Securing SCADA and EMS Communications to Thwart Advanced Persistent Threats and Surpass NERC CIP Requirements

Dwight Anderson and Huba Leidenfrost
Schweitzer Engineering Laboratories, Inc.

Published in
Sensible Cybersecurity for Power Systems: A Collection of Technical Papers Representing Modern Solutions, 2018

Previously presented at
Rockwell Automation TechED, May 2015

Originally presented at the
Power and Energy Automation Conference, March 2013

Securing SCADA and EMS Communications to Thwart Advanced Persistent Threats and Surpass NERC CIP Requirements

Dwight Anderson and Huba Leidenfrost, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—This paper shares new advances in technical solutions that secure supervisory control and data acquisition (SCADA) and energy management system (EMS) communications. Many of these advances surpass current regulatory requirements, such as those found in the North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP) requirements. This paper explores the use of firewalls, virtual private networks, authentication proxy services, and application whitelisting as the means to not only exceed regulatory requirements but help mitigate advanced persistent threats (APTs), such as Stuxnet.**

I. Introduction

The focus of power system automation design is on keeping power safe, reliable, economical, and, as a part of this, secure. Automation design includes many devices interacting to behave as a system to protect the power system and make its operation robust. Power system automation continues to evolve and helps improve the daily life of power consumers, system operators, engineers, and even power linemen. The security of these automation systems is critical for reaching the design goals. There are improvements to be made, like there are improvements to be made for safety and reliability. Security is a part of an overall quality process with opportunities for improvement as new research and technology are uncovered [1].

An example of new research is the Digital Bond SCADA (supervisory control and data acquisition) Security Scientific Symposium (S4) that was held in Florida in early 2012. Just over 60 engineers met to dissect and review the Stuxnet worm in an attempt to determine the best path forward for the security of control systems connected to critical infrastructure. At the conference, many demonstrations were given to examine Stuxnet and included research about methods that could generate zero-day exploits in human-machine interface (HMI) programs that run on Microsoft® Windows® operating systems. Also, the conference attendees discussed various methods and designs to mitigate zero-day exploits and other vulnerabilities.

A zero-day exploit means someone has found a way to attack a product or system by exploiting a vulnerability that is otherwise unknown. By this definition, current antivirus software that looks for known exploits is not able to identify and stop these attacks. An attacker exploiting the vulnerability can gain unauthorized access into system operation. A zero-day exploit becomes an advanced persistent threat (APT) when it is able to remain hidden inside a system and is not easily identified or removed by current security measures. APTs can also be defined as adversaries with unlimited funding and time to find weaknesses.

The question that comes to mind is whether a power automation system design can thwart zero-day attacks, even APTs. We believe the answer is a cautious "yes," but requires using multiple layers of security and a defense-in-depth security approach. It also requires training people who come into contact with an automation system to ensure that they do not inadvertently or unwittingly become part of the process for a hacker to gain unauthorized access.

This paper delves into new technical areas that are able to improve overall automation system robustness, including safety, reliability, and security. The paper also addresses these items in light of North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP)—specifically, improving the robustness of an automation system for safety, reliability, and security to exceed the NERC CIP requirements.

In order to explore the security of control network designs, this paper focuses on HMI systems. The paper examines the security of computer systems, specifically the security of the Windows operating systems. The Windows operating systems present multiple challenges to providing operational robustness and security [1]. This paper looks at the security issues that are present in the updates for the operating system, HMI software, and signature updates for antivirus software. Solutions to these issues, as well as others, must become part of the policies, plans, and procedures of an enterprise in order to provide robust security.

There should always be an implied first rule of safety—safety first, reliability second, and security third. It is critical to always test power automation system and design changes, including updates and patches, in nonproduction environments. Then, after successful testing, cautiously deploy these changes into systems located in operational environments.

II. Overview of Network Designs

A recommended approach for power automation system network design is to adopt a zoned approach to the infrastructure design. By using a zoned approach, as shown in Fig. 1, the design yields multiple areas for security solutions that integrate together, forming a tightly coupled system that streamlines the organizing, documenting, and consolidating of strong security and protection measures. These technical controls also allow for supporting policies, plans, and processes for meeting regulatory requirements, such as those found in the NERC CIP requirements. The cybersecurity solutions scattered throughout the zones provide a defense-in-depth design and should address the higher-layer and lower-layer operational requirements. Also, using varied and appropriate security devices in a zoned approach makes it more difficult for an intruder to gain control of critical systems and cause disruption. Conversely, placing all security functions in one zone or area leaves the other zones open to a greater likelihood of compromise. The network security design should also allow for the system to securely manage, control, and document who has access and provide detailed audit trails of who did what and when to critical devices [2].

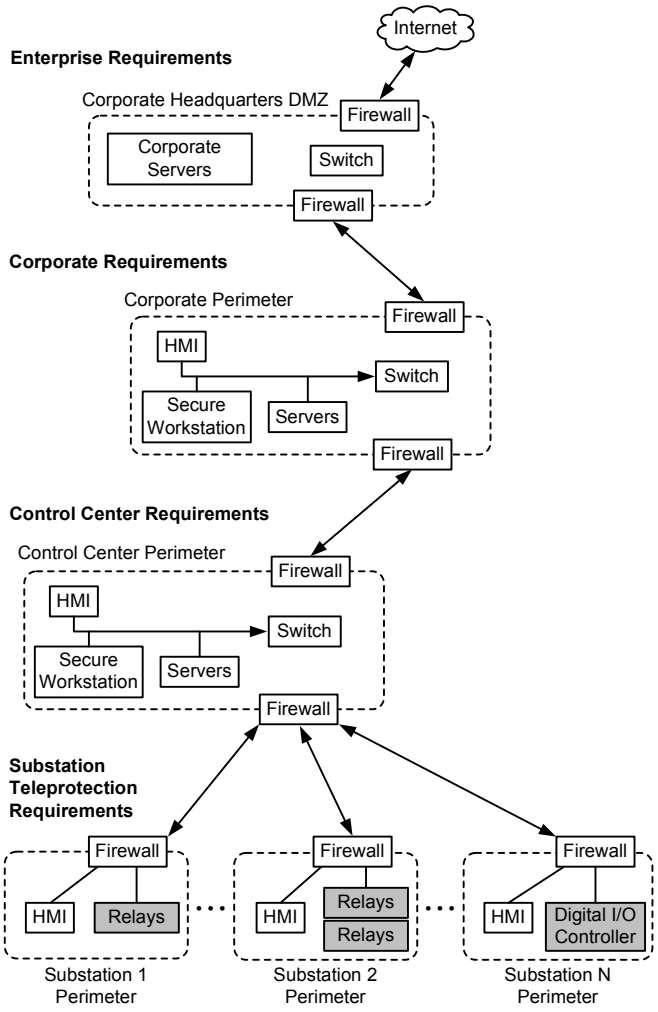

Fig. 1. Simplified Network Example

The overarching system design in Fig. 1 shows a representative, simplified layering of system operations. It is important to note that the operational requirements of the upper layers differ drastically from those of the lower layers. Unfortunately, this is often a contentious area in enterprise group discussions. One reason for this contention is that at the upper layers of the network design, security often focuses on data confidentiality, integrity, and availability. However, at the lower layers (the SCADA and substation zones), the focus changes, and operations must prioritize data availability (timing), integrity (validity of data), and confidentiality. For example, someone operating an email server in the corporate zone is generally more concerned about keeping the data confidential and less concerned about millisecond timing of the email delivery. In a substation with teleprotection, the timing for the data delivery is critical and must occur at less than power-line cycle rates (milliseconds) for the safe and reliable operation of the power system. The teams that control these zones have different concerns and priorities, which can cause many embittered battles. This also gives rise to the need for security prudence—security technical controls must not negatively impact the operational aspect of protection systems. We cannot simply drop a security technical control from one zone into another zone and expect the system to continue to operate smoothly. The security defensive controls must be engineered to the operations of each zone.

Fig. 1 uses a military term known as the demilitarized zone (DMZ), which is analogous to the strip of land, or corridor, between two untrusting entities, such as the DMZ between North Korea and South Korea. The DMZ is a tightly controlled corridor for the exchange of people, goods, and services between the two countries. In similar fashion, a company may reach out to the public Internet, identified here as the company DMZ and located at the highest level or zone. Fig. 1, for example, shows a dual-homed DMZ. It is dual-homed because the design places firewalls at two locations. One firewall is shown as a gateway to the Internet and the other as data entering into the company intranet (internal network). This design also works well at the lower levels and does not negatively impact operation at the SCADA and substation layers. In traditional DMZ designs, the firewalls are from two different manufacturers because this prevents cyberattackers from simply repeating their attack efforts at both the firewalls. Fig. 1 shows an implied DMZ arrangement in the substations holding critical cyberassets, as defined by the NERC CIP requirements.

In this design, the control system DMZ acts as a secure corridor between the SCADA network, or critical network, and the corporate network of the company. The DMZ supports our recommendation that automation systems should never use publicly routable non-RFC 1918 Internet Protocol (IP) addresses in these zones. Also, the firewalls should drop all RFC 1918 traffic attempting to egress or ingress these zones. The goal of this design is to prevent, in the event of a firewall or router failure, non-RFC 1918 packets from propagating

across and outward to the public Internet and inadvertently providing access to SCADA and automation systems due to a security device failure.

III. Firewalls

A firewall is an important security appliance whether it operates as a separate device or as the firewall that is found inside an HMI computer. Newer antivirus software often combines firewall functionality with its antivirus features. Firewalls should use a deny-all-by-default policy for protecting critical infrastructure. Unfortunately, firewalls are often not set up or not implementing rules that are correct for use in a control system environment. The firewall settings should only allow authorized traffic to flow to and from an HMI computer and a specific substation device (or devices). The firewall should reject or drop those packets not allowed.

Firewall rules are not very difficult to set up and are easily fine-tuned, but configuration and setup must be done correctly because mistakes can allow malware to communicate and propagate on a network infrastructure. Most firewalls use the following structure:

> Firewall action (accept, drop, or reject), source IP address, source port (or ports), destination IP address, destination port (or ports)

In the case of an HMI computer with IP address 192.168.1.2 that needs to conduct a DNP3 poll on a substation device with IP address 192.168.1.3, the DNP3 port is 20000. The firewall rule would be similar to the following:

> Allow 10.10.10.2, 1 – 65535, 192.168.1.3, 20000

In this example, the source port may be too wide, but part of the configuration process is to lock it down to something more restrictive. The point is to lock down the firewall rules so that only those ports, protocols, and services in operation communicate. For example, the above firewall rule was changed to the following in order to lock down the system:

> Allow TCP 10.10.10.2, 1 – 65535, 192.168.1.3, 20000

Most newer firewall versions allow both ingress and egress filtering and follow the state of the connection. Rules may need to be set up to cover both directions of data flow, both into and out of critical infrastructure. Because substations rarely start conversations with outside networks, the egress rules should be limited. In order to understand all the ports and services required for firewall rules, perform a simple network scan on a simulated nonproduction SCADA or substation power automation system. For example, a scan using software tools such as Nmap (Network Mapper), Zenmap, or Nessus® provides enough documentation to set up the access control list, or rules, for the firewall to use. We do not recommend conducting direct scanning of live power systems because incorrectly configured scanners can disrupt power system safety and reliability. Always test scans on simulated systems first. Fig. 2 depicts an example scan result showing the IP addresses and ports. These items are useful for setting up the rules for a firewall and providing a method to identify if there are any unused ports or services that can be turned off.

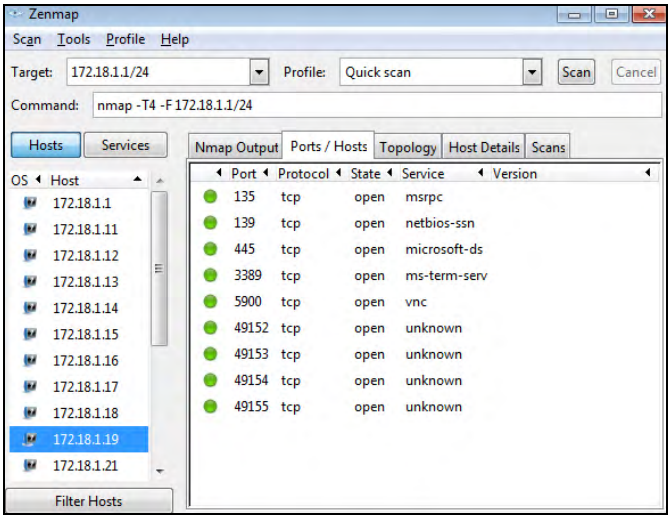

Fig. 2. Zenmap Scan Results

Scanning a system to find open ports and services is also important for NERC CIP compliance. It demonstrates to an auditor a process and document trail for strong security controls. Taking the results of the Zenmap scan and placing them into the firewall rules on the HMI computer provide a means to communicate to an observer or auditor that the power system automation designer takes security seriously.

Firewalls are a somewhat useful tool because they only allow communication to and from appropriate sources and destinations. As seen in Fig. 3, there are next generation firewalls, discussed in Section XI of this paper, that help alarm on unexpected packets. Current firewall technology is less likely to be successful in dealing with a zero-day attack; however, next generation firewalls show promise.

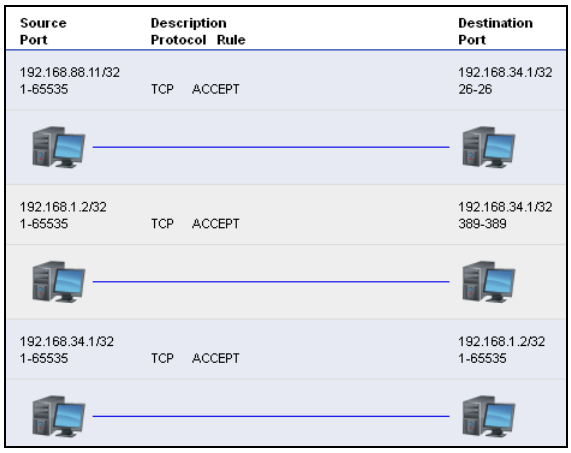

Fig. 3. Firewall Rule Example

IV. Reconnaissance and Shodan

In this section, we describe the process an attacker might follow to create a zero-day exploit of an HMI system program. The goal in sharing this process is to help clarify various security techniques for addressing zero-day attacks. These techniques are discussed later in this paper.

There are basic actions that an attacker takes to gain unauthorized access to generate a zero-day exploit. For example, the attacker must perform some form of system reconnaissance. The attacker might do this with help from social engineering. Social engineering is the act of obtaining personal or social information through illegitimate means. Even a little information, such as a name, birth date, and favorite car model, can be all it takes for the attacker to impersonate someone and gain emergency access into a system. The attacker can also go to an Internet search engine, such as www.shodanhq.com, to locate information that is publicly available about a target system or even systems that connect to a target system. System reconnaissance gives the attacker information about the manufacturer of the HMI software.

For example, in 2011, Shodan listed the IP address of a web camera (as shown in Fig. 4), which seems innocent enough. However, the web camera clearly showed the HMI screen of a wastewater treatment system. The IP address did not require any authentication to see the screen of the HMI software. On the screen, a viewer could read the IP address of the system, the name of the HMI software manufacturer, various wastewater levels, and other important information.

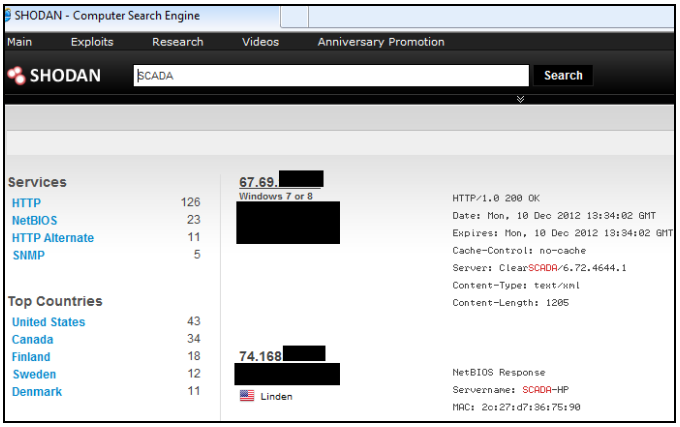

Fig. 4. Example From Shodan

At this point, it is important to address the often-heard statement: "But my system is not connected to the Internet." Keeping the system separated from the Internet is a critical part of security. However, as discussed at length in many security meetings, one of the key lessons from Stuxnet was that there was no direct connection from the Internet to the Iranian nuclear centrifuges. In the case of Stuxnet, the system was compromised by a USB flash drive or laptop computer that moved between the centrifuge network and another network with access to the Internet. As the laptop or flash drive moved from one network to the other, it communicated and transmitted malware to and from the control center network to the Internet. In this example, antivirus software would not stop or prevent the malware from executing because Stuxnet was a zero-day exploit (i.e., there were no antivirus definitions available to stop it). So, having no direct access to the Internet is not a guarantee of security.

A key lesson for control system security is to consider a use policy that does not allow devices to migrate into or out of the control system network. Other methods in conjunction with network isolation provide security of automation systems against APTs, such as Stuxnet.

An attacker can also make efforts to uncover the operating system, version, services, ports, and HMI program name. This is why it is important to label or classify information on a need-to-know basis and determine who should have access to such information. Also, it is critical to not allow any of these systems to connect directly to the Internet. In the case mentioned previously, the web camera should not be directly connected to the Internet and should be filtered or controlled.

Attackers are searching for a vulnerability that allows them to gain access into the HMI system at a privileged level. Once at the privileged level, attackers can modify and change the HMI settings or modify the display such that everything looks normal when in fact the process is out of control or nonoperational.

V. Creating a Zero-Day Exploit With Fuzzing

The goal of attackers is to discover a vulnerability that they can exploit in some way, such as by reverse engineering the software and uncovering hidden access points, passwords, and/or hidden administrative accounts that give privileged access to the HMI. To find these vulnerabilities, an attacker may fuzz the inputs to the program. Fuzzing means to send corrupted data as an input. For example, to fuzz *1234*, an attacker might enter *ABCUDEZDG12340000&&&**** instead of entering *1234*. By fuzzing the program inputs, the attacker hopes to find a failed input validation routine. For example, one typical error is when the HMI program does not validate input strings. If the input string error is not validated and the error is not handled well, it may provide a buffer overflow that could lead to unauthorized privileged access to the computer and HMI software.

In fact, one of the authors was able to use such a technique to uncover a vulnerability in an example HMI program by fuzzing the HMI program. The fuzzing technique yielded root-level menu access with no need to enter either the username or password. This allowed the author to create his own user account and password in the HMI program and access that program at will.

The intent of a software fuzzer, such as the Microsoft MiniFuzz File Fuzzer shown in Fig. 5, is to act as a software development utility for programmers to find faults in their programs prior to release. However, attackers can use the same software for their purposes.

By creating a zero-day exploit, the attacker has the means to gain unauthorized access into an HMI system. Security measures, such as current antivirus software, may not have the ability to prevent the attack.

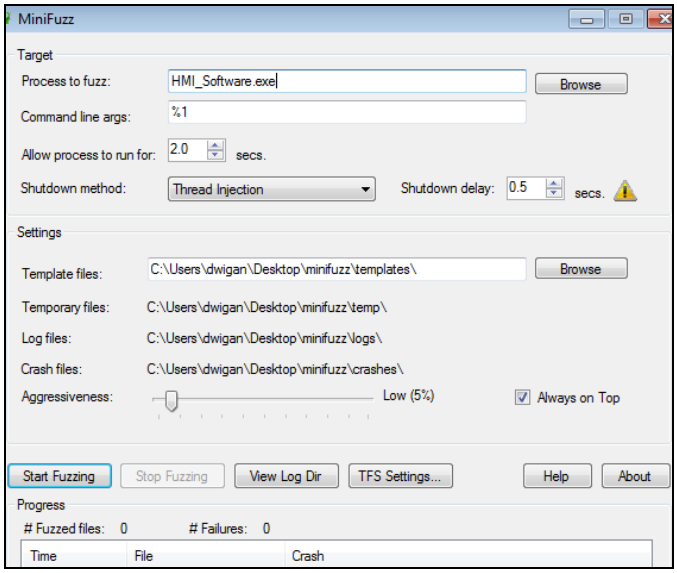

Fig. 5. MiniFuzz File Fuzzer User Interface

VI. Current Antivirus Software

The capabilities of current antivirus software are advancing. For the purpose of this paper, we distinguish between antivirus software and whitelisting as two distinct ways to accomplish security control. However, for various manufacturers of antivirus software, these boundaries are blurring. In this section, we refer to antivirus software as a program that scans and reviews the operation of other files in a computer and looks for signatures that indicate a virus is present.

Antivirus software looks at a list of known bad signatures. If it finds a program with those signatures, it takes remedial action to quarantine that program. For example, antivirus programs will flag and prevent a program from trying to rewrite or change the contents of the operating system files. This can be useful in dealing with some zero-day exploits that attempt to modify operating system files. However, it will not prevent a zero-day exploit from changing the contents of an HMI software program.

The other issue that is problematic with antivirus software is the need to update its signatures. If the system has no connection to the Internet (as we recommend), the ability of the antivirus software to secure power automation systems is inhibited.

There are solutions that address this issue, such as creating a proxy antivirus server located at the corporate zone. The duplicate proxy server connects to the Internet through a DMZ and, from there, is able to download signatures and updates to that server. Then the files are burned to a DVD and manually transferred onto the antivirus management server connected to the power system automation network. The updates are then distributed out to systems connected to the SCADA and substation networks.

This solution presents several problems for the automation system designer. There are often daily updates of antivirus signature files. It is now required to provide updates to the operating system of two servers, as well as the antivirus software. Also, this solution could create a path for a virus to transfer onto the power system automation network. There needs to be a way to ensure that no viruses transfer from the Internet-facing server to the automation system antivirus proxy server. One answer is to scan the DVD prior to transferring the signatures, but typically, a zero-day exploit cannot be caught by antivirus software until weeks or months after the zero-day exploit is first discovered. Whitelisting is explored in greater detail in the next section and affords a potential antivirus solution to zero-day exploits.

Antivirus software that exists across a power system automation network must undergo some form of system-wide management; otherwise, it must be done in a piecemeal fashion. To address this problem, most antivirus software manufacturers provide a master repository and utility to manage the policy and distribution of updates, as well as antivirus definition updates. Also, in some instances, the software scans the network for rogue computers and logs such devices.

Antivirus management software provides a means for system administration from a central location, but the downside is that it takes time to learn and requires solid knowledge of system networking, as well as operating system administration. Also, once an administrator becomes familiar with the antivirus manager of one manufacturer, it is much more likely that the administrator will not switch to another manufacturer because of the rather steep learning curve.

On the positive side, once it is learned and operational, an antivirus software manager makes administration of all the antivirus updates, signatures, and configurations on all control system computers much easier, and it becomes self-documenting. From a NERC CIP perspective, all self-documentation of the antivirus software is extremely helpful in meeting the requirements for change and configuration management.

There is some networking bandwidth overhead associated with antivirus management, but it is minimal. However, there are aspects of antivirus software that do create network and operational overhead, such as full system scans. A full computer scan for viruses is less problematic on HMI computers if there are no real-time decisions taking place. If there is any form of time-sensitive decision making taking place on the computer, then a full scan may delay the operation and therefore impact safety and reliability and should not be implemented.

The other stumbling block for SCADA systems is configuring the antivirus software. The National Institute of Standards and Technology (NIST) and Sandia National

Laboratories published findings that warrant careful antivirus configuration for application in a power system automation system [3]. At issue is manual or on-demand scanning that also affects system central processing unit (CPU) availability. Whereas on-demand processing or active scanning has little effect, antivirus signature updates tend to spike CPU activity to 100 percent temporarily, thereby having an effect on control system processes and requiring careful planning and management [3].

VII. Application Whitelisting

One rather interesting approach to battling malware is application whitelisting. "Essentially, whitelisting flips the antivirus model from a 'default allow' to a 'default deny' for all executable files" [4]. This is a welcome paradigm shift because enumerating goodness is preferred over trying to enumerate badness [5]. In the past, the number of good programs on the Internet outnumbered the bad programs. The trend has changed over the last two or three years so that the bad programs are starting to outnumber the good programs.

In this section, we explain what application whitelisting is and show how whitelisting satisfies multiple NERC CIP requirements, such as CIP-003-1 R6, CIP-007, and CIP-008. We end by addressing the question of whether application whitelisting can protect against APTs.

A. Definition of Application Whitelisting

Application whitelisting is simply software with low-level hooks into the operating system to intercept all execution attempts and determine if a program is on a list of known good file hashes (the whitelist). If the program is on the whitelist, it will be allowed to execute. Nothing else is allowed to execute. Everything is logged—both successful and unsuccessful execution attempts.

Although this idea has been around for a while, it has been gaining traction in recent years and improving greatly, with more manufacturers offering application whitelisting security software, as shown in the following list:

- McAfee® Application Control, Change Control, and Integrity Control
- CoreTrace® Bouncer®
- Lumension® Application Control
- Faronics Anti-Executable
- Microsoft AppLocker®
- Savant™ Protection

B. How Application Whitelisting Can Satisfy Multiple NERC CIP Requirements

There are multiple whitelisting features that can be applied as security controls to address the NERC CIP requirements. For example, CIP-003-1 R6 requires the documentation of a process of change control and management for any modifications to critical cyberassets. Only whitelisting administrators, trusted users, or trusted applications can change the whitelist, so application whitelisting only allows authorized changes to critical cyberassets.

CIP-007 includes the need to prevent malware, as well as the requirement to document the implementation of security patches. Application whitelisting is a compensating control for security updates. It allows a user to deploy patches after fully testing them instead of rushing patch deployment, which can impact availability. The requirement to use antivirus and other anti-malware tools to prevent malware on all cyberassets within the electronic security perimeter (ESP) is met by application whitelisting because it prevents malware more effectively than antivirus software does. Also, application whitelisting offers sophisticated anti-attack protection against memory exploits.

CIP-008 requires keeping documentation of cybersecurity incidents for a minimum of three calendar years, which is satisfied because application whitelisting logs events, provides an identification tag of which users made unauthorized attempts, and allows for easy reporting. Documented investigation of all suspicious reports is satisfied by the application whitelisting log never expiring and being set to not automatically erase; logs may also be protected on the source system.

C. APTs and Application Whitelisting

To consider how whitelisting helps to protect against APTs, we first examine why APTs are difficult to protect against. First, the *advanced* in APT means we must assume that adversaries can leverage whatever intrusion techniques they wish to break into a computer system. This includes everything from the most common publicly shared exploits against publicly disclosed vulnerabilities to privately researched custom exploits developed with the specific target in mind. Second, the *persistent* means the attackers do this as their day job. They are employed to do this and to get positive results. They are fired if they are not successful. This is a highly tuned operation administered with project managers and is like an army, with a chain of command and consequences for failure. The attackers can use whatever means and interaction level to accomplish their mission; they are not limited to only attacks involving code execution.

> *Threat* means the adversary is not a piece of mindless code. The opposition is a threat because it is organized and funded and motivated. Some people speak of multiple "groups" consisting of dedicated "crews" with various missions. [6]

Given adversaries this powerful using whatever means necessary to achieve their goals, it is difficult to imagine any one technology as a silver bullet for protecting against such attacks. However, application whitelisting can make it more difficult for an adversary to break into the computer and can provide incident handlers a tool to detect such attacks. Application whitelisting, while not perfect, provides a level of visibility into files being introduced into the environment, which can be helpful for incident handlers. By using the technology, an organization can significantly reduce the risk from current malware. Application whitelisting is the most

effective way to reduce malware impact on modern systems because whitelisting prevents any new code execution and includes memory protection techniques. The low memory overhead of application whitelisting allows it to be used for critical cyberassets that have performance requirements that keep antivirus software from being installed because of the load antivirus scans and updates can put on the system. Whitelisting scans are quick and imperceptible.

Moving from the allow-all-programs-to-execute default to a deny-all default is a great step in a secure direction. On older legacy systems, application whitelisting can act as a compensating control for patch management (e.g., if no patches are available for Windows XP Professional with Service Pack 2 because the operating system has reached its end of life) and to extend the life of out-of-support systems by providing a malware protection compensating measure.

The leading applications for whitelisting also provide the following:

- Memory protection features to protect against memory attacks, such as buffer overflow attacks, and tamper-proofing protection to prevent deletion, renaming, and overwriting of authorized code. Attacks are blocked and reported.
- Automatic updaters that can be specified as accounts, network shares, digitally signed code, or trusted updaters. Certain types of HMI and protocol conversion software with license managers require the license managers to change the underlying code to thwart pirating. Without the automatic updater mechanism, the software would not run in a whitelisting environment.
- Write protection to authorize writes only to the operating system, application configuration, and log files. All others are denied.
- Read protection to authorize reads only for specified files, directories, volumes, and scripts. All others are denied.
- The ability to maintain the whitelist from a central management tool. Central management is a useful time-saving feature. A good question to ask manufacturers is whether central management and reporting are supported by their product.

VIII. Disk Encryption

Disk encryption, as the name implies, encrypts all the data on a computer hard drive. Its purpose is to secure information if the computer is lost or stolen. It is also useful to prevent unauthorized access to the information on the computer by employees that do not have a reason to access that information. Using disk encryption should be part of a policy for the use of laptop computers.

Disk encryption is easy to implement but can present problems when troubleshooting application issues. For example, in one case, Windows was terminated because the computer lost power. Upon restart, the computer required the encryption keys to be entered. It is extremely important to print out those keys when asked to do so and store them in a secure location.

With Windows, disk encryption is accomplished via BitLocker® Drive Encryption. BitLocker is able to encrypt the entire operating system hard drive and will even encrypt new files added to the computer at a later point in time. On the operating system hard drive, BitLocker watches for conditions such as changes to the basic input/output system (BIOS) or startup files. If they undergo changes, BitLocker locks the operating system drive and requires the user to enter a special BitLocker recovery key to unlock it. It is critical for the administrator of the HMI system that uses BitLocker encryption to create the recovery key the first time BitLocker is enabled; otherwise, it is possible to accidently and permanently lose access to the drive and all files stored on it.

BitLocker encrypts files only while they are on the encrypted drive. When a file is transferred to another drive, BitLocker decrypts the file prior to the transfer. If the other drive is not encrypted, the file remains unencrypted while it is stored on the unencrypted drive. If files are shared with other users and/or systems, such as through a network, the files remain encrypted only while on encrypted drives. Even while encrypted, the files can be accessed normally by authorized users accessing the shared portion of the drive.

Disk encryption does not provide direct security that works against a zero-day exploit or other APT. The main focus of disk encryption is on lost or stolen drives and preventing unauthorized access to those lost files. Also, disk encryption is not available on all versions of Windows, and the user has the ability to turn it off and on with appropriate administrative permissions.

However, if a computer is stolen from a substation, an encrypted disk helps prevent access to important information located on the hard drive.

Disk encryption has almost no overhead impact—the small amount of impact is dependent on the CPU speed, with typical performance impact from BitLocker ranging from 5 to 8 percent. The initial data encryption on the drive operates at speeds of 500 MB per minute. It can take over 3 hours to encrypt an entire 30 GB drive, but this process can run in the background and occurs only once at setup.

It is also important to have a means to secure data in transit. In order to encrypt data in transit, consider the use of programs and/or devices such as Internet Protocol Security (IPsec), BitLocker To Go, and IronKey™.

NERC CIP provides some support for disk encryption requirements, such as NERC CIP-003 regarding access and handling of sensitive data stored on a hard drive. In addition, one NERC CIP guideline suggests disk encryption as a means to protect data [7].

IX. Microsoft EMET

Microsoft provides an Enhanced Mitigation Experience Toolkit (EMET), which is a free security solution that adds a layer of security and creates more depth for a Microsoft-based operating system platform. Placing EMET into operation on an HMI computer helps prevent exploitation of vulnerabilities.

It is important to note that Microsoft does not guarantee EMET will mitigate exploitation of all zero-day vulnerabilities, but it adds another layer of security that an attacker must compromise to gain access into the HMI computer. According to Microsoft, the EMET security mitigation technologies "work to make exploitation as difficult to perform as possible. In many instances, a fully-functional exploit that can bypass EMET may never be developed" [8].

It is important to note that EMET protects third-party (non-Microsoft) software in addition to Microsoft software, but there may be incompatibility issues between third-party software, such as HMI software, and EMET. Therefore, it requires full testing prior to deployment on a power automation system network.

Once EMET is installed, the user must configure it to protect the HMI software. This is a manual process. In fact, all programs in operation on the computer must be added to the EMET protection. A benefit of the laborious configuration process is that EMET gives greater visibility to the security and alarm notifications of an event. EMET is configured and applied on a per-program basis. Alarm events occur on a per-program basis as well.

EMET provides protection for Structured Exception Handling Overwrite Protection (SEHOP), as shown in Fig. 6. SEHOP is a common area in computer memory that an attacker may try to exploit in Windows platforms. It also protects against buffer overflow attacks. A buffer overflow attack using exception handling is when an attacker overwrites and creates execution of an exception record on the Windows stack, as illustrated in Fig. 7. Once an exception happens, the attacker executes code on the handler that makes the operating system jump to wherever the attacker wants, such as to a location that points to the program code of the attacker. EMET validates the exception record chain before the operating system calls any exception handlers. If the chain is corrupted, EMET terminates the process without calling any of the handlers. In the case of a buffer overflow attack, the code from the attacker would not execute.

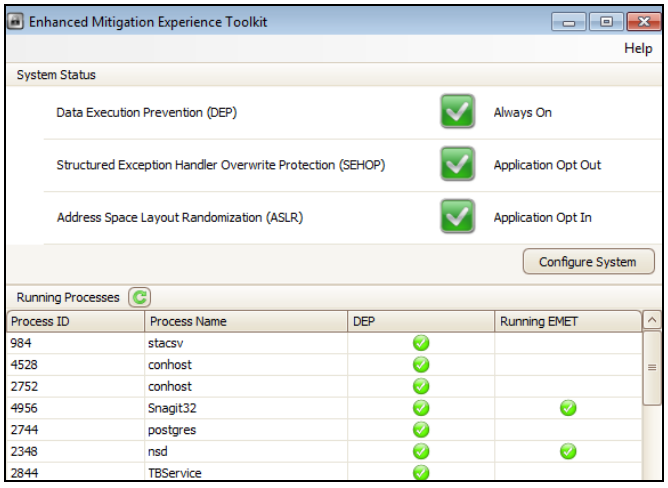

Fig. 6. EMET Configuration Menu

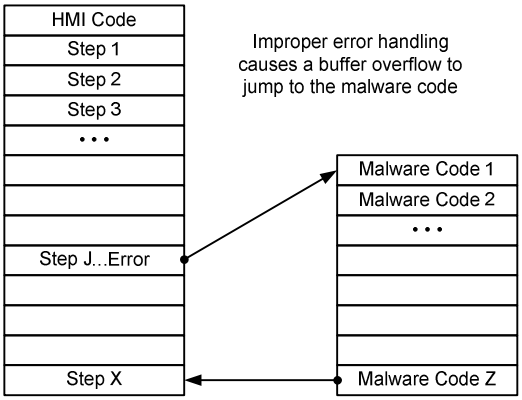

Fig. 7. Simplified Example of a Buffer Overflow Attack

The Windows XP operating system provides a feature known as dynamic Data Execution Prevention (DEP); however, many software manufacturers do not employ it. EMET allows for the implementation of DEP even if the software does not employ its use. DEP is very useful against zero-day attacks. Just like the previously mentioned SEHOP, DEP prevents an attacker from exploiting a vulnerability. For example, it prevents jumping to a memory location where the code of the attacker resides, such as in the heap or stack. DEP marks the stack and heap as nonexecutable, and any attempt to execute malicious code from these regions is thwarted at the processor level.

To increase the odds of success, most attacker exploits use a heap spray technique. This technique places copies of an attacker shellcode in as many memory locations as possible, spraying their malicious code into the computer. EMET attempts to mitigate this technique through DEP.

The last area of focus for EMET is mandatory address space layout randomization (ASLR), which randomizes the addresses where modules are loaded. This helps prevent an attacker from leveraging data placed into predictable locations. EMET prevents attacks by not using predictable mapping for dynamic link libraries (DLLs)—it randomizes locations. EMET forces modules to be loaded at randomized addresses for a target process, and this prevents exploits that rely on predictable mapping. Since Version 2.1, EMET has included a Bottom Up Randomization option, which further randomizes addresses where modules are loaded [9].

EMET supports the following operating systems and service pack levels:

- Windows XP Service Pack 3 and above.
- Windows Vista® Service Pack 1 and above.
- All service packs for Windows 7, as well as Server Operation Systems.
- Windows Server 2003 Service Pack 1 and above.
- All service packs for Windows Server 2008 and Windows Server 2008 R2.

Since EMET Version 3, full group policy support and logging are included. Even though EMET is free, it is worth noting that the tool is fully supported by Microsoft.

EMET is a free and underutilized security control to help mitigate against zero-day exploits and APTs. Its applicability to NERC CIP is not easily discerned because it is not antivirus software or encryption. EMET is, however, a strong security control that helps lock down a Windows-based HMI located in a substation.

X. Virtual Private Networks (VPNs)

Getting data to securely traverse untrusted or even trusted networks is not an impossible task. There is a standard protocol (IPsec) for such a task. The IPsec protocol allows the automation system designer to create a secure VPN either between two computers or between two electronic access points located on ESPs. The protocol works with multiple operating systems, firewalls, and routers.

The protocol encapsulates a payload with strong security algorithms, including encryption and authentication. There are multiple choices to make in setting up a secure IPsec VPN, and like the configuration of firewalls, this can be problematic. However, newer systems make the configuration of the IPsec tunnel much easier than in the past. Configuration of the IPsec VPN depends on the network topology and use model. For more traditional VPN use, the tunnel mode creates a secure tunnel that connects between two firewalls. For example, this is an excellent way to secure communications from the NERC CIP ESP guarding critical cyberassets located in a substation to the SCADA center ESP. The IPsec protocol also allows for securing the communications link between two peer devices, such as two computers (e.g., from a computer located inside the substation to a computer located in the energy management center). This secure channel ensures that communication between the two gateways, or computers, remains free from tampering as well as private.

One consideration is that because IPsec encrypts the data packets, any intrusion detection system that is conducting deep packet inspection as part of its trigger mechanisms for intrusion is no longer going to operate as expected.

The benefits of IPsec for securing the transport of data are excellent for data confidentiality and integrity, but zero-day exploits or APTs are not mitigated by an IPsec tunnel. In some cases, the tunnel may confuse the operation of the network intrusion detection system, as explained in Section IX. However, from a NERC CIP perspective, use of IPsec communicates to an auditor that defense-in-depth strategies are used with multiple layers of security and that security is taken seriously [10].

XI. Future Trends

One trend that seems to be a promising way to secure communications within an ESP is the next generation firewall. The next generation firewall blocks unwanted applications and inspects the allowed applications for threats. This type of firewall continually classifies all traffic across all ports along with user information, allowing the firewall to tie the application and user to a location-independent policy. This protects traffic against exploitation.

Next generation firewalls are currently undergoing deployment into traditional data centers and may have a very important security role to play in the substation automation ESP. There is not much information about the impact on local-area network determinism, resiliency, and robustness when this technology is employed for teleprotection of power automation systems.

XII. Continual Learning

Continuing security education is an important activity and should be part of the plan of a power automation designer for increasing the in-depth defense of a system. Ultimately, system security comes down to the security mindset of the people that come into contact with the automation systems that guard and protect the power system. Continuing security education is also part of the CIP-004 requirements.

There are several places to learn about cybersecurity with online resources, such as the International Information Systems Security Certification Consortium, Inc. ((ISC)$^{2®}$), the SANS™ Institute, and the Information Systems Security Association (ISSA). Each organization provides many online courses or webinars, but in the opinion of the authors, a more useful resource is to join and actively participate in a local chapter of a security organization, such as those just mentioned or the U.S. Federal Bureau of Investigation (FBI) InfraGard® program. Often, chapter meetings occur on a monthly basis. Both authors of this paper participate in local ISSA and (ISC)^2 chapter meetings. Attendees at these meetings represent multiple industry segments, including law enforcement and universities. The participants in the meetings provide short talks about what currently works well for them, even getting participation from security solution manufacturers. The ability to meet with people and discuss current trends and issues in a one-on-one forum is invaluable for keeping up with the newest threats and vulnerabilities, as well as potential or future solutions. InfraGard also offers networking and collaboration among similar industry players. We strongly encourage seeking out and meeting regularly with other professionals as a means to improve overall security.

XIII. Conclusion

This paper shares a variety of technical security solutions that enable power system automation designers to keep power safe, reliable, economical, and secure. The paper describes the means that enhance the protection of automation systems, making their operation increasingly robust. In many cases, these technical controls surpass current regulatory requirements, such as those found in NERC CIP.

Firewalls and application whitelisting that use a deny-all default rather than an allow-all default are part of the many new technical means to help mitigate zero-day exploits or even APTs, such as Stuxnet.

XIV. References

[1] R. Langner, *Robust Control System Networks, How to Achieve Reliable Control After Stuxnet*. Momentum Press®, LLC, New York, 2012.

[2] P. Oman, E. O. Schweitzer, III, and D. Frincke, "Concerns About Intrusions Into Remotely Accessible Substation Controllers and SCADA Systems," proceedings of the 55th Annual Georgia Tech Protective Relaying Conference, Atlanta, GA, May 2001.

[3] J. Falco, S. Hurd, and D. Teumim, "Using Host-Based Antivirus Software on Industrial Control Systems: Integration Guidance and a Test Methodology for Assessing Performance Impacts," NIST Special Publication 1058, September 2006. Available: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=823596.

[4] J. Beechey, "Application Whitelisting: Panacea or Propaganda?" December 2010. Available: http://www.sans.org/reading_room/whitepapers/application/application-whitelisting-panacea-propaganda_33599.

[5] M. Ranum, "The Six Dumbest Ideas in Computer Security," September 2005. Available: http://www.ranum.com/security/computer_security/editorials/dumb/.

[6] R. Bejtlich, "Understanding the Advanced Persistent Threat," *Information Security*, July 2010. Available: http://searchsecurity.techtarget.com/magazinePrintFriendly/0,296905,sid14_gci1516312,00.html.

[7] NERC, "Security Guideline for the Electricity Sector: Protecting Potentially Sensitive Information," September 2011. Available: http://www.nerc.com/docs/cip/sgwg/Protecting%20Sensitive%20Information%20Guideline%20Draft%20Revision%208-30-11%20v04.pdf.

[8] Microsoft Support, "The Enhanced Mitigation Experience Toolkit," October 2012. Available: http://support.microsoft.com/kb/2458544.

[9] D. Stevens, "Microsoft's Enhanced Mitigation Experience Toolkit," *(In)Secure*, Issue 30, June 2011. Available: http://www.insecuremag.com.

[10] IEEE Standard 1402-2000, IEEE Guide for Electric Power Substation Physical and Electronic Security.

XV. Biographies

Dwight Anderson received his BS in electrical engineering from Steven's Institute of Technology. He is an engineer for Schweitzer Engineering Laboratories, Inc. (SEL) in SEL Engineering Services in Pullman, Washington. Prior to joining SEL in 2005, he worked 20 years for Hewlett-Packard/Agilent Technologies as an aerospace and defense business development manager and systems engineer, working on projects ranging from electronic warfare countermeasures to SCADA system programming. Dwight holds a professional engineering license from the state of Texas and a Global Security Essentials Certification (GSEC) from Global Information Assurance Certification (GIAC). He is also a Certified Information Systems Security Professional (CISSP).

Huba Leidenfrost grew up in Liberia, West Africa, and received his BS in computer science from the University of Idaho. He is a lead application engineer for Schweitzer Engineering Laboratories, Inc. (SEL). Prior to joining SEL in 2007, he worked 15 years for the University of Idaho as a system administrator, network analyst, and network security analyst, where his projects ranged from managing central UNIX servers and Internet security services to investigating computer security incidents. Huba has served as an expert witness in federal and civil court cases involving computer crime. He is a Certified Information Systems Security Professional (CISSP) and a founding board member of the Palouse Information Systems Security Association.