SCADA systems. Related research work is discussed on this section.

There is a growing need for systems that allow not only the detection of complex attacks, but after the fact understanding of what happened (Tang & Daniels, 2012).

Several researches address threats in SCADA systems. For the identification of threats, authors typically use questionnaires and interviews. To detect anomalous behavior, authors use approaches based on inspecting network traffic, validating protocol specifications and analyzing data readings. Process-related attacks typically cannot be detected by observing network traffic or protocol specifications in the system. To detect such attacks one needs to analyze data passing through the system, and include a semantic understanding of user actions (Hadziosmanovic et al.,2011).

A group of researchers who met at the Colloquium for Information Systems Security Education (CISSE 2008) to brainstorm ideas for the development of a research for Digital Forensic, concluded that actual SCADA systems are potentially more vulnerable to attack and more likely to need associated digital forensics capabilities. Unfortunately, most process control systems were not built to track their processes, but merely to control them. As a result, significant research and development categories were identified under this area, including among the most important: mechanism form the collection of evidence in the absence of persistent Memory and hardware-based capture devices for control (Nance, 2009). Figure 1 shows the list of topics in need for further development. It can be noticed that areas for Live Acquisition and Control Systems.

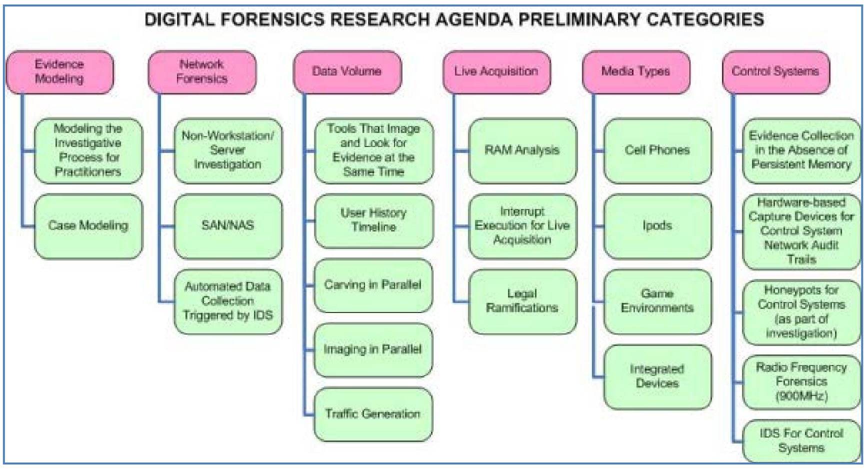
DIGITAL FORENSICS RESEARCH AGENDA PRELIMINARY CATEGORIES
Evidence Modeling
Modeling the Investigative Process for Practitioners
Case Modeling
Network Forensics
Non-Workstation/ Server Investigation
SAN/NAS
Automated Data Collection Triggered by IDS
Data Volume
Tools That Image and Look for Evidence at the Same Time
User History Timeline
Carving in Parallel
Imaging in Parallel
Traffic Generation
Live Acquisition
RAM Analysis
Interrupt Execution for Live Acquisition
Legal Ramifications
Media Types
Cell Phones
Ipods
Game Environments
Integrated Devices
Control Systems
Evidence Collection in the Absence of Persistent Memory
Hardware-based Capture Devices for Control System Network Audit Trails
Honeypots for Control Systems (as part of investigation)
Radio Frequency Forensics (900MHz)
IDS For Control Systems

Figure 1: Research Topics for Digital Forensics

Chen & Abu-Nimeh (2011), developed a deep research over the case of Suxnet malware. According their report, this was the first malware written exclusively to attack SCADA platform.

Stuxnet experience has shown that isolation from the Internet isn't an effective defense, and an extremely motivated attacker might have an unexpected combination of inside knowledge, advanced skills, and vast resources. Existing technologies would have difficulty defending against this caliber of attack (Chen & Abu-Nimeh, 2011). Therefore the need of new forensics methods that goes beyond the traditional prevention mechanism.

Ahmed, Obermeier & Naedele, David, Chaugule & Campbell (2012), propose a forensic mechanism denominated Live Forensics as a viable solution for SCADA systems. Live data acquisition involves acquiring both volatile data (such as the contents of physical memory) and non-volatile data (such as data stored on a hard disk). It is different from traditional dead disk acquisition, which involves bringing the system offline before the acquisition, where all volatile data is lost.

However, despite the importance of live data acquisition, it is still unclear how contemporary live data acquisition tools should be run on a SCADA system so that they minimize risk to SCADA system services (Ahmed *et al.*, 2011).

Aldenstein (2006) established that the possibility of implementing live forensics over SCADA systems relies on the capability of the operating system to provide the list of running processes. Therefore, he recognized the need for tools capable of examining the raw memory of a machine. These tools are analogous to the static tools that open the raw disk device and impose the file system structure on it to extract files, directories, and metadata (Adelstein, 2006).

Sutherland *et al* (2008), performed exploratory studies for live forensics within Windows operating systems environment and also determined the need for more invasive tools that allows better access to information related to memory, network and system activity were assessed to determine the impact on the file system, system registry, memory and the usage of DLLs.

Hadziosmanovic et al. (2011) proposed a tool-assisted approach to address process related threats. They presented an experimental study where SCADA threats that unlikely to happen or that does not occur on a systematic manner are detected and logged for investigation. An example could be when an attacker manages to get valid user credentials and performs disruptive actions against the process. However this effort was limited to post-mortem log analysis containing data for single event operations and does cover anomalous command process sequences. Likewise, it was determined that an attacker might gain unauthenticated remote access to devices and change their data set points. This can cause devices to fail at a very low threshold value or an alarm not to go off when it should. Another possibility is that the attacker, after gaining unauthenticated access, could change the operator display values so that when an alarm actually goes off, the human operator is unaware of it. This could delay the human response to an emergency which might adversely affect the safety of people in the vicinity of the plant (Zhu & Anthony, 2011).

SCADA systems are increasingly commonly being attached to networks, and typically offer no persistent storage for logging of network activity. The challenge for the digital forensic research community is to develop methods to allow an investigator to determine how these devices interacted with the network during a time period of interest (Nance, Hay & Bishop, 2009).

There is continuing interest in researching generic security architectures and strategies for managing SCADA and process control systems. Documentation from various countries on IT security does now begin to recommendations for security controls for (federal) information systems which include connected process control systems. Little or no work exists in the public domain which takes a big picture approach to the issue of developing a generic or generalizable approach to SCADA and process control system forensics (Sly & Stinikova, 2009).

Collection of adequate records or logs of events that happened near incident time is crucial for successful investigation. Logging capabilities of SCADA systems are geared towards discovering and diagnosing process disturbances, not security incidents, and are thus often not adequate for forensic investigation (Fabro & Cornelius, 2008)

Kilpatrick *et al* (2008) developed an architecture based on the Modbus TCP (Transmission Control Protocol) using two control devices and one HMI (Human Machine Interface) station. This architecture comprised two agents and a central warehouse. Various agents were deployed over the SCADA network. These agents captured network traffic containing real time data and stored them into the warehouse. Relational databases query mechanisms were used in the event of a forensic investigation. However, Ahmed *et al* (2011) established that state of the art forensic analysis tools do not support the unique features of diverse SCADA environments, which include supporting SCADA protocols and numerous SCADA applications' proprietary log formats etc. Thus plugins or modules for contemporary forensic tools need to be developed to augment the forensic analysis in SCADA systems.

Nehimbe & Nehibe (2012) proposed a timed series methodology to analyze forensic logs. During their research they concluded that actual for forensic tools may not necessarily generate the needed results. Due to two basic limitations on these tools: Some of them only have recovery and imaging capabilities and some intrusion analysis tools are flawed in terms of how they analyze intrusion logs.

Hunt & Slay (2010) proposed an approach named security information event management (SIEM) with the purpose to provide a tool that allows any networked system to auto adapt itself based on forensic logging. Their works showed that a SIEM system is an ideal point at which to store log data emanating from security devices and the point at which forensic logging needs to occur. However, although they were able to achieve the implementation of forensically sound log files in some systems; their approach is by no means universal. They concluded that their works still falls short of addressing the core domain of real-time forensically sound adaptive security.

With the purpose of rebuilding an attack scenario Herreria & Gomez (2007), proposed a log correlation model to support the evidence search process in a forensic investigation. In this work, they proposed a system composed by a set of agents in order to collect, filter, and to normalize events coming from diverse log files. Events may come from systems logs, application logs, and security logs. Once events are joined together in the same place and under the same format, they are sent to a correlation engine. The engine compares and processes the events in a global fashion in order to follow all actions taken by the attacker (Herrerias & Gomez, 2007).

Su & Wang (2011) developed a formula using probability theory and mathematical statistics to quantitatively calculate the degree of memory change on a live system. Their conclusions states that since the live memory state frequently changes is natural limitation for the purpose of live forensics. In their experiments they tried to restore to the same system state each time, however, the real state has been changed after one or two seconds. Therefore, they were only able determine and approximate of the system memory in every repeated process.

Further work is required to assess tools over various operating systems. This would be of value to the forensic investigator, but the way memory is handled and its analysis varies greatly between Windows Service packs let alone other operating systems; as a result the area of memory forensics is deeply complex and requires a significant amount of time and effort invested by the forensic examiner to begin to comprehend how memory works in modern Operating Systems (Sutherland, Evans, Tryfonas & Blyth, 2008).

Other research approaches are related to autonomic attack detection and response. Cardenas et al. (2011) showed that by incorporating a physical model of the system they were able to identify the most critical sensors and attacks. They also proposed the use of automatic response mechanisms based on estimates of the state of the system. However, they concluded that this methodology might be problematic, especially when the response to a false alarm is costly (Which could be the case for SCADA environment). As a result their model should be considered as a temporary solution before a human investigates the alarm.

Approach

While there have been a good number of research efforts investigating the suitability of forensics mechanism for SCADA, this work would be different in that it is intended to develop an investigation into *what is required to develop a forensic computing model to support incident response and digital evidence collection process, without interfering with the "always running" condition of SCADA platforms.* The intended approach is an extension of the works from Ahmed *et al.* (2011), who proposed a forensic mechanism denominated Live Forensics as a viable solution for SCADA systems. Live data acquisition involves acquiring both volatile data (such as the contents of physical memory) and non-volatile data (such as data stored on a hard disk). It is different from traditional dead disk acquisition, which involves bringing the system offline before the acquisition, where all volatile data is lost.

From a forensic perspective, a SCADA system can be viewed in different layers based on the connectivity of the various SCADA components and their network connectivity with other networks such as the Internet.

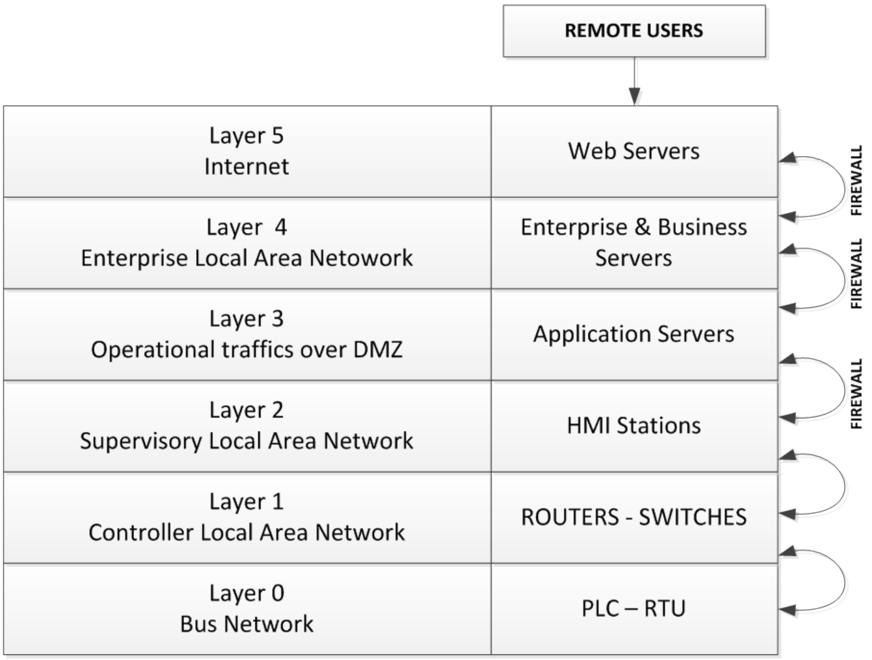

Figure 2: Layer Model for SCADA

The lowest layer represents the physical elements designed to interact directly with the industrial hardware or machinery. These devices are connected via bus network. Layer 1 receives electrical input signals with are decoded as a bit streams over a standard network protocols. The result is transferred to the uppers layers form analysis and controlling response. Layer 3 and above, represents the enterprise network which is also interconnected to the Supervisory systems. At this stage all traffic containing database content and applications supporting the business logics for the operation is managed. As stated by Amehd *et al,* (2010), live forensic analysis for the SCADA system must focus on the Layers 0, 1 and 2.

The initial approach has the intention of developing a forensic watch dog by means of a finite state automaton that would function as an agent that is constantly listening SCADA events. When any particular event is sensed, the input values are read and compared to a set of predefined rules in order to decide the change of state. Figure 2 represents the proposed automaton model.

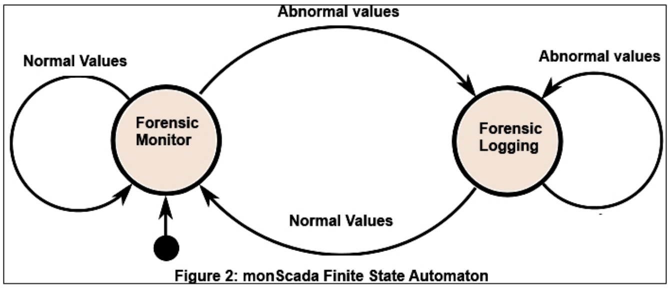

Figure 3: Finite State Automaton as SCADA Live Monitor

This two state automaton or agent constantly monitors the state of the SCADA system, including measures from: System Variables, Sensor Tags, Network Traffic and Command Executions. These values are checked against a set of behavioral rules. If a read is detected to be above the normal range, the agent automatically switches to Forensic mode and initiate the logging of forensic information. Ideally it would need a separate backup system to continuously dump the abnormal lectures from the SCADA tags and creates a record of this event appending all the information available about the system state. Including, but not limited to: CPU load, sensor names, sensor values, state of the physical memory, state of the virtual memory, state of the networking variables, state of mounted disk and network drives, list of active process in memory including name, executable name, working directory, command line, user name, user ids and group ids, threads, connections, file descriptors, etc. The information logged would be exclusively related to the period of time of the anomaly. Once the system start reading normal values, it switches back to Forensic Monitor and stop logging.

Normally a SCADA system reads every sensor or control registry on the system. These registries are known as tags and the logging frequency can vary from a read every second or even every 300 milliseconds. A typical SCADA system can have up to 40,000 tags. A system of such magnitude can generate approximately 400GB of data for every 24 hour period of operation. This calculation is based estimating an average size of 120 bytes per record. It can be seen, that the live acquisition is just part of the challenge. Dealing with vast amounts of data needs to be considered. These volumes requires manipulation by means of database query processors and moreover, require a fast capture and writing process that must be able to: (1) keep up with the logging process at the same time that new data comes into the system; and (2) all this must be done without incorporating additional workload over the monitored SCADA system. In other words, needs to be accomplished in a non-invasive manner.

Another challenge imposed to the intended solution is that SCADA system components can be found running on legacy hardware and operating systems. In such cases, a SCADA system provides limited system resources for data acquisition and therefore demands lightweight data acquisition tools and the gathering process might not represent a large resource consumer. However the data conversion process, if a relational database would be implemented for a better data analysis, would requires specialized hardware to reduce the processing time and speed up files conversion and querying processes.

Another important aspect from this experiment that would be suitable for further development is that current forensic analysis tools do not provide a standard support for the variety of SCADA hardware versions, protocols and log formats. Therefore, we have an interesting opportunity to expand this experiment with the development of plugins and applications and interface layers in order to increase the number of SCADA forensic tools as an expansion of the works of Hadziosmanovic et al., (2011), who stated that despite the fact that there are several vendors, system architectures in various SCADA systems are similar and the terminology is interchangeable.

In conclusion, future work leads to the creation of a high level software application capable of detecting critical situations like abnormal changes of sensor reads, illegal penetrations, failures, physical memory content and abnormal traffic over the communication channel. One of the main challenges is to achieve the development of a tool that has minimal impact over the SCADA resources, during the data acquisition process. In previous exercises it was observed that the processes for acquiring low level information, such as processes or memory information does not represents an extensive load on the actual system that is processing the task. However, it is expected that the amount of demanded resources increase as the number of SCADA tags and the frequency of logging increases. Therefore on real live SCADA system, the acquisition process could be competing for resources that should be available for the normal operation of the SCADA systems.

Barriers and Resources

From the literature review it can be determined multiple barriers and issues that could be anticipated for the development of this work. The next section presents a summary of the know limitations and challenges determined by previous research efforts on this field.

Forensic data gathered from a live system can provide evidence that is not available in a static disk image. Live forensics also operates with different constraints—specifically, the evidence gathered represents a snapshot of a dynamic system that *cannot* be reproduced at a later date. Standards for acceptance are evolving, and legal precedents are still being established (Adelstein, 2006).

Given that volatile data in a running system changes continuously, Ahmed et al., (2011) established two main challenges involved during live data acquisition over operational SCADA systems: (1) Live data acquisition needs to be performed as quickly as possible after an incident in order to capture any traces of the incident on volatile data before the processes or services on the running system overwrite useful data; (2) Cryptographic hash of the actual evidence on the compromised system and its acquired copy, which is used for all the examination and analysis. If, however, the compromised system remains live, the state of the data may change between the copying and the hash calculation, rendering hashing ineffective as an integrity check (Ahmed *et al.*, 2011).

From a forensic standpoint, modifying the original system memory state is unavoidable, therefore, it is needed that changing as little as possible on the process of collecting live forensics. For a real live forensics case, it should content digital forensics that collected by the forensic tools, the analysis and evaluation of the uncertainty. However, it is difficult to measure how much of the volatile memory is modified by a forensics tool. Moreover, it is difficult (if not impossible) to calculate the extent of the memory's impact caused by a running process on the volatile memory. So, measuring the extent of the volatile memory changes caused by running a live forensic tool becomes more and more important (Su & Wan, 2011).

Because the architecture of production operating systems prevents applications from accessing kernel memory and storage devices without using the kernel, kernel-based rootkits will always be a threat to live analysis. Future directions in live analysis techniques involve the use of specialized hardware to collect the raw memory and storage data for a dead analysis (Carrier, 2006).

Research has shown that an attacker with control of the target system can manipulate memory mappings so that the CPU and devices on the PCI or Firewire buses don't necessarily get the same view of memory. In such cases, attempts to acquire the memory's contents could crash the target system or enable the attacker to mask sections of memory without that masking being apparent to the investigator (Hay, Bishop & Nance, 2009).

Furthermore, factors like the continuous availability demand, time-criticality, constrained computation resources on edge devices, large physical base, wide interface between digital and analog signals, social acceptance including cost effectiveness and user reluctance to change, legacy issues and so on make SCADA system a peculiar security engineering task (Zhu & Anthony, 2011).

Finally, no matter how well the simulations and models emulate a possible solution, any given conclusion needs to be tested over a real SCADA systems. Real SCADA systems are expensive to build and thus require significant research funding. Access to sample information or security failure scenarios could be difficult because the critical nature of SCADA systems demands the owners and operators not share any information about their system.

Conclusion and Expected Contributions

Applying traditional information security mechanism directly to SCADA systems is not possible. SCADA systems cannot afford non-deterministic delays in performance, security controls that require a lot of memory, block access for safety or relatively long intermediate processes. Security measures that can be applied to SCADA systems should consider this special operating paradigm.

Process-related attacks typically cannot be detected by observing network traffic or protocol specifications in the system. To detect such attacks one needs to analyze data passing through the system, and include a semantic understanding of user actions (Hadziosmanovic *et al.*, 2011).

In conclusion, there is no generic model for understanding the forensic computing processes necessary to gather digital evidence from Process Control and SCADA systems. Therefore, the need for developing a forensic computing model to support incident response and digital evidence collection process is justified.

As a consequence this work could help to improve critical infrastructure protection and provide appropriate tools that could be used for dealing with incident responses and forensics analysis over interconnected SCADA systems.

References:

Adelstein, F. (2006). Live forensics: diagnosing your system without killing it first. *Commun. ACM*, *49*(2), 63–66.

Ahmed, I., Obermeier, S., Naedele, M., & III, G. R. (5555). SCADA systems: Challenges for forensic investigators. *Computer*, *99*(1), 1.

Andersson, G., Esfahani, P. M., Vrakopoulou, M., Margellos, K., Lygeros, J., Teixeira, A., … Johansson, K. H. (2012). Cyber-security of SCADA systems. In *Innovative Smart Grid Technologies, IEEE PES* (Vol. 0, pp. 1–2). Los Alamitos, CA, USA: IEEE Computer Society.

Boca, L., Croitoru, B., & Risteiu, M. (2010). Monitoring approach of Supervisory Control and Data Acquisition downloadable data files for mission critical situations detection. In *International Conference on Automation, Quality and Testing, Robotics* (Vol. 3, pp. 1–6). Los Alamitos, CA, USA: IEEE Computer Society.

Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., & Sastry, S. (2011). Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (pp. 355–366). New York, NY, USA: ACM.

Carrier, B. D. (2006). Risks of live digital forensic analysis. *Commun. ACM*, *49*(2), 56–61.

Chan, E., Venkataraman, S., David, F., Chaugule, A., & Campbell, R. (2010). Forenscope: a framework for live forensics. In *Proceedings of the 26th Annual Computer Security Applications Conference* (pp. 307–316). New York, NY, USA: ACM.

Chen, T. M., & Abu-Nimeh, S. (2011). Lessons from Stuxnet. *Computer*, *44*(4), 91–93.

Hay, B., Bishop, M., & Nance, K. (2009). Live Analysis: Progress and Challenges. *IEEE Security & Privacy*, *7*(2), 30–37.

Hay, B., & Nance, K. (2008). Forensics examination of volatile system data using virtual introspection. *SIGOPS Oper. Syst. Rev.*, *42*(3), 74–82.

Herrerias, J., & Gomez, R. (2007). A Log Correlation Model to Support the Evidence Search Process in a Forensic Investigation. In *Systematic Approaches to Digital Forensic Engineering, IEEE International Workshop On* (Vol. 0, pp. 31–42). Los Alamitos, CA, USA: IEEE Computer Society.

Hunt, R., & Slay, J. (2010). The Design of Real-Time Adaptive Forensically Sound Secure Critical Infrastructure. In *Network and System Security, International Conference On* (Vol. 0, pp. 328–333). Los Alamitos, CA, USA: IEEE Computer Society.

Kilpatrick, T., Gonzalez, J., Chandia, R., Papa, M., & Shenoi, S. (2008). Forensic analysis of SCADA systems and networks. *Int. J. Secur. Netw.*, *3*(2), 95–102.

Linda, O., Vollmer, T., & Manic, M. (2009). Neural Network based Intrusion Detection System for critical infrastructures. In *Neural Networks, IEEE - INNS - ENNS International Joint Conference On* (Vol. 0, pp. 1827–1834). Los Alamitos, CA, USA: IEEE Computer Society.

Morris, T., Vaughn, R., & Dandass, Y. S. (2011). A testbed for SCADA control system cybersecurity research and pedagogy. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research* (pp. 27:1–27:1). New York, NY, USA: ACM.

Nance, K., Hay, B., & Bishop, M. (2009). Digital Forensics: Defining a Research Agenda. In *Hawaii International Conference on System Sciences* (Vol. 0, pp. 1–6). Los Alamitos, CA, USA: IEEE Computer Society.

Nehinbe, J. O., & Nehibe, J. I. (2012). A Forensic Model for Forecasting Alerts Workload and Patterns of Intrusions. In *Computer Modeling and Simulation, International Conference On* (Vol. 0, pp. 223–228). Los Alamitos, CA, USA: IEEE Computer Society.

Peterson, D. (2009). Quickdraw: Generating Security Log Events for Legacy SCADA and Control System Devices. In *Conference For Homeland Security, Cybersecurity Applications & Technology* (Vol. 0, pp. 227–229). Los Alamitos, CA, USA: IEEE Computer Society.

Richard,III, G. G., & Roussev, V. (2006). Next-generation digital forensics. *Commun. ACM*, *49*(2), 76–80.

Rodrigues, A., Best, T., & Pendse, R. (2011). SCADA security device: design and implementation. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research* (pp. 25:1–25:1). New York, NY, USA: ACM.

Su, Z., & Wang, L. H. (2011). Evaluating the Effect of Loading Forensic Tools on the Volatile Memory for Digital Evidences. In *2010 International Conference on Computational Intelligence and Security* (Vol. 0, pp. 798–802). Los Alamitos, CA, USA: IEEE Computer Society.

Sutherland, I., Evans, J., Tryfonas, T., & Blyth, A. (2008). Acquiring volatile operating system data tools and techniques. *SIGOPS Oper. Syst. Rev.*, *42*(3), 65–73.

Tang, Y., & Daniels, T. E. (2005). A Simple Framework for Distributed Forensics. In *2012 32nd International Conference on Distributed Computing Systems Workshops* (Vol. 2, pp. 163–169). Los Alamitos, CA, USA: IEEE Computer Society.

Yen, P.-H., Yang, C.-H., & Ahn, T.-N. (2009). Design and implementation of a live-analysis digital forensic system. In *Proceedings of the 2009 International Conference on Hybrid Information Technology* (pp. 239–243). New York, NY, USA: ACM.

Zhu, B., Anthony, Joseph, & Sastry, S. (2011). A Taxonomy of Cyber Attacks on SCADA Systems. In *2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*. San Diego, California: IEEE Computer Society.