Smart Grid Security – Attack and Defense Stratagems

Introduction

* Why your topic is important (convince us!)
* Where is it used? Applications
* What you will talk about / do
* Overview of the rest of your paper

Background and related work

* Any relevant and specific info, e.g. software / hardware statistics, equipment used
* What other people had to say on this topic
* What other people did on this topic
* Problems and shortcomings of their work
* How your work is different and better

Proposed methodology

* Your approach to the problem
* What you did
* Code / Algorithms
* What did / didn’t work
* Results – include graphs, equations, pictures, as appropriate

Conclusions

* What was accomplished / learned
* What you would have done differently
* Future work

References

Appendix

* **Main point:**

Technology Discussion

1. Background on Smart Grid Security
2. False data injection attack
   1. Attack on integrity; Integrity violation
   2. Examples: Puerto Rico, disaster events
   3. Defense includes protection of state estimation or raw meter data
   4. Defense:
3. Popping HMI attack
   1. Target Critical Time Delay <3msec
   2. People can find out information about the network
   3. Defense: Deterministic Browser
4. Privacy Attack
   1. Vehicles/Terminals
   2. Place some sort of privacy algo, present current research on topic
   3. Defense: Hadamard basis to anonymize

Problem Approach

1. Build Deterfox, setup necessary Windows applications to emulate IEC standard and one machine, See if I can find a target machine time
2. Perturb the power grid network dataset either using RAPPOR or other method.

Conclusion

* Write something lel

**Introduction**

Today’s society rely on industrial control systems to provide services like gas, water, and electricity with differing loads of demand and supply. Supervisory Control and Data Acquisition (SCADA) is a common framework used to describe control systems feedback in industrial systems. In the 21st century, SCADA systems adopt increasingly more computers to control industrial processes, and therefore are targets for nation-states, individuals, and organizations.

SCADA systems do not only exist in the industrial plant, but also play a role in building automation systems. Most universities or organizations that manage a collection of buildings within the campus will utilize a networked SCADA system to maintain control over temperature, water, electricity, and mechanical systems such as elevators.

However, all SCADA systems at some point need to be human operable. Whether it is a mechanical switch, a console command, or a web interface, the feedback system needs to have human intervention and control right down to the individual mechanical components. Today’s corporations mainly use the Human Machine Interface (HMI), to represent this component of SCADA and are typically web interfaces that can be accessed through major web browsers.

In this paper, we discuss the development of the HMI, summarize the research that attempts to secure vulnerabilities, and present further protections for the HMI with an implementation created in a SCADA honeypot to model threats.

**Background and related work**

1. *History of SCADA development*

To understand the research and implementations to protect SCADA vulnerabilities, a quick overview of the development of SCADA is required. Industrial control systems developed rapidly as the demand for services like natural gas, water purification, and electricity generation escalated during the 1960s along with technological innovations in both engineering and computers.

SCADA has developed through generations of improvement with new technology. These generations are referred to as the monolithic, distributed, networked, and Internet of Things (IoT) stages [Wagneer, 18]. Monolithic and distributed describe systems that used a mixture of mechanical and command line interfaces. The distributed, networked, and IoT SCADA systems rely on a mixture of present day protocols like HTTP and TCP as well as protocols specified by IEC 60870 like Modbus or DNP3.

Most modern day SCADA systems include Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs). These devices control the machines running on the industrial plant based on current and past data readings. Other computers placed either on or off site delegate instructions to PLCs and RTUs in order to meet the supply and demand of the plant. SCADA systems also include data historian and logging servers that serve to track production of the plant and calls to services.

1. *Research and Response*

However, the networks for SCADA systems are very vulnerable to attack since most of the computer to device protocols like BACnet does not include any sort of cryptographic security [Sciencedirect] article] and could not be upgraded to include cryptographic protocols since the data buses are very small. With the invention of the World Wide Web in the 1990s, protocols with more security features could be used to wrap around the vulnerable ones, so most research focused on using cryptography to secure the networks controlling SCADA systems [https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1612765&tag=1]. Controlling the TCP or UDP connections within the network became a priority, and eventually protocols were established like ISA 95/99 and ASME standards and government guides were produced from organizations like NIST, FIPS, and ISO [https://ac.els-cdn.com/S0167404806000514/1-s2.0-S0167404806000514-main.pdf?\_tid=52dce760-cac4-4c08-a1b1-a7500b38aa30&acdnat=1524864535\_1e42ac4b1d32fc50a118a2be11ddda8c]. Current research focuses on models or introduces a new security framework to protect SCADA systems.

1. *Problem of the HMI*

By looking at the SCADA system from a layered view of computers, current and past research has succeeded in modeling, implementing, and improving the protection of networks within SCADA systems. However, the applications that run within these networks are now targeted towards attack. Different systems have different tech stacks, but all need a common interface in which to issue commands either on-site or remotely: the HMI.

In the last two decades, the popularity of web browsers redefines the HMI into a browser-compatible space. All interactions now go through a web application that is easier to modify and update. Companies that create SCADA packages now offer the hardware PLCs and RTUs along with web and mobile accessible pages. While these applications increase usability, web technologies increase the attack space further for SCADA systems more so than that of network protocols, which are slower to change and develop.

Out of the standards and guides researched, only ICS-CERT had any mention of potential vulnerabilities in web applications [https://ics-cert.us-cert.gov/sites/default/files/recommended\_practices/RP\_CaseStudy\_XSS\_20071024\_S508C.pdf] using cross-site scripting. Since there is no research into security vulnerabilities in HMI, there are no codes in any standard or guide and therefore no models in existing tools to detect web application vulnerabilities. If there are any attacks on HMIs in SCADA systems, they are normally reported to ICS-CERT and have dedicated CVEs, such as the case with Siemens WinCC version 7 (CVE-2014-8551).

1. *Attacking the HMI*

In this research, we will implement attacks on the HMI of a mock SCADA setup. The attack will not include compromising authentication or access control and assume the attacker controls an off-site computer with access. The point of the attack is to see if a vulnerable HMI triggers the intrusion detection tools and logs to ensure that a loss of integrity within the system can be tracked.

1. False data injection attack
   1. Attack on integrity; Integrity violation
   2. Examples: Puerto Rico, disaster events
   3. Defense includes protection of state estimation or raw meter data
   4. Defense:
2. Popping HMI attack
   1. Target Critical Time Delay <3msec
   2. People can find out information about the network
   3. Defense: Deterministic Browser
3. Privacy Attack
   1. Vehicles/Terminals
   2. Place some sort of privacy algo, present current research on topic
   3. Defense: Hadamard basis to anonymize

**Background**

1. *Realms of the Smart Grid*
   1. Say top-bottom but disclaimer its not actually.
   2. Reference RTU, AMI, and SCADA

**False Data Injection**

## a) Premise of Integrity Violations

In our first attack, we look both conceptual research and current impacts in violations of integrity. Integrity is defined as the manipulation of data either by a malicious adversary or an error-producing source that compromises the authenticity of information that can be gained from the data. In networks protocol design, some protocols use checksums or error correction codes in order to prevent or fix integrity violations. Unfortunately, only one power substation protocol, DNP3, gives any kind of integrity checking with checksums [DNP3 Primer], but DNP3 is not as widespread as other, more insecure protocols like BACnet or Modbus. Today, most power substation setups utilize these insecure protocols as minimal as possible and only in machine-to-machine networks.

However, the Smart Grid aims to ubiquitous and secure across its realms. Therefore, new problems arise while implementing past-day solutions. An adversary might not have access to all parts of the Smart Grid, he or she can attack either at the top at the power generation realm or at the bottom at the distribution, end user, or service provider. Therefore, making use of either the top or bottom might compromise the other. By injecting integrity violations, the adversary can push incorrect data that will affect operations and lead to issues in power delivery service.

## b) Integrity violation via calculations

With respect to integrity violations, Mrabet et al. [Mrabet] describe the *false data injection attack*. In the scope of the Smart Grid, false data injection is typically implemented at the bottom realms of power distribution, end user networks, and the service provider. By influencing the calculations on the metrics of power consumption, an adversary violates the integrity of the data and ensures a false result once the metrics reach to the SCADA core. Specifically, Liu et al. [Liu et al.] give the calculation of state estimation as describe by

By impacting values in z to find an attack vector, the estimation function H can be altered and produce a false result of the estimate. Liu et al. also debunk previously stated beliefs that an attacker would need a substantial probability to find an attack vector to compromise the network. In the experimental evaluation, the attacker only needs a 20 percent chance of random false data injections before the attack vector is found. A sample setup of this attack can be modeled by an attacker having access to a variety of RTUs or parts of the AMI such as a smart meter for a house. The formation of the state estimation for Smart Grids is primarily used as the basis for modeling an Intrusion-Based Detection System (IDS).

## c) Integrity violation via hardware

Along with the manipulation of data via software inferred calculation, an integrity violation with easy accessibility to the attacker is found in the smart meter devices themselves as a part of the Smart Grid. An article written by reporter Brian Krebs informs of widespread hacking of smart meter devices in Puerto Rico [Krebs]. Puerto Rico is among one of many locations around the world that have implemented smart meter infrastructure due to natural disasters destroying the infrastructure. In Puerto Rico, some citizens have modified their smart meters to cut off metrics being sent back to the public power utility, PREPA. As a result, the FBI claims 400 million US dollars will be lost in the long term.

Halim et al. [Halim] give a review of various hardware hacks that can implemented on smart meters [Halim]. Smart meter infrastructure, just like the rest of Smart Grid components, are still early in their development and deployment, but still ship with hardware vulnerabilities. First off, smart meters come without any encrypted or obfuscation of memory locations. Therefore, it is easy to get memory readouts via the pins connecting the devices and inject data to cause integrity violations. Another method is simply unplugging the meter’s metric connection or placing a strong magnet on the meter—the technique used in Puerto Rico. No data at all still means disruptions of power consumption. Whole communities within the end user realm could use these techniques which are highly incentivized by the financial gain and highly accessible due to rise of a technology-savvy generation.

## c) Preventing false data injections

By the Smart Grid’s own interconnectedness, false data injection can cause widespread disruption in the network. As mentioned before, the calculation for state estimation is typically used for modeling an IDS. The solution proposed by current research involves various algorithms that try to decrease the number of false positives while maintaining true negative accuracy. Chen et al. [Chen et al.] tries a machine learning approach that differs in the traditional statistical-based IDS. The authors formulate a consistent-inconsistent region to measure how much of a grid is reliable or not. Then, each state and its neighbor in a set is compared by trust-based voting to see if their state estimations are reliable. Finally, elements are targeted as “Good”, “Abnormal”, and “Unknown” if they fall into the consistency region or not.

In experimental evaluation, this method proved to produce false-positive rates two-thirds lower than the next best algorithm. It also provides configurable regions to let end users decide on­ how reliable they want the algorithm to perform. Various improvements can be made by strictly checking the “Unknown” components. According to the solution, “Unknown” components are caused when there is not enough data on state estimations to say a neighbor is reliable or not. Data-sparse regions in the Smart Grid are plentiful in more rural areas or places with poor data connection. If the solution is to be made for real-time correlation, some development in data-sparse areas should be considered, perhaps even utilizing data from the previous statistical-based models if applicable.

For the issue of hardware violations, it is representative of the more “security through obscurity” problem largely present in the current power generation grid. Mandatory requirement for future standard IEC 62056-21 should be implemented across the Smart Grid which includes simple passwords, encrypted passwords, and handshaking to smart meters. Future specifications of the standard should include higher cryptographic protocols. The meters themselves should be secured based on a minimal set of requirements to delay tampering or notifying the end user or power company.

**Popping HMI Attack**

Mrabet acknowledge the HMI or *Human Machine Interface* in the paper as part of the exploitation an attacker can use to gain access to critical systems in the Smart Grid. The HMI is defined as the mechanism for human interaction to the SCADA interface. In the past, this used to relate to buttons and knobs of generation one SCADA via mechanical switches. However, today’s generation and the future will utilize web browsers as the main HMI since they are well-supported and tested through everyday use.

Mrabet brings up the point that exploiting HMI vulnerabilities “does not require advanced networking skills or significant experience in security and industrial control system to perform” and puts the attack as a high severity, but then puts it as low probability for implementation which seems contradictory. HMI attacks high severity due to their ability to compromise confidentiality, integrity, and accessibility throughout the Smart Grid. However, the research for HMI attacks in SCADA and other components of Smart Grid systems only count to a few studies and therefore represent a blind spot for security.

*Popping the HMI* is a term used to relate attacks on the HMI interface. Any attacks on web browsers apply to the HMI of the smart grid system at all levels. Normally, attacks are part of an implementation of a library or system that the HMI requires. For example, most web browsers use OpenGL as the main shading language for users and files to gain access to the GPU functions to render content. While thoroughly developed and optimized, there are still some problems outside of actual software vulnerabilities. GPU operations have regularly timed I/O operations that can be captured and analyzed to see what the user was doing at the time. This timing aspect is what makes up one of the greatest threats to the Smart Grid.

## a) Timing attacks on the HMI

While there are many different vectors of attack on the HMI, ranging from cross-site scripting to library substitution, we focus on timing attacks on the HMI that reveal what the system was doing at the time. The attacker’s goal is reconnaissance of the target network topology in the Smart Grid. Specifically, the attacker wants to know which devices that are Type 1A/P2 that need a critical time delay of 3 milliseconds to conduct operations—these devices will be highly targeted areas of the network [Wang et al]. Over a large sample of requests, the attack will either divulge the target devices or give enough information to conduct additional attacks.

The procedure of the attack is nothing new to vulnerabilities of web browser development. Javascript and the loading of HTML and CSS at particular times by Content Delivery Networks give enough information to determine which web site a user is browsing, and since a user-agent string is sent via Content headers in HTTP the attack space narrows to specific browsers. The attacker uses a time statistic to base their measurements, sends some code to the target machine, and then sees how long the process takes.

Since HMIs are sold along with the machines that make up components of the SCADA core, fingerprinting browser can go down to the application level and development stacks of the power substation setup. For example, the Siemens WinCC HMI had a CVE (Common Vulnerability and Execution) that came from its Windows Server Version that required an update. Since Siemens HMI will be perform and execute at consistent times with caching and perform at different times relevant to other popular HMIs, the attacker has learned crucial and valuable information that can be now shifted into other aspects of the the Siemens SCADA system.

## b) Defense with Deterministic Browsers

Due to the potential of attacks, browsers built for security tend to reduce their fingerprinting as far as possible. The Tor browser, built on top of the Tor networking protocol, tells its users to always keep a low profile by keeping every aspect of the modern web browser in check. Tor has options to reduce screen resolution to keep with the most popular formats, disabling Javascript entirely, and, according to Wang et al. [Wang et al.], even prevents timing attacks by adding “jitters to the browser clock”.

However, mitigating attacks on the timing of web browsers has proven difficult with options becoming more distinct over time with the different hardware and software provided in the modern day. As mentioned before, SCADA systems will use the most supported web browser, but the Smart Grid SCADA core requires even more security. The *deterministic browser* provides a possible defense against timing attacks for web browsers of the Smart Grid.

Wang described the process of developing deterministic browser to defeat timing attacks. The deterministic browser produces a time based on the attacker’s input so the attacker will get a different time than the actual time the process took to complete. The browser is constantly ticking time on the clock even if there are no process tasks to be done. The actual time the process takes is slotted inside a constant time window that is resistant to timing attacks. Currently, Wang has built a custom build of Firefox named Deterfox that defends against timing attacks. Even over a large sample of devices, files, times, and popular websites, there was no useful information gained from conducting performance metrics. All operations seem to complete in constant time as seen by the adversary.

If adopted as a primary browser for the Smart Grid and researched further, the deterministic browser can become a new defense against a difficult attack. However, it should be noted that the Smart Grid should adopt other secure methods of HMI web browser interaction such as private VPN tunneling, secure HTTPS, and firewall configurations to not divulge information about target machines in plaintext.

The issues for the deterministic browser is the Javascript time execution. All operations conducted through Javascript calls have to be considered to predict the time the process will take and the resultant confusion for the attacker. Javascript and its now popular superscript Typescript will introduce new synchronous and asynchronous functions that will break compatibility with this deterministic browser. Also, any calls to the performance API in Javascript will break compatibility with the deterministic browser and revert back to regular true system time. Many HMIs might require access to these APIs and functions.

* Cao, Y., Chen, Z., Li, S. and Wu, S. (2018). *Deterministic Browser*. Available at: https://arxiv.org/abs/1708.06774
* Chen, P., Yang, S., McCann J., et al. (2018). *Detection of false data injection attacks in smart-grid systems - IEEE Journals & Magazine*.
* El Mrabet, Z., Kaabouch, N., El Ghazi, H., & El Ghazi, H. (2018). *Cyber-security in smart grid: Survey and challenges.* Computers & Electrical Engineering*,* 67, 469-482.
* Halim, F. Yussof S., and Rusli E. (2018). *Cyber Security Issues in Smart Meter and Their Solutions.* IJCSNS International Journal of Computer Science and Network Security, Vol.18 No.3, March 2018
* Ken Curtis. *A DNP3 Protocol Primer*. (2005). DNP3 Primer, Revision A, 20 March 2005
* Krebs, B. (2017). *FBI: Smart Meter Hacks Likely to Spread.* Available at: https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/comment-page-1/
* Liu, Y., Ning P. and Reiter, M. (2011). *False data injection attacks against state estimation in electric power grids*. ACM Transactions on Information and System Security, Vol. 14, No. 1, Article 13, Publication date: May 2011.
* Wang, W. and Zhuo, L. (2018). *Cyber Security in the Smart Grid: Survey and Challenges. Computer Networks: The International Journal of Computer and Telecommunications Networking.* Volume 57 Issue 5, April, 2013. Pages 1344-1371