INL/EXT-05-00109

Improving Control System Security Through the Evaluation of Current Trends in Computer Security Research

Bri Rolston

March 2005

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-05-00109

Improving Control System Security Through the Evaluation of Current Trends in Computer Security Research

Bri Rolston

March 2005

Idaho National Laboratory
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Homeland Security
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

ABSTRACT

At present, control system security efforts are primarily technical and reactive in nature. What has been overlooked is the need for proactive efforts, focused on the IT security research community from which new threats might emerge. Evaluating cutting edge IT security research and how it is evolving can provide defenders with valuable information regarding what new threats and tools they can anticipate in the future.

Only known attack methodologies can be blocked, and there is a gap between what is known to the general security community and what is being done by cutting edge researchers—both those trying to protect systems and those trying to compromise them. The best security researchers communicate with others in their field; they know what cutting edge research is being done; what software can be penetrated via this research; and what new attack techniques and methodologies are being circulated in the black hat community.

Standardization of control system applications, operating systems, and networking protocols is occurring at a rapid rate, following a path similar to the standardization of modern IT networks. Many attack methodologies used on IT systems can be ported over to the control system environment with little difficulty. It is extremely important to take advantage of the lag time between new research, its use on traditional IT networks, and the time it takes to port the research over for use on a control system network.

Analyzing nascent trends in IT security and determining their applicability to control system networks provides significant information regarding defense mechanisms needed to secure critical infrastructure more effectively. This work provides the critical infrastructure community with a better understanding of how new attacks might be launched, what layers of defense will be needed to deter them, how the attacks could be detected, and how their impact could be limited.

CONTENTS

ACRONYMS

cDc	cult of the Dead cows
CERT	Computer Emergency Response Team
COTS	commercial-off-the-shelf
CS	control system
DDOS	distributed denial of service
IDS	intrusion detection system
IOS	internetworking operating system
IRC	internet relay chat
OS	operating system
OSVDB	Open-Source Vulnerability Database
POC	proof of concept

Improving Control System Security through the Evaluation of Current Trends in Computer Security Research

1. INTRODUCTION

At present, control system security efforts address two predominant issues: a) how to secure older, proprietary networks, and b) how to secure new systems, which incorporate Commercial-Off-the-Shelf (COTS) software and standardized networking protocols, from known threats. The focus of this work, however, has been primarily technical and reactive in nature, i.e., adding new signatures to the anti-virus libraries to detect new worms released. What has been overlooked up to this point is the need for more proactive efforts, focused on the direction from which new threats might emerge.

By tracking the work currently underway in the IT security research community and how it affects control system security, computer-based threats to control system networks can—to some extent—be anticipated. Evaluating cutting edge IT security research and how it is evolving can provide defenders with valuable information regarding what new threats and tools they can anticipate in the future.

2. WHY TRENDS IN IT SECURITY RESEARCH ARE IMPORTANT FOR CONTROL SYSTEM SECURITY

IT Security research is always ahead of the defensive curve because defensive actions are, by definition, reactionary. People hoping to secure a computer network, including control system environments, can only protect the network against attacks that can be identified, have patches or fixes available for the software, have workarounds for the holes that still provide for functionality, or can be blocked by perimeter defenses. In short, only known attack methodologies can be blocked, leaving a gap between what is known to the general security community and what is being done by cutting edge researchers.

The best security researchers, however, are communicating with others in their field and are influenced by the work and research being done by their peers. They know what cutting edge research is being done; what software can be penetrated via this research; what new attack techniques and methodologies, also known as 0-day exploits, are being circulated in the black hat community; and how defenders can begin to protect their networks from such attacks. Security professionals who are not members of this very small, close-knit research group do not have access to such information and are dependent upon the 0-day exploit becoming well-known, allowing vendors and development groups to gather enough information about the exploit technique to fix the vulnerability.

Information sharing among the very best hackers and researchers in the IT security field is based upon the relationships the people have with each other. While a hacker may often associate with other black hat individuals and groups, he or she is just as likely to associate with white hat individuals or groups, too, because of the nature of the community. This allows outsiders to track the flow of information from the inner circle of the elite outward to the general IT security community, much like watching the ripples in a pond flow outward from a disturbance in the water. This ripple-like flow of data can be used to determine which information is valuable, where it is in its life cycle, and who may have access to more privileged data of the same type.

Note: *This type of analysis requires a high level of technical expertise on the part of the analyst, familiarity with the IT security research community, and a focus on the research being done—not the researcher.*

3. BACKGROUND INFORMATION

A number of points must be established in order to ensure analysts are working from the same frame of analytical reference with regard to this project. The following are a list of general assumptions regarding control system networks and their particular security configurations.

1. No computer network can be completely secured. A determined, skilled attacker can find a way into a system given enough time.
2. A control system (CS) environment includes both the control system network and the business network. The control system network is comprised of control system specific hardware, software, and network protocols that actually manage the data, measurements, and control responses of the equipment. The business network is the general IT network to which the CS network is connected. The connection provides a way for data from the CS network to provide quality assurance, safety information, or other data generated by the CS applications and used by the company or organization to manage the business aspects of CS production.
3. The CS network is at higher risk for attack because the CS networks are frequently connected to the business network for data exchange purposes and indirectly connected to the public Internet via the business network. Previously, network connectivity on control systems was limited to hard-wired connections from the control system element to the communications device and transmitted over a variety of telecommunications network architectures. Today, the communications are transmitted over a much wider range of devices and mediums.
4. Control system vendors are moving toward more standardized networking protocols like TCP/IP and DNP3, as well as standardized operating systems (OS) like Linux and Windows. As with traditional IT networks 20 years ago, the standardization of the OS and network communication protocols results in increased efficiency, lower cost of ownership, and a greater risk for attack. Additionally, the types of attacks to which the CS networks are vulnerable are standardizing as well. For example, if a vendor builds his data historian application for the Windows 2003 Server platform, the data historian requires the same level of patching as do those Microsoft hosts on a traditional IT network because the operating system and its vulnerabilities are well-known.
5. Defenders cannot anticipate who or why people will compromise a network because there are simply too many potential suspects and motives for the attack. Many of the less skillful attacks can be deflected through a layered approach to security. A skilled attacker, however, will not be deterred by tight security and is familiar with techniques, tools, and attack methodologies to defeat such measures.
6. Due to the very nature of computer networks and vulnerability research, security efforts will always lag behind the development of new attacks. Computer security is primarily reactive, not proactive, meaning there will always be unknown, new attacks that can potentially compromise a network, both business and CS, that cannot be detected or blocked by defenders.
7. In this report, the computer security research community does NOT refer to the whole of the IT security community, i.e., IT security vendors, research laboratories, etc.; rather, the term applies to those people who are researching new ideas for securing computers through the use of black hat security techniques. Some members of the computer security research community may be members

of the IT security profession; however, many of them may work in other career fields and do computer security research as a hobby.

4. HOW IT SECURITY RESEARCH MUST BE ASSESSED

Not all of the traditional IT attacks will work in a control system environment. Nor have all the attacks that would work on a CS network been ported over for the purpose of disabling critical infrastructure. To identify the security research that must be considered for potential use in a computer-based attack against a control system, defenders must perform two tasks.

First, they must evaluate current research to determine if the techniques or tools could be used to successfully attack a control system network. If the information could be used to attack a CS network successfully, then the defender must evaluate what strategies, tools, and policies are available to deflect, mitigate, prevent, or identify the attack. Second, if the information could be used to successfully attack a control system and there are few—if any—defense measures available to the IT security world in general, then the research must be carefully evaluated to identify who is developing the information and what the trends in that particular arena are.

There are several key issues that must be considered in the analytic process. They include:

1. The analyst must focus on the significance or "what" of the research, NOT who is developing the field of study. While the credibility of the researcher must be considered to determine the quality of the research, the researcher himself is not relevant to the analytical process.
2. Announcements and discussions of new vulnerabilities, attacks, exploit code, and ideas must be followed to discern who is doing what and how in the security community.
3. The analyst must understand the culture of the computer security research community, both the researchers and the hackers, and how the tight-knit, exclusive, collaborative nature of the field affects vulnerability discovery and exploit code development life cycles.
4. The life cycle of vulnerability discovery and exploit development, including the symbiotic development process, dependency upon the personal collaborations of the researchers, etc., must be clearly understood in order to assess the risk that the exploit poses to a control system network.

5. THE "WHO" OF COMPUTER SECURITY RESEARCH

5.1 What Is a Security Researcher

A security researcher is someone who investigates new problems and vulnerabilities in computer security and uses their research to help improve security awareness or defenses as a whole. Very good researchers are curious individuals with an outstanding understanding of operating systems, networking and network protocols, and application development. They typically specialize in one or two areas of expertise, know how software should work, evaluate the specifications of protocols and design specifications from unusual perspectives, are very creative, and are excellent problem solvers.

5.2 Black Hats vs. White Hats

The term "black hats" refers to a type of computer security researcher who attacks networks and computers using previously unknown vulnerabilities and exploit tools with malicious intent. In order to successfully exploit systems, a black hat needs to find new ways to break into computer systems and

networks, which entails unearthing new vulnerabilities and writing exploit code that makes use of the vulnerabilities.

The term "white hats" refers to a type of computer security researcher who assesses and protects systems and networks from attack by using black hat tactics. White hats also unearth new vulnerabilities and write exploit code that makes use of the vulnerabilities. For example, a white hat may perform penetration testing or vulnerability assessments with black hat tools on a network to identify previously unknown vulnerabilities and to recommend a method for remediation. Or, the white hat may work for an IT security research firm, evaluating products for unknown vulnerabilities, writing proof of concept (POC) code, and working with vendors to resolve the issues. The very best white hats frequently exchange information on new research and techniques with black hats, although they do not attack systems with malicious intent.

Note: *In this paper, a black hat will also be referred to as a "hacker." Although hackers can be either black or white hat, the term "researcher" will be used to identify white hat hackers who perform security research without the malicious intent to attack networks in order to avoid the negative connotations associated with the term "hacker."*

5.3 What a Hacker or Researcher Is Not

For the purpose of this project, a good hacker or researcher is NOT any of the following:

1. System administrators who use but do not develop their own security tools or discover new vulnerabilities;
2. Script kiddies—unskilled attackers who do NOT have the ability to discover new vulnerabilities or write exploit code and who are dependent on the research and tools of others;
3. Worm and virus writers—attackers who write the propagation code used to spread mobile malware such as worms, viruses, and Trojans, but who do not write the exploit code used to penetrate the systems infected; and,
4. Web defacers—attackers, typically script kiddies, who specialize in the defacement of web pages.

5.4 Types of Hackers and Researchers

The two primary types of hackers and researchers whose work must be considered when determining what IT security research is applicable in a control system environment are bug hunters and exploit coders.

Bug hunters actually search through OS, application, network protocol technical specifications, etc. for errors or faults in the code which would allow an attacker to escalate privileges or gain unauthorized access to system resources. Once a likely issue has been discovered, through techniques such as fuzzing and reverse engineering, the bug hunters develop the exploit idea and write rough tools used to demonstrate proof of concept (POC). POC tools are often rough drafts used to develop more sophisticated tools; they often only work on a few test hosts and are not ready to be used for mass exploitation. POC tools are 0-day exploits and are often given to exploit coders in return for industrial strength exploit tools.

Exploit coders find writing industrial strength exploit code more interesting. They take the rough POC tools and refine them so they work on an entire version set of the vulnerable software. For instance,

the exploit coder may exchange an exploit for a new POC tool from a bug hunter that works on only a few, specifically configured Windows 2000 hosts. After examining the POC code and the OS flaw, the exploit coder refines the code so it works reliably on Windows NT 4.0, 2000, XP, and 2003 all of the time. At this point, the exploit is still relatively unknown and can be exchanged by the exploit coder for other POC tools or other industrial strength exploits.

5.5 What Motivates Hackers and Researchers

Hackers and researchers are generally driven to research by one of three motivations: curiosity, money, or strong personal beliefs. Some researchers and hackers research computer security issues as a hobby. Their curiosity and "just to see if they can" attitude drives them to explore applications, operating systems, and networks in ways not typically considered by developers. In general, the unconventional approach they take to investigating the software allows them to identify weak sections in code and ascertain how to exploit those areas in ways not imagined by the people writing the software. Hackers and researchers who perform research for curiosity's sake often publish their tools and findings in restricted circles to share their work and gain a reputation for being very good at what they do.

Other hackers and researchers are paid to perform the research. Hackers for hire are paid to write tools for unknown holes or paid to break into networks. Hackers for hire do not generally publish advisories or tools. Rather, they accumulate tools and share research with a very closely monitored, tight circle of associates who have tools and research to exchange as well. This is how they increase their toolkits, improve their ability to break into varied networks and systems for their customers, and diversify their own skill level by remaining abreast of what is cutting edge research in the field. Examples of professional hackers would include state-funded information warfare or operations teams, as well as groups such as the Source Code Club, a group who has purportedly offered portions of the Cisco Internetworking Operating System (IOS) and Napster source code for sale via various Internet Relay Chat (IRC) channels.

Professional researchers are often paid to do penetration testing or vulnerability assessments, as well as to write code specifically designed to detect vulnerabilities not previously identified by their customers. Researchers in the security field MUST produce tools or advisories of new vulnerabilities they have discovered in much the same way university professors publish research papers. This is an important aspect of a professional researcher's career, which helps establish his or her credibility and brings in more clients. Other hackers and researchers are more likely to exchange ideas and tools with someone who has demonstrated ability to generate new ideas and produce solid code. Examples of professional researchers include Simple Nomad, the former L0pht Heavy Industries, Dave Aitel, and others.

Activism through hacking, or hacktivism, is another driver for hackers. Generally, professional researchers do not indulge in the hacktivism attacks because they have a great deal to lose if they are caught running black hat attacks. But, several very good white hat researchers and many other black hats have provided tools and run attacks in the name of patriotism, human rights, etc. As with any other attacker, this type of motivation makes a computer-based attack in the name of a cause much more difficult to anticipate and deter. Examples of an attack performed for hacktivist reasons include the Distributed Denial of Service (DDOS) attack on Mexican President Ernesto Zedillo's website in 1998 by the Electronic Disturbance Theater, the group credited with organizing the DDOS, as a show of solidarity with the Zapatistas. One of the best known hacktivist tools was released by cult of the Dead cows (cDc) and is a web browser called Peekabooty, which allows people whose access to the Internet is tightly controlled by the government—such as China and Iran—to bypass standard firewalls and restrictions.

5.6 Nature of the Computer Security Research Community

The computer research community, i.e., good security researchers and hackers, as opposed to the entire community of computer security professionals and academic experts, is very tightly knit, exclusive, and suspicious of newcomers. To successfully gain entrance into the circle, a person must demonstrate a very high level of knowledge about computer security, contribute to the atmosphere of constant learning, and be prepared to share ideas and tools in exchange for those of others.

Due to the nature of the community, hackers and researchers, regardless of the color of their hats, interact frequently to share ideas, research, and tools. And, because people tend to specialize in only one or two areas of expertise, hackers and researchers depend on this swap of information to become better at what they do and to enhance their abilities to assess and secure or to assess and attack networks. This bartering of research allows the researchers to develop the weaker areas of their tool kits, as well as providing them with an audience with whom they can further develop or create their own ideas.

6. HOW EXPLOITS ARE DEVELOPED

6.1 Why Understanding Exploit Life Cycle Is Important

In order to truly understand the significance of new research and tools, an analyst must know where the exploit or idea is in its development stage. The more rapidly vital ideas can be identified, the better an analyst can evaluate the research for its implication and applicability against control system networks.

To comprehend how exploits are built, an analyst must know how researchers and hackers develop and share new vulnerabilities, POC code, and industrial strength exploits. This includes knowing the key areas of computer security expertise, how vulnerability discovery works, how exploits are refined, and how the data is shared among researchers and hackers.

6.2 Eleven Areas of Computer Security Research Expertise

The eleven primary areas of computer security research expertise are listed below. Researchers and hackers may be very good at one or even two of the areas. However, when evaluating a network or planning an attack, they often need tools or skills in which they are not as strong. To acquire the information or tools, they share knowledge, tools, and exploits with others who are skilled in areas complementary to their own.

Note: *While each researcher or hacker may demonstrate aptitude in one or two aspects of research, an excellent researcher is familiar with all eleven fields and will use techniques from each to evaluate or attack computers.*

1. Reverse engineering—Software reverse engineering involves reversing a program's machine code back into the source code in which it was written, using program language statements. In security research, reverse engineering is performed against applications, operating systems, and network protocols. Typically, the researchers and hackers will evaluate error reports from random events or forced error events generated through fuzzing techniques to see how the system responds to unusual data requests or packet structures. Once they have determined how the system responds to a stimulus, they are able to ascertain where the software may be vulnerable to attack and why, enabling them to begin writing POC code to test the hole.

2. Packet crafting is the manipulation of standard packets or generation of unique packets that force a network service device, operating system, or application to respond in a manner providing the attacker with root, or complete administrative access to the vulnerable computer. Packet crafting is primarily a network protocol-based attack type and requires a deep knowledge of networking architecture, protocols, and network service device handling techniques. One of the more well-known packet crafting exploits is the use of fragmented packets to bypass a firewall.
3. Intrusion Detection System (IDS) evasion research concentrates on bypassing IDSs, either host- or network-based. Attackers must be able to break into a system, run commands and software, and communicate remotely with the compromised system without being detected by the defenders. Methods and techniques of bypassing or hiding activity from detection systems are critical when trying to break into a network or system. A common technique for bypassing the Snort IDS is fragmenting a packet and inserting a reset packet between the fragments. The IDS can't match the fragmented packet against its signature set and breaks state on the session, allowing the traffic through. Tools such as Whisker, written by Rain Forest Puppy, and FragRouter are commonly used to defeat IDS software.
4. Operating system attacks take advantage of vulnerabilities within the operating system itself. Developing new vulnerabilities and exploits for operating systems call for low-level expertise in operating system architecture and design, how the OS actually implements the design protocols, and what services and configurations typically run on specific operating systems. Examples of OS attacks include script injection, memory error techniques such as buffer, stack, and heap overflows, or format string attacks.
5. Embedded systems experts prefer to focus on routers, printers, network, security appliances, or other computer systems that use a stripped down version of an operating system or highly compact OS for performing real-time tasks. Embedded systems usually only perform a limited number of computing functions, but need to perform them at a very rapid rate. Printer bounce attacks or Cisco IOS exploits are examples of embedded system hacks.
6. Database researchers and hackers specialize in the design and development of database vulnerabilities and exploit tools. Since databases often have full administrative access to the operating system, a successful attack against the database frequently results in a root-level compromise of the computer. The most predominant form of database exploit currently in use is the SQL injection.
7. Web and application specialists prefer to write tools for use against web servers or other key application software such as FTP clients, anti-virus clients, web browsers, or media players. If the application software is widely used, then it provides a large population of victims for hackers or another venue of entry onto a network by researchers. As with databases, web and application software often run with full administrative access or can be compromised in ways that allow the attacker to easily escalate privileges on the system or network. Common application attacks include cross site scripting or script injection.
8. Mobile device researchers and hackers focus their work in wireless and handheld device exploitation. PDAs, handhelds, cell phones, Blackberries are all common targets and run customized operating systems with different architecture and functionality than standard computers. With the advent of wireless networks and rising use of wireless devices, mobile device hacking is becoming more and more popular. Those devices that offer a more full range of computing capabilities like the Blackberries and iPaqs can be used as a point of entry from which to compromise networks. But, POC viruses (Cabir, MetalGear) and Trojans (Mosquitos) for cell phones have also been released.

9. Shellcoders generally write two different portions of an exploit. They develop the shellcode wrappers, the delivery portion of the exploit, and the shellcode itself, which is the payload of a buffer overflow. Shellcode wrappers are the delivery mechanism of an exploit and manipulate the conditions of a specific vulnerability. Once the wrappers have successfully negotiated the conditions of the vulnerability, the shellcode can be executed. The shellcode is the executable code that results in the root compromise of the computer. Most shellcode spawns a root shell or command prompt from which various commands can be run, allowing the attacker to manipulate the computer and its resources at will.
10. Rootkit writers develop the software that is loaded on the compromised system and used to remotely control its resources. This software also helps clean up log files, prevents detection of the system's compromise by masking illicit activities, provides for remote administration of the host by the attacker, etc. BO2k and t0rn are popular rootkits.
11. Covert channel experts develop the techniques and tools used by researchers and hackers to hide the communications between the compromised host and the attacker. These researchers and hackers create communication channels that are hidden or difficult to detect, so the system administrator and security personnel do not realize illicit activity is happening on the victim network.

6.3 Types of Researchers and Hackers

The two primary types of researchers and hackers are bug hunters and exploit coders. Each type of researcher or hacker specializes in one or two of areas of security research, but prefers either to find new holes or to write exploit code in that area.

Note: *Even though the researcher or hacker may prefer to do bug hunting or exploit coding, he will also demonstrate proficiency at both types of research as the knowledge is essential to become an outstanding researcher or attacker.*

Bug hunters: Bug hunters prefer identifying new vulnerabilities in software to writing industrial strength exploits. They do write POC code or workable exploits, but their core competence lies in their ability to find new vulnerabilities. Vulnerabilities discovered by these hackers and researchers are known to be reliable, result in root compromise of the victim computer, and can be used to develop industrial strength exploits.

The process for finding new bugs or vulnerabilities is outlined below. Bug hunters:

1. Find vulnerabilities through reverse engineering or other techniques
2. Discuss ideas with a close cadre of other researchers
3. Write initial proof of concept code
4. The POC code works on a limited number of hosts but not all instances of the vulnerable software
5. Exploit isn't able to be detected by standard security tools
6. Pass the POC code on to other researchers in exchange for new ideas or tools.

At this point, the POC code is still a 0-day exploit and is not easily detected or stopped by standard IT security tools.

Exploit coders: Exploit coders often fine tune or refine the POC tools given to them by others in exchange for industrial strength tools, but they may also write their own tools based on vulnerability research they have performed. Exploits written by these researchers or hackers are well-known for their reliability and high quality code, meaning they will work on most versions of vulnerable software, regardless of individual configuration, and can be run without interfering with or crashing the system. Such high quality exploits are also known as industrial strength tools.

The process for refining or developing exploit code is outlined below. Exploit coders:

1. Review initial POC code for ease of implementation, reliability of use, and portability to other software or versions of the affected software
2. Discuss ideas regarding the refinement with a close cadre of other researchers and hackers
3. Make changes to the POC code or rewrite the exploit altogether so it works reliably every time it runs on the largest variety of software possible
4. Pass the POC code on to other researchers in exchange for new ideas or tools.

At this point, the industrial strength code is still a 0-day exploit and is not easily detected or stopped by standard IT security tools.

6.4 Awareness in the General IT Security Arena

0-day exploits come to the attention of the general security populace in one of three ways. A researcher or hacker publishes the vulnerability and perhaps POC exploit code for it in order to be recognized for their work. The tool is discovered and analyzed during a forensics investigation of a successful attack in which the exploit was used. Then, the investigators publish their findings to others in the general security world. Or, the tool becomes so widely distributed in the IT security research community that it is eventually leaked to members of the general IT security community because it is no longer viable as 0-day code. Rather, the security vendors and developers have enough information to detect, mitigate, or stop the exploit.

Once the code becomes public knowledge, vendors and developers can issue patches, IDS and AV vendors can distribute signature files to their users, and perimeter defenses can be hardened. People defending general IT networks can begin responding to the problem, preventing the exploit's use on a widespread basis.

6.5 Applicability to the Control System Environment

Many of the general IT vulnerabilities and exploits do not work in control system environments. But, as control system vendors move toward using standardized network protocols, application architectures, and operating systems, these problems will begin affecting control system networks. To mitigate the damage these tools could do once they are ported over for use in a control system network, control system security personnel can begin evaluating IT security research and ascertaining what defensive measures can be taken to prevent their use.

7. HOW IT SECURITY RESEARCH SHOULD BE ANALYZED

7.1 Evaluation Criteria

Not all of the research being done in the IT security community is relevant to control system environments. The two most important criteria for determining if the IT security research could be used against a control system network are: a) the ability to modify an attack methodology for use against a control system network component, operating system, or application, and b) the portability of existing tools for use against a control system network component, operating system, or application. This key issue must be carefully considered when evaluating research for its applicability in a control system environment.

Methodologies: Certain attack methodologies could be modified to work successfully in a control system network. In this report, an attack methodology refers to the way an attack is performed against a control system environment. For instance, packet fragmentation, an example of an attack methodology, can be used to force an error while reverse engineering a vulnerability, to bypass firewalls, or to evade detection by IDS software. Packet fragmentation could also be used against control system networks for the same purposes.

Existing Exploit Code: Some of the exploit code or tools already available for use against IT networks could be modified to work against control system networks as well. Nmap, a popular open-source port-scanning tool, is used by both IT security researchers and hackers to evaluate networks for open ports and services. Currently, it works against the most common operating systems such as UNIX, Linux, Macintosh, and Microsoft Windows. Because several of these operating systems are becoming more popular with control system developers, Nmap features could be modified to identify specific services and ports that individual control system applications use when loaded on the common operating systems. Or, code modules allowing Nmap to assess specific control system components could be developed, giving system-specific information such as proprietary vendor protocols and application services being used on the devices.

7.2 Analytical Tasks

Determine which components of control system environments use software or network architecture common in the general IT world. UNIX, Red Hat Linux, and Microsoft Windows are the most common operating systems in use by both IT and CS vendors. The use of TCP/IP on the networks is burgeoning, along with standard network backbone equipment like switches, routers, or firewalls produced by popular IT vendors like Cisco and Checkpoint. Meanwhile, the applications are becoming more similar to those on IT networks as well. They are increasingly easy to use, well laid out, and equally as dependent on common development environments as those in the general IT world. The use of popular databases like Oracle or Microsoft SQL, web servers like Microsoft IIS, and portable web-based applications built with Java and PHP is rapidly increasing in control system environments, leaving them vulnerable to the same IT attacks.

1. Research what vulnerabilities, attack methodologies, and exploits already exist for the common components by following discussions of new vulnerabilities on websites such as the ISS X-Force or on mailing lists like VulnWatch and FullDisclosure. Read the white papers and discussions of the vulnerabilities and exploit tools posted at research sites like www.insecure.org or www.sophos.com. Attend conferences like LayerOne or ToorCon to learn about the most recent

advances in security technology. If possible, participate in law enforcement forums on security to identify where computer crime is experiencing the most change. These are advance indicators of what research is being done, who is doing it, and how it is being applied.

2. Determine who leads the field in security research for these common technologies by seeing which authors or individuals posting comments are most highly regarded by the IT security community. Try to find out who has the most comprehensive understanding of the issues, workable algorithms for exploiting flaws, frequently used tools, or popular defense techniques because they will also be the most likely people to be influencing how IT networks are secured. They often work with major vendors on critical problems, so vendor security bulletins reference their research in the postings, as do the Open-Source Vulnerability Database (OSVDB) and Computer Emergency Response Team (CERT) entries. The best researchers are also most frequently selected to give presentations at major security conferences like BlackHat. Understanding who is most literate and credible in the IT security research field provides perspective on what exploration of vulnerabilities and exploits will most likely provide pertinent information about network defense in the near future.
3. Track the work being done by the top researchers and their associates to see what fresh directions the research takes. Reading the latest publications, postings, and presentations by these authorities will provide insight into what they consider important or significant in security. Audience response at security conferences like DefCon or CanSecWest provides a key indicator of what studies are truly ground-breaking, as will the length and quality of discussion of the topic by members of security forums and chat channels. If several of the best researchers are discussing the same innovative topics, then the ideas must be considered for their applicability in CS settings and how CS networks can best be protected from attacks originating from the work.
4. Evaluate the work being done in the general IT community to secure their networks against the emerging trends, where the defensive gaps may lie, and what security measures are emerging to deal with the new issues. Follow what the anti-virus and IDS vendors are doing with the new research. And, more importantly, monitor the responses of the vendors to the security issues presented by these experts. Closely track what defensive measures are being taken, the lessons learned from defending against new methods of attack, and how the best researchers are evading these security measures. This gap analysis of new research and current defensive techniques is needed to understand how control system networks will need to adapt their responses to the same offensive ideas and tools before the research can be ported over for use against them.
5. Compare the ability of control system network to defend against these new security issues to those of the general IT community. CS computer systems are much more focused on reliability and little down time for patching and testing, leaving CS administrators with a much less flexible means of securing their systems. By evaluating the lessons learned in the general IT environment and the defensive techniques available, CS administrators can better predict possible issues and workarounds for vulnerabilities. However, combining IT defense information with the understanding of germane, contemporary ideas from security research helps refine these efforts even further by allowing the defenders to prepare for future attacks with more precision, allowing more flexibility in CS system defensive measures.

8. SUMMARY

The standardization of control system applications, operating systems, networking protocols is occurring at a rapid rate and following a path similar to the standardization of modern IT networks. Because of this, control system environments will become more susceptible to common methods of attack as the networks integrate software and software architecture native to general IT networks. Many of the attack methodologies from the IT world can be ported over to the control system environment with little difficulty.

It is extremely important to take advantage of the lag time between new research development, its use on traditional IT networks, and the time it takes to port the research over for use on a control system network. Currently, control system networks are not faced with the overwhelming number of attacks IT networks face each day. However, as the CS networks become more dependent on IT applications and software and become more standardized, compromising these networks will become easier to do because standard IT attack methodologies and tools will work without modification, or can be modified quickly to work against a standardized target.

At this time, people responsible for defending CS networks have a great deal more time to anticipate new forms of attack by studying cutting edge IT security research and evaluating the effectiveness of current defensive responses, and they have time to test and deploy security measures before the research can be altered to work in the CS environment. With the move to interdependence on IP-based networking, integration of CS systems communications over the public Internet, and the interconnectivity, this luxury will not exist for much longer. As the move towards standardization and integration continues, consideration of new IT security research can help influence the inclusion of strong, layered security architecture for CS environments.

Because the defense of any computer network, IT or control system, is a defensive task, any time and information gained by defenders through predictive analysis of IT security research greatly enhances their chance of stopping new types of attacks. Analyzing nascent trends in IT security and weighing their applicability against control system networks would provide significant information regarding the defense mechanisms needed to secure critical infrastructure more effectively. This work would provide the critical infrastructure community with a much better understanding of how new attacks might be launched, what layers of defense will be needed to deter them, how the attacks could be detected, and how their impact could be limited.