

# **Attack Methodology Analysis: Emerging Trends in Computer- Based Attack Methodologies and Their Applicability to Control System Networks**

Bri Rolston

June 2005



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

INL/EXT-05-00477

# **Attack Methodology Analysis: Emerging Trends in Computer-Based Attack Methodologies and Their Applicability to Control System Networks**

**Bri Rolston**

**June 2005**

**US-CERT Control Systems Security Center  
Idaho Falls, Idaho 83415**

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

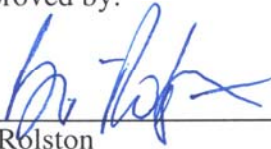
# US-CERT Control Systems Security Center

## Attack Methodology Analysis: Emerging Trends in Computer-Based Attack Methodologies and Their Applicability to Control System Networks

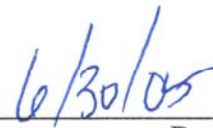
INEEL/EXT-05-00477

June 2005

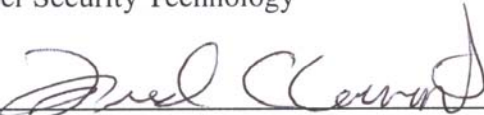
Approved by:



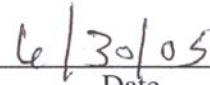
\_\_\_\_\_  
Bri Rolston  
Author  
Cyber Security Technology



\_\_\_\_\_  
Date



\_\_\_\_\_  
Fred C. Cowart  
Program Manager  
US-CERT Control Systems Security Center



\_\_\_\_\_  
Date



\_\_\_\_\_  
Robert W. Hoffman  
Department Manager  
Cyber Security Technology



\_\_\_\_\_  
Date



## **ABSTRACT**

Threat characterization is a key component in evaluating the threat faced by control systems. Without a thorough understanding of the threat faced by critical infrastructure networks, adequate resources cannot be allocated or directed effectively to the defense of these systems. Traditional methods of threat analysis focus on identifying the capabilities and motivations of a specific attacker, assessing the value the adversary would place on targeted systems, and deploying defenses according to the threat posed by the potential adversary. Too many effective exploits and tools exist and are easily accessible to anyone with access to an Internet connection, minimal technical skills, and a significantly reduced motivational threshold to be able to narrow the field of potential adversaries effectively.

Understanding how hackers evaluate new IT security research and incorporate significant new ideas into their own tools provides a means of anticipating how IT systems are most likely to be attacked in the future. This research, Attack Methodology Analysis (AMA), could supply pertinent information on how to detect and stop new types of attacks. Since the exploit methodologies and attack vectors developed in the general Information Technology (IT) arena can be converted for use against control system environments, assessing areas in which cutting edge exploit development and remediation techniques are occurring can provide significance intelligence for control system network exploitation, defense, and a means of assessing threat without identifying specific capabilities of individual opponents.

Attack Methodology Analysis begins with the study of what exploit technology and attack methodologies are being developed in the Information Technology (IT) security research arena within the black and white hat community. Once a solid understanding of the cutting edge security research is established, emerging trends in attack methodology can be identified and the gap between those threats and the defensive capabilities of control systems can be analyzed. The results of the gap analysis drive changes in the cyber security of critical infrastructure networks to close the gap between current exploits and existing defenses. The analysis also provides defenders with an idea of how threat technology is evolving and how defenses will need to be modified to address these emerging trends.



# CONTENTS

ABSTRACT.....	v
1. INTRODUCTION.....	1
2. WHY TRENDS IN IT SECURITY RESEARCH ARE IMPORTANT FOR CONTROL SYSTEM SECURITY.....	3
2.1 0-day Exploits as Threat Vectors .....	3
2.2 Ripple Effect of Data Sharing .....	3
3. SIGNIFICANCE OF OPEN-SOURCE IT SECURITY RESEARCH.....	5
3.1 Life Cycle of a 0-day Exploit .....	5
3.2 Value of Open-Source Security Research .....	5
3.3 Nature of the Computer Security Research Community .....	5
3.4 Applicability to the Control System Environment .....	6
4. ATTACK METHODOLOGY ANALYSIS (AMA).....	7
4.1 AMA as a Threat Analysis Tool.....	7
4.2 What is AMA?.....	7
4.3 Level of Vulnerability .....	7
4.4 Value of Systems.....	8
4.5 Example of a Threat Assessment Using AMA.....	9
4.6 Using Gap Analysis to Direct Defenses .....	10
5. ANALYZING IT SECURITY RESEARCH FOR THREATS.....	13
5.1 Ascertaining What Information Is Useful .....	13
5.2 AMA Tasks .....	13
6. SUMMARY .....	14
Appendix A What Is A Security Researcher?.....	15
Appendix B Preliminary Analysis Information .....	21
Appendix C Life Cycle of 0-Day Exploits.....	25





# **Attack Methodology Analysis: 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.

Historically, threat analysis relied upon a solid understanding and analysis of four aspects of threat: a) who is interested in attacking the U.S., b) how strong their motivation for doing so might be, c) where U.S. defenses are weakest, and d) what capabilities known adversaries have to exploit those weaknesses. Conventional threat analysis techniques do not work well, however, when trying to characterize computer-based threats. Established techniques fail when applied to cyber threats because it is difficult to obtain necessary information rapidly enough and the traditional threat model is not as scaleable or as dynamic as is needed to adequately analyze cyber threats.

Too many people have the resources required to carry off a successful cyber attack—an Internet connection, an understanding of control system networks, and strong technical skills—to identify potential cyber adversaries with any degree of accuracy. Because of the ready availability of the required resources, people who may not have the motivation or assets necessary to perform a physical assault can perform computer-based strikes against a target with relative ease.

How, then, can an effective understanding of cyber adversaries be developed? Because exploit methodologies and attack vectors developed in the general Information Technology (IT) arena can be converted for use against control system environments, analysts can identify which of these ideas are most likely to be used against control systems and what types of defenses will be needed to deter the threat. Understanding how hackers evaluate new IT security research and incorporate significant new ideas into their own tools provides a means of anticipating how IT systems are most likely to be attacked in the future.

This analysis technique, Attack Methodology Analysis (AMA), can be used to validate exploit technology that can be used against control system environments, identify gaps that exist in control system defenses, and associate known mitigation techniques with the defensive gaps. This research maps IT-based exploit tools and attack methodologies to specific vendor control systems, thereby supplying defenders with pertinent information regarding the detection and mitigation of cyber threats to control systems. By assessing areas in which cutting edge exploit

development and remediation techniques are occurring, analysts can develop significant intelligence regarding control system network exploitation capabilities, practical defensive techniques for new threat vectors, and threat assessments in a dynamic fashion and without having to characterize adversarial capacity.

Attack Methodology Analysis begins with the study of what exploit technology and attack methodologies are being developed in the IT security research arena within both black and white hat communities. Once a solid understanding of the cutting edge security research is established, emerging trends in attack methodology can be identified and the gap between those threats and the defensive capabilities of control systems can be analyzed. The results of the gap analysis drive changes in the cyber security of critical infrastructure networks to close the gap between current exploits and existing defenses. The analysis also provides defenders with an idea of how threat technology is evolving and how defenses will need to be modified to address these emerging trends.

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. Once these trends have been identified, a gap analysis between cutting edge research and current defensive capabilities of control system (CS) networks can be performed to pinpoint where the CS networks are most vulnerable. This approach to threat analysis allows defenders to focus their resources on the most vulnerable points of the network in a logical manner.

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

### **2.1 0-day Exploits as Threat Vectors**

*Note: 0-day exploits are those exploit tools that utilize vulnerabilities unknown to the general IT security community or exploit code for known vulnerabilities that the general security community does not have detection mechanisms. Once 0-day exploits become known to the general security community, protective measures are rapidly developed to block, detect, mitigate, or fix the problem.*

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 have already been identified by members of the general IT community. Security professionals, or those IT experts who implement computer security technology but do not contribute to security research, are not members of a very small, close-knit research group that develop cutting edge attack methodologies and techniques, also known as 0-day exploits.

Rather, they are dependent upon 0-day research becoming well-known, allowing vendors and development groups to gather enough information about the exploit technique to fix the vulnerability. In short, only known attack methodologies can be blocked, leaving a gap between what is known to the general security community and what true capabilities exist to carry off a successful attack against IT—and, potentially, control system—networks.

By ascertaining which 0-day research could be used effectively against control systems, threat vectors on control systems can be identified by defenders before those openings are used to mount an attack against the network. If the holes a hacker would use to get onto the control system network can be detected early on, defensive measures can be put in place to eliminate the threat vector entirely or block an attacker's access to that point of entry.

### **2.2 Ripple Effect of Data Sharing**

While most people in the general IT security community do not have access to information regarding 0-day research, a small cadre of IT security experts who are very well informed about the latest trends in attack methodologies and current 0-day exploits does exist. (See Appendix A for more information on IT security researchers and hackers.)

These are the IT security researchers, who are always communicating with others in their field and who 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 the new attack techniques and methodologies (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.

Attack Methodology Analysis takes advantage of the ripple effect of data sharing within the elite circles of IT security research to identify emerging threats and how they can be mitigated. This analysis can be used to detect cyber threats to specific software components of a control system, ranges of control systems produced by individual vendors, or a particular network architecture deployed by a company or organization.

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

## **3. SIGNIFICANCE OF OPEN-SOURCE IT SECURITY RESEARCH**

### **3.1 Life Cycle of a 0-day Exploit**

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 proof of concept (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, Intrusion Detection System (IDS) and Anti-Virus (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. (See Appendix C for more information regarding exploit development.)

### **3.2 Value of Open-Source Security Research**

Much of the best work in the IT security research is open-source and can be evaluated or used by anyone with an Internet connection and interest in the field. While much of the newest, most innovative work is known only among those in the elite research community, the researchers do share some new ideas in open-source channels such as mailing lists, vulnerability advisory postings, and security conferences. Other individuals may actually publish working exploit code for previously unknown vulnerabilities on hacking websites, covert IRC channels, or limited access mailing lists.

This is significant for two reasons: a) mature exploitation tools are readily available for potential attackers to use, which increases the threat to control system networks, and b) open-source information regarding potential threats can be gathered without requiring information regarding the attack capabilities of specific adversaries.

### **3.3 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.

### **3.4 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. (See Appendix B, “Preliminary Analysis Information”.)

## **4. ATTACK METHODOLOGY ANALYSIS (AMA)**

### **4.1 AMA as a Threat Analysis Tool**

Attack Methodology Analysis was developed specifically for performing threat assessments on computer-based networks. As such, it is a flexible, dynamic, and scaleable model for measuring cyber threats and can be used without having to identify a specific adversary or adversarial capability. AMA uses baseline information about system security and publicly available information regarding existing threat technology that could be used by an adversary to take advantage of the system's weakest points.

Once a network's vulnerabilities have been base-lined and a value for the system has been established, open-source research can provide information regarding what attack methodologies and tools are available for a potential adversary to use. Threat analysis efforts can then be focused on the threat posed by exploit technology rather than the threat posed by a specific adversary.

### **4.2 What is AMA?**

AMA is the process of identifying components of control systems, identifying vulnerabilities in those components, mapping existing exploits or attack tools to those vulnerabilities, and analyzing the gap between current defensive capabilities for those vulnerabilities and accessible exploit technology. The gap between the defensive techniques and the offensive resources is the level of vulnerability that exists on the system. The threat can be determined by weighing the level of vulnerability on the system in conjunction with the value the system has on the network.

Threat is measured using the following formula:

$$\text{Threat} = \text{Level of Vulnerability} + \text{Value of System}$$

### **4.3 Level of Vulnerability**

The level of vulnerability on a computer host, including those in a control system, is based upon the number of security holes open on the host in any given configuration. Because new vulnerabilities are published to security websites or mailing lists daily, the level of vulnerability existing on a network changes frequently as well. Once these weak areas are identified by vendors or software development communities, a means of securing the hole can be developed. Until the vulnerability is published and a fix is released, however, systems remain vulnerable to attack via the hole.

The significance of a vulnerability is based upon how widespread the flaw is, how much access it provides an attacker, and how severe the consequences are once the hole has been exploited. If a problem exists in a piece of software that is widely used, for example a popular database, operating system or anti-virus program, it raises the importance of the vulnerability. If the hole grants an attacker a great deal of access to resources on a susceptible computer, i.e., automatically running a malicious program on the host, the severity of the vulnerability is

increased as does the need to develop and deploy a fix for the problem. Finally, if the flaw results in severe consequences such as permitting the adversary to gain remote control or to escalate privileges on the box, the vulnerability becomes even more significant in the eyes of the security community. The more significance or critical the nature of the hole, the more important it is to fix the problem as quickly and thoroughly as possible.

The level of vulnerability inherent on a host changes frequently as new software is added, applications are upgraded, administrative policies are modified, or additional security is implemented. This dynamic state is one reason it is difficult to apply traditional threat analysis to cyber security threat modeling. The rapidly evolving profile of the target system makes it difficult to truly assess threat from a specific adversary. Threat assessments based on identifiable weaknesses of a computer provide a more flexible means of evaluating a control system's true threat profile.

Establishing a computer's level of vulnerability can be done by simply tracking open-source information security research and having a general understanding of a network's architecture. For example, a highly-esteemed security researcher posted news in November 2004 that his company's penetration testing software was leaked. The researcher excels at memory attacks and enjoys world-wide recognition for his skill. The product was his company's primary commercial offerings and contains easy-to-use, reliable, 0-day exploits to use against a wide variety of operating systems and applications. Of most interest, though, was the inclusion of a previously-unknown attack targeting the Microsoft server platform and one of its key networking components. This was a serious problem for Microsoft systems administrators because the exploits were unknown in the security community, there were no existing means to detect the exploit, and the vendor had not yet released a patch to fix the hole. Security experts began identifying attacks using the exploit within days of the leak once details of the flaw had been published by the author. The code provided black hat attackers with a reliable, easy way of taking over any host running Microsoft server with the WINS server service turned on, which it is by default in 2 of the 3 most widely deployed versions of the operating system. Therefore, any control system deployed with a Microsoft server running the WINS service could immediately be classified as having a high level of vulnerability.

This type of vulnerability assessment, which establishes the level of vulnerability, can be done for individual software components of a control system, for specific computers on the control system network, or for the control system network as a whole. This approach takes advantage of the accessibility of open-source information regarding software flaws and exploits to establish a level of vulnerability based on the software components on a computer or network. This eliminates the need to identify a specific adversary or his attack capacity.

## **4.4 Value of Systems**

A computerized host, including the computers on a control system network, can have different values in an organization. The computer could be a key component of the control system and its failure could destabilize the network should hardware failure occur or any of its software fail to operate correctly. The data on a box might have significant economic or functional value to an organization; the loss or corruption of the data would greatly impact business or control system operations. The system may also be of high consequence because it is



connected to other high value systems and has been identified by adversaries as an entry point for their attacks. The value of the system as a target for a cyber adversary is what is considered in AMA.

In AMA, the value of the host is determined by its worth to a potential attacker and the probable consequence of its compromise. For instance, any trusted host residing on the control system network that has permission to connect to the business network would be of high value to an adversary because it can legitimately exchange data outside the trusted network and is an access point to the control system network. The system would also be of consequence because it resides on the trusted network for a reason and is most likely has permission to connect with other high-value targets.

Using the example of the exploit code leaked from the toolkit of a prominent IT security researcher above, a basic determination of the value of a system can be performed for any control system running Microsoft server. (A list of vendors and systems with integrated Microsoft servers can be gathered from the Internet.) Any computer running Microsoft server operating system and the WINS server service is most likely a key networking component on the control system network. Microsoft operating systems are commonly deployed because they centralize network control in one or two key points, generally computers running the server software, for ease of administration. Any Microsoft server with WINS server service enabled would probably have greater access and higher level of privilege on any of the other Microsoft hosts in the environment. Given the basic understanding of a Microsoft environment, a control system that had integrated Microsoft server software with WINS services is a high value target for an attacker.

Again, AMA can be used to generate the value for systems by analyzing easily accessible technical information. Since most control system vendors post supported software on their websites, gathering the basic data needed to evaluate system value is a time-consuming but not difficult task.

#### **4.5 Example of a Threat Assessment Using AMA**

The following threat evaluation is an example AMA product assessing the threat faced by an operational IT network and by those customers with Microsoft servers integrated in their control system networks in December 2004. The example uses the WINS Server Service vulnerability detailed in previous sections.

Several mitigation strategies could be used, but most of them are not practical to use on a control system network, are not widely deployed, or would cause too much interference with system functionality. The exploit traffic could be blocked if firewalls or Intrusion Prevention Systems (IPS) were in place and correctly configured. However, many control system networks allow dial-up access that often bypasses the IPS and firewalls between the control system and corporate networks. Intrusion Detection Systems (IDS) could detect the attack if the IDS signatures were up-to-date and the logs were being monitored regularly. Of the Microsoft server platforms, only Windows Server 2003 has a host-based firewall and most control system do not incorporate host-based protections such as a firewall. The WINS service can not be disabled on Windows Server NT 4.0 without greatly interfering with the networking functionality; Windows

Server 2000 and 2003 generally run the WINS service as well—although the operating systems can run using only DNS—in order to provide backward compatibility with NT 4.0 machines or because the domain structure is dependent upon WINS.

The level of vulnerability for a Microsoft server was very high until a patch was released in December, leaving systems vulnerable to the exploit for almost three weeks. The value of a system running the software was also high given the nature of basic Microsoft network architecture. The threat level was assessed as high and was verified within days of the release of the technical parameters to the general IT security community. IDS development groups, honeynet researchers, and systems administrators all announced malicious activity utilizing the exploit tool immediately after the publication of the exploit methodology. Further investigation revealed attacks had been carried out against systems before the general IT security community received notification of the problem. So, in this case, the threat to control systems running Microsoft server was very high until the patch was released and verified for use on control system networks by individual vendors.

## **4.6 Using Gap Analysis to Direct Defenses**

The final step in the AMA process is performing a gap analysis between a network's current defenses and existing exploit technology. The gap between what the network's protective measures can deter and what can still penetrate the network's barriers directs how defensive resources should be allocated. The short-term or immediate threats to system security are driven by the existing threats and the level of security that can be implemented to prevent them.

Control system networks require more carefully constructed security architecture because they lack the resilience and flexibility of general IT networks and there are few customized defense mechanisms built into them. The primary short-term defenses on current control system networks should focus heavily on detection and blocking mechanisms while workarounds and core problem solutions are tested for use on the most valuable systems.

Emerging threats on the IT network should also be evaluated for potential use against control system environments; and, as these threat vectors make their way into the general IT community, members of the control system community should pay close attention to the mitigation techniques used. By studying the lessons learned in the IT community and tracking the evolution of the exploit and defensive technology, the control system industry can produce solid mitigation procedures that are engineered specifically for control system environments and begin to integrate core structural defenses into the control system architecture from the beginning of the product's lifecycle.

A gap analysis using the WINS Server Service vulnerability and threat assessment from preceding sections was performed against an operational IT network and a test control system network. The time lag between the release of the vulnerability's technical parameters and the security fix from Microsoft was especially problematic. The author of the exploit code is a professional security researcher whose firm specializes in providing advanced penetration tools to its customers. The code was reliable and worked against most Windows servers running WINS service.

The general IT community immediately began to generate detection signatures for the anti-virus and IDS software. Incoming traffic characteristic of the attack, UDP calls to port 42, were blocked on perimeter routers and at the firewall. Servers that did not require WINS service were configured to run with the service shut down or host-based firewalls were run on the systems. Servers that could not shut down the service could be monitored to check for successful exploitation using internal IDS systems and the servers' own event logs. Once the patch, which was developed by Microsoft with the help of the researcher who wrote the exploit code, was released, the general IT community deployed the fix across their enterprise networks.

The control system community was not as fortunate because the short-term gap analysis showed significant problems with existing defenses and the WINS exploit. While perimeter defenses such as firewalls are often used between the corporate and control system networks, simply blocking the exploit traffic is not enough. An attacker could use another attack vector to get on the control system network and still use the WINS attack to take over a key Microsoft server. IDS systems help detect that the attacks are occurring, but they have to be deployed on the control system network and its key traffic flow areas to be of use. Many control system environments lack customized IDS systems and the personnel, who must have experience with log analysis and the CS network activity baseline, to monitor the logs.

Turning off the WINS service may have worked on CS servers running newer versions of the Microsoft Windows operating system (Windows Server 2000 and Server 2003), but older Microsoft operating systems (Windows NT 4.0 and Windows 3.5.1) must have WINS service running to function. Finally, operating system patches must often be tested and approved by the vendor before they can be deployed to operational control system environments. And, once the vendor approves of the patch, extensive testing on individual CS back up servers should be performed to ensure that the operating systems on customized networks do not interfere with functionality. These issues cause security deployment on control system networks to be a much more problematic task than security deployment on general IT networks.

Long-term gap analysis of the WINS Server Service attack methodology and control system defenses showed key areas that should be addressed by the control system vendors. Some of these issues include:

- Relatively inflexible control system applications—CS applications should be designed more robustly so they can withstand the occasional vagrancies of patch deployment. Flexible software architecture design became more critical in the general IT community as security became a higher priority and CS developers can use many of the same techniques to improve their products.
- Lifecycle management of core COTS components—Microsoft designs their operating systems to last five years and then security support for those products is slowly discontinued; NT 4.0 patches are no longer released unless a customer pays for the service specifically, and Windows 2000 Server support is scheduled to end within the next 18 months. CS vendors and consumers should begin to plan on how they will manage the security risks posed by legacy COTS software and what the software upgrade cycle should be for CS applications using COTS products.

- Lack of patch management and security policy—Standard test procedures for patches and software upgrades should be well-documented by vendors and consumers so security fixes can be deployed as soon as possible when critical problems arise. Security policy should be designed to facilitate the rapid deployment of layered security tools.

## 5. ANALYZING IT SECURITY RESEARCH FOR THREATS

### 5.1 Ascertaining What Information Is Useful

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 exploit tools could be used to run attacks against a CS network immediately, then the defender must evaluate what strategies, tools, and policies are available to deflect, mitigate, prevent, or identify the existing exploit tools and test them for use on a CS network. Second, if attack methodology could be used to develop customized CS attacks, then significant effort needs to be put into developing the security architecture deployed in control system environments to address emerging threats.

### 5.2 AMA Tasks

There are five general tasks that must be performed to perform a cyber threat assessment successfully.

- Baseline the control system network architecture and components. This can be done for a specific network, i.e., an individual utility's CS network, or for individual components integrated into multiple vendor systems, i.e., primary COTS software used to develop web-based consoles for the application.
- Determine the value of control system components to an attacker and the impact a root compromise of the system would have on the control system network. This helps determine where defenses are most critical and prioritizes the security deployment timeline.
- Map known vulnerabilities to the control system components on network communication and infrastructure, operating system, and application levels to establish a level of vulnerability. This establishes a level of vulnerability and can also be done for specific components or for individual networks.
- Match existing exploit tools and attack methodologies to the vulnerabilities identified on the control system network or its integrated components. If exploit tools for the recognized vulnerabilities are available to the general IT security community, a higher level of threat exists on the network and the security deployment timeline should be accelerated. If emerging threats can be identified in the attack methodology trends but no known exploits are circulating in the general security community, the evolution of the methodology should be tracked as it becomes more widely used on general IT networks.
- Evaluate current defensive strategies and tools deployed on the CS network for the mapped vulnerabilities, exploit tools, and attack methodologies. The gap between the CS network defensive capabilities and existing exploit code should drive security measures and defensive resource allocation.

## 6. SUMMARY

Attack Methodology Analysis allows the control system community to assess the threat of a cyber-based attack more flexibly and dynamically than traditional threat models. AMA is an actor-independent threat assessment technique that allows control system defenders to utilize known information regarding system value and level of vulnerability to determine threat level, rather than focusing on the attack capabilities of certain adversaries.

By tracking open-source IT security research and mapping emerging trends to control system components, the control system community can begin to identify potential attack vectors onto the CS networks and the tools most likely to be used to exploit those vulnerabilities. This information would allow the vendors and CS consumers to focus their efforts in a more effective, timely fashion.

The intelligence generated by AMA analysis can be used to improve security in a number of ways. For example, control system operators could be trained to assess their networks with AMA or have someone else perform an AMA threat assessment; then, they could make immediate changes to security architecture and policy based on the threats specific to their network. AMA analysis tying emerging threat information to particular CS applications could be used to help vendors design better control system applications with integrated security functionality and guide future security development efforts. Finally, AMA evaluations could be used to identify threat trends in the general IT security community that impact the COTS software or CS standards that are integrated into various vendor systems. If the trends can be detected early enough, notifications with technical parameters and recommended mitigation techniques could be sent to affected vendors and consumers before widespread attacks occur.

## **Appendix A**

### **What Is a Security Researcher?**





## Appendix A

### 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.

**Note:** *While each researcher or hacker may demonstrate aptitude in one or two aspects of research, i.e., covert channel communication, IDS evasion, etc., an excellent researcher is familiar with multiple fields and will use techniques from each to evaluate or attack computers.*

### 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.”*

### 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.

## **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.

## **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 Peekabooby, 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.



**Appendix B**  
**Preliminary Analysis Information**



## Appendix B

### Preliminary Analysis 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 is 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 research computer security as a hobby.



**Appendix C**  
**Life Cycle of 0-Day Exploits**



## Appendix C

### Life Cycle of 0-Day Exploits

#### 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.

#### 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 IDS software, 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 cannot 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, and 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 iPaks 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.

## **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 is not 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.

## **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.