Lessons Learned in Cyberspace Security

Margaret M. McMahon, Ph.D. and Lori DeLooze, Ph.D.

ANRC, LLC

5309 Wurzbach Rd.

Suite 101

San Antonio, Texas 78238

{mac,lori}@anrc-services.com

Paper: Number 097

Lessons Learned in Cyberspace Security

Margaret M. McMahon, Ph.D. and Lori DeLooze, Ph.D.

Abstract

The lessons learned in cyberspace operations continue to shape cybersecurity education. When a computer is connected to any network, it is immediately vulnerable to both direct and automated attacks. The number of threats in cyberspace are beyond those experienced in the physical world.

After analyzing the history and shape of evolving cyberthreats, several key concepts emerge. In keeping with the theme of Lessons Learned from Research and Operations, the authors discuss their lessons learned about cybersecurity. Their experience was gained during their years in operational communities, doing test and evaluation, and later, as educators of military and DoD students. The intent of this paper is to bridge the language and topics of malware used in academia to the operational community, and to provide a lingua franca to support a dialog between the communities.

We enumerate the top ten concepts that operators, developers, maintainers, and managers need to address to stay safe in cyberspace. Each concept is briefly discussed; its impacts are explained; the main takeaways; and the relevance to Command and Control (C2). The paper also discusses how individuals can continue to increase their awareness of cybersecurity threats and vulnerabilities.

Introduction

Security professionals must learn to communicate using the vocabulary of the security domain. Knowing the common language used when discussing malicious software (malware) enables a user to communicate problems and concerns with other specialists in the field. This language includes the types of malware and the lessons learned in studying each.

Malware can be categorized into one of several different categories according to specific security features, or it can be characterized by features that fit into one or more categories concurrently. This paper focuses on malicious software and not the misuse of protocols, such as when normal network conversations are used maliciously in distributed denial-of-service (DDoS) attacks. In addition, we will concentrate primarily on the Windows Operating System (OS). Because it is the most widely-used consumer OS, it is a major target for malware programmers. Windows users need to be aware of the wide range of threats to their computers. Even older threats can be destructive to a computer that is does not have up-to-date antivirus software or the network it connects to is not protected by a well-configured firewall.

Background

The authors have had the privilege of teaching cybersecurity to those on the front lines of cyberwar, and to those who support them. Our approach to educating cybersecurity students contains a minimum of lecture with a large percentage of hands-on exercises. During course development, we strive for less than forty percent the class time dedicated lecture with at least sixty percent of the time available for related labs to reinforce the important concepts. In a malicious software (malware) class, students follow a standard malware analysis process [14], build an analysis workstation, and learn new tools as they investigate numerous malware samples. Students use a series of static and dynamic analysis tools and techniques to determine the mechanisms used by the malware to cause changes in the files, registry, open ports, and processes (FROP) on the victim machine. Students then interact with the malware and investigate the differences between a clean system and an infected one.

The lessons we have learned about cyberspace are enumerated in the top ten concepts that operators, developers, maintainers, and managers need to address to stay safe in cyberspace.

Related Work

Carpenter [4] brings to light the evolution of Command and Control (C2) being coupled to emerging technology. The evolution of C2 in malware also follows technology. Dittrich and Deitrich [6] define malware's C2 as being single-threaded, distributed, and peer-to-peer. Malware has evolved from a single thread that reaches back to a source to download a malicious executable, to botnets that make a computer function as a node in a peer-to-peer network.

Top Ten Concepts

The top ten concepts will begin with a simple virus and build in complexity to advanced persistent threats, with each concept building on the previous ones, as illustrated in Figure 1.

Each of the ten concepts will include: a brief definition; the impact to a computer or organization, including typical behavior; the takeaways that each person must understand about the concept; and how it can potentially affect C2. While we focus on the examples and mechanisms of malware on computers running the Windows operating system, the same principles will apply to other operating systems. For malware to run on a computer, its code must be built to execute on that computer's specific processor, using that computer's architecture. For example, a virus written for a computer running Windows will not execute on a UNIX computer.

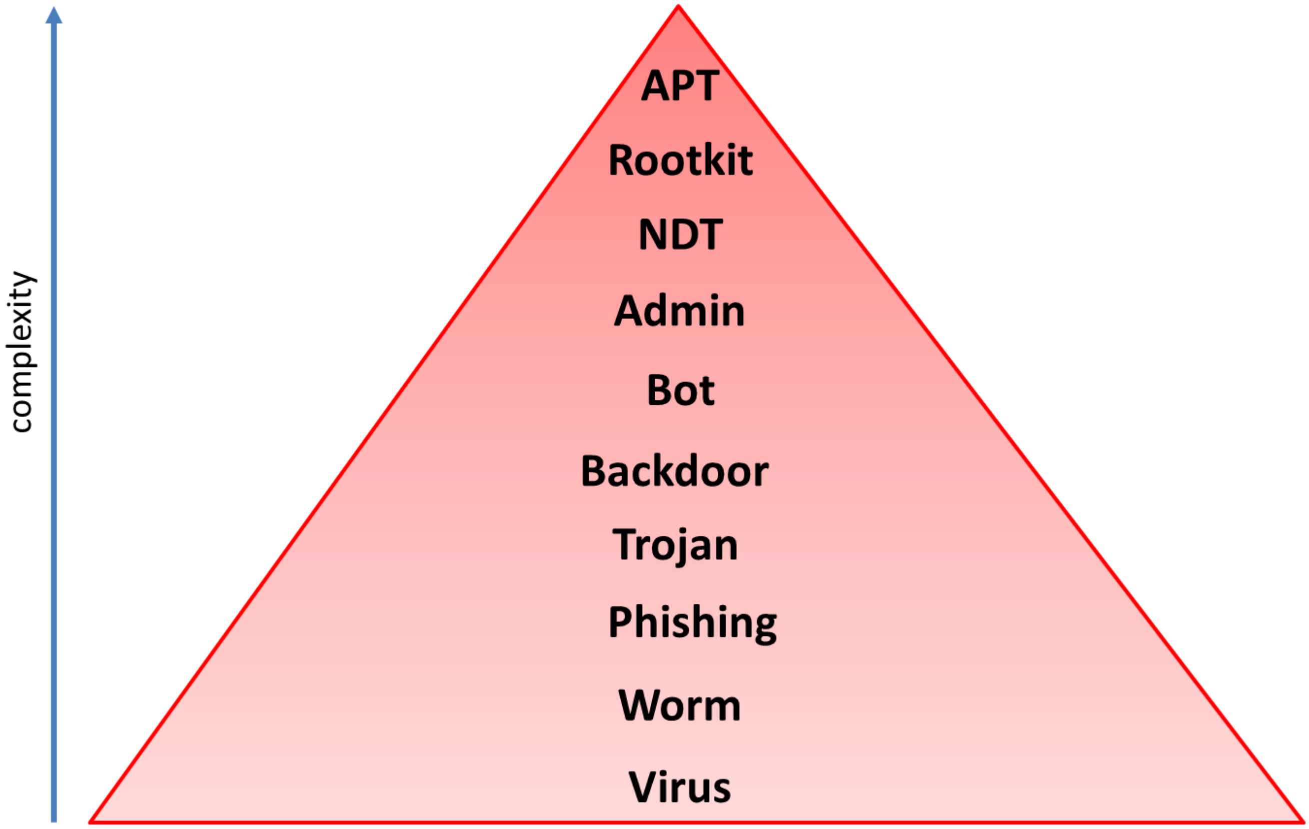


Figure 1. Threats by Complexity.

Virus

Description. A virus is a piece of code that lives as part of a program. Like a virus in the real world, it needs a living program to propagate, and cannot move to another computer without being placed there. Propagation can also occur with automated mechanisms that are enabled on the infected machine, such as a macro.

Melissa is an example of a Macro virus. When it runs, it checks the computer's registry to see if the computer has been previously infected. On uninfected machines, it mails itself to top 50 contacts in the mail program. When the minute after the hour matches the day of the month, for example at 11 minutes past each hour on the 11th of December, the message stored in the virus payload is inserted into an infected document.

Impact. While the virus, unleashed in March 1999, did not erase files or permanently damage the 1.2 million computers it infected around the world, prosecutors say it caused widespread disruption and cost businesses $80 million [12].

Takeaway. The virus propagated itself through e-mail attachments because macros were enabled by default on the infected machines. An organization needs to be running antivirus (AV) scans on both incoming e-mail and documents, and disable the default use of productivity tools, such as macros.

Relevance to C2. Malware that disrupts C2, like Stuxnet, can be downloaded as a virus attachment to e-mail.

Worm

Definition. A worm is a program that replicates itself, and propagates on its own. The first worm unleashed on the Internet was the Morris Worm, written by student at Cornell in 1988 [15]. The Conficker worm replicated itself and subverted Windows and third party security software, making the computer vulnerable to infiltration and further infection [3].

Impact. Any computer on a network is vulnerable. Since a worm reproduces itself, a system's performance can be seriously affected by the number of processes that begin running. The Morris worm prompted DARPA to fund the establishment of the Computer Emergency Response Team (CERT) at Carnegie Mellon University to give experts a central point for coordinating responses to network emergencies.

Takeaway. Any shared drive or trusted connection allows a worm to propagate. The way to contain a worm is to control propagation and end the worm programs running on the computer. By hardening password protection of shared drives, propagation can be prevented.

Relevance to C2. Worms propagate quickly, consuming resources, such as bandwidth in a communications channel or processor cycles in a key computer in a network. For example, infecting a computer in an air operations center could disrupt the C2 infrastructure by propagating throughout a sector and consuming resources, ending real-time situational awareness.

Phishing

Definition. A user is lured into opening an e-mail with an enticing subject, or a demand that urges immediate action [5] and clicks on a link in the e-mail, initiating the download of a piece of software. There may be a second phase, where downloaded software contacts a computer on the Internet to request a larger piece of software. If the software uses an address expressed in the characters of a uniform resource locator (URL), its request can be made to a different computer at different times. A phishing e-mail can be sent to a random group of e-mail addresses, or it may be targeted. Spear phishing is when the sender has identified a specific group with which the e-mail address owned is associated, and effort is invested to tailor the e-mail. In whale fishing, a larger amount of effort is expended to customize an e-mail to be attractive to a CEO, or other senior corporate officer.

Impact. It is easier to put software on a computer inside a network by phishing than breaking into a computer. Malware can also be downloaded by loading a page with an embedded script or clicking on a link in social media.

Takeaway. Users have to be educated how to recognize fraudulent e-mails, not to click on links in e-mails, and to disable automatic scripts. Simply put, do not open e-mails from people you do not know. Unfortunately, a sender's addresses can be spoofed. All incoming traffic needs to be scanned by AV software for matching known signatures of malware.

Relevance to C2. After researching an organization through social engineering and publically available information, whaling and spear phishing e-mails can be written that appeal to key personnel in a C2 organization.

Trojan

Description. A Trojan is a combination of software; something undesirable is packaged together with something desirable. A classic example was Elf Bowling attachment, which ran rampant through the authors' former school. It combined a fun program featuring elves as bowling pins, however it was packaged with SubSeven (Sub7) malware that allowed remote access to the infected machine. IExpress, which is delivered in the Windows OS, is one of the legitimate tools for packaging multiple software programs together; a self-contained executable can have an installer program packaged with an application executable. It takes very little effort to take an existing malware program and wrap it with a desirable application to make a Trojan.

Impact. Users can install these programs purposely, without any hint of the installation of the malware.

Takeaway. Trojans can be constructed easily. When installing software, verify that the executable has an Md5 hash that matches a published one for the program. If there the good executable is packaged with something else, the hash values will not match. The consequences of a Trojan horse may be the installation of a backdoor, bot, or administrator tools.

Relevance to C2. Similar to phishing attacks, a Trojan horse delivery vector for malicious code that can disrupt and delay C2 channels.

Backdoor

Description. A backdoor allows a remote user to connect to a program running on an infected system in a stealthy manner.

Impact. A component of a Trojan might be a program that opens a port and install a program to allow a remote user full access to the infected computer. A new user could be added with a trusted connection. An Easter egg that includes special code added by a programmer to allow special privileges might be installed.

Takeaway. For secure systems, all source code should be inspected and compiled by your organization. Recognize that the use of proprietary OS and application code makes this impossible. The use of open source software requires that internal assets become experts in the source code, which may not be economically feasible. Applying dynamic malware analysis techniques when running the code may yield insights into the legitimacy of its behavior, but a program may execute differently each time it runs.

Relevance to C2. The installation of a backdoor allows an intruder to access, modify, and delete critical configuration information. In this manner, the C2 nodes can either cease to function properly, or worse function with detrimental effects.

Bots

Description. A bot is a program running on an infected machine that becomes one of a group of zombie computers that respond to the commands of a herder. The herder controls the bots through a C2 structure of a botnet. By commandeering the resources of many computers, the herder can solve a distributed processing problem such as password cracking, send spam, or launch a Distributed Denial of Service (DDoS) attack. A herder either uses all the computers personally, or rents them out. One study in 2013 estimated that 61.5% of the traffic on the Internet was from bots [8].

Impact. An infected computer in your network will begin to beacon to its controller, sending period heartbeat messages. It may download file(s) to execute, and coordinate with other bots to perform a task or mount an attack. If it becomes part of an attack, it may generate a lot of traffic to one address.

Takeaway. The generic sign of an infected machine is the network traffic it generates, however it may exhibit different behavior over time based on who is leasing it.

Relevance to C2. A bot can be a located in a C2 node, using that node's bandwidth for its covert channel. A node in a tactical setting, such as a radio with limited bandwidth used in an amphibious assault, could be beaconing as a bot, or worse, covertly exfiltration critical information. In an operational area, the amount of bot traffic on the network could go undetected, and the ability to reprogram an affected unit would be extremely limited.

Admin tools

Description. System administration tools can be installed as a service on a computer, and there are legitimate reasons to remotely access the service through an open port. However, these same tools can be installed and used by attackers. One of the malware delivery methods described earlier can bring the tool to your computer, and allow an attacker full access to a computer in your network.

Impact. Since frameworks of tools are widely available, those without extensive technical skills can use them against your computer. After installing these tools, a remote user has complete access to an infected system, including the addition, deletion or modification of FROP.

Takeaway. Leaving ports open for legitimate tools is always risky, because it is an open door for attackers. When using these tools, monitor activity and check it against a baseline of normal activity. The existence of an unauthorized admin tool requires an investigation into its origin.

Relevance to C2. In addition to full control to add, delete and modify anything on a computer, these allow remote access to video feeds. Using a video feed, an intruder is present in the room or on the computer to observe actions or intentions. These tools could be installed using a Trojan horse vector.

Network Diagnostic Tools

Description. A Network Diagnostic Tool (NDT) automates a checklist for vulnerability tests on the systems in your network. Tools such as Network Mapper (NMAP) assist a system administrator by providing insights into the configuration of computers on your network [16]. However, when these tools are surreptitiously installed by one of the previous methods discussed (e.g. clicking on a link) it can inventory all the operating systems on your network, and provide an attacker with a map of your networks and a list of the vulnerabilities.

Impact. The immediate impact on your system is an increase in internal network traffic. The reports of the scans may be sent outside the network to an attacker, or held for later retrieval.

Takeaway. NDTs can be both good and bad. In the right hands, they are powerful diagnostic tools; in the hands of an intruder they are powerful reconnaissance tools. An intrusion detection system (IDS) and internal sensors need to be installed to detect unauthorized scanning activity on your network.

Relevance to C2. The output of the vulnerability scanners available in network tools provide a blueprint for a successful attack on the computers in a network.

Rootkits

Description. Rootkit malware gives the attacker full administrative privileges, and can hide the existence of FROP from normal tools. Rootkits can operate at different levels of OS privilege, and might be delivered by one or the previous methods, or arrive in the firmware of a new computer. The ones usually encountered with malware are user or kernel rootkits.

Impact. Rootkits interfere with what the user tools report about FROP, allowing malware to subvert your computer without leaving a trace. In effect, you do not know what you do not know. Once a rootkit is found, it may be incredibly difficult to eradicate, and the best choice may be to reload the OS.

Takeaway. There are times when you cannot trust normal tools. Take care when installing privileged types of software that run in privileged mode. For example, do not use unsigned drivers in Windows OS. Regularly scan with the tools that bypass the OS to locate hidden FROP.

Relevance to C2. As with any computer, the installation of a rootkit on a node allows any or all of the above types of malware to operate covertly. This feature of malware is particularly insidious, because it has to be diagnosed indirectly or through the use of special tools.

Advanced Persistent Threat

Description. As the name implies, these are attack mechanisms that are created by skilled developers, who may use series of stages, to meet a specific objective. These attacks are by forces who have determined they will get something from their target. Unlike random attacks of script kiddies, they will use any technique and invest copious amounts of time and resources until they obtain their goal [1, 9]. The most widely publicized APT to date is the Stuxnet virus that targeted a specific industrial control system.

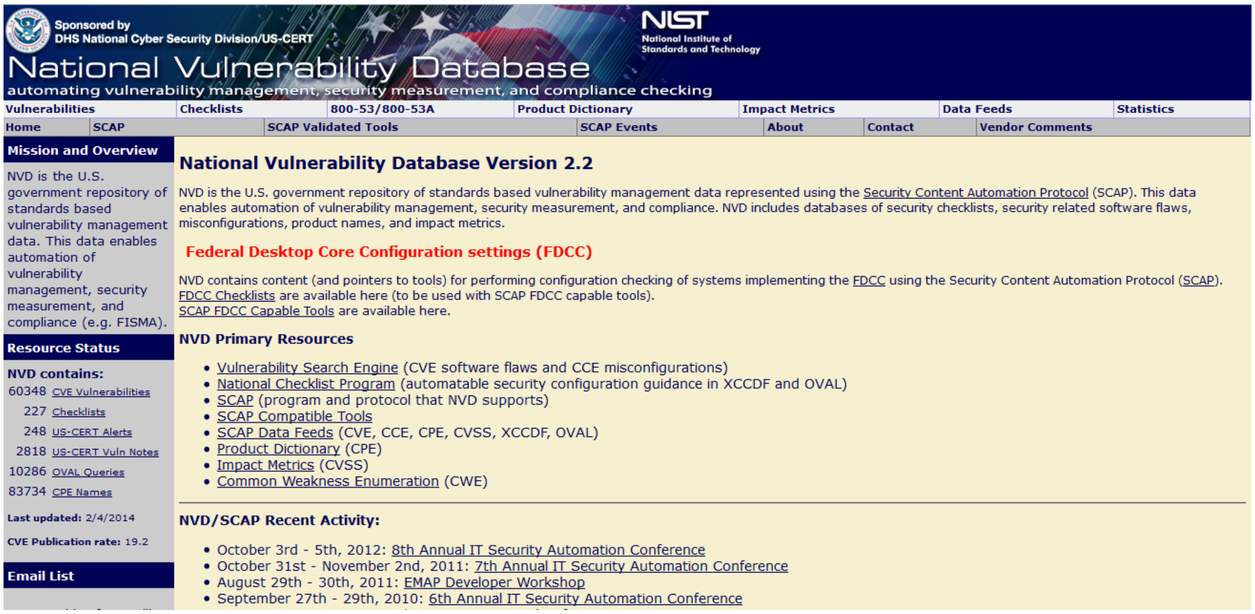
Impact. These attackers will continue to use any and all techniques at their disposal, and will not stop until they reach their objective. APTs often use zero-day exploits that do not have associated signatures in protective anti-virus software.

Takeaway. The best approach is using a defense-in-depth strategy, characterized but not limited to insuring both the networks and the individual computers are protected, and keeping every computers OS and applications up-to-date. These attacks are persistent and defense against them requires vigilance by scanning your internal and external network with sensors, and auditing activity logs. APTs are the worst type of cyberthreat because of the intensity and commitment of the attackers, who may be funded by governments with competing interests [9].

Relevance to C2. An APT is a major offensive weapon because of the resources that are committed to disrupt a target's C2 systems, and the target's critical infrastructure Although a difficult concept, we may have to accept that dedicated attackers will get into C2 nodes, and approaches will need to reduce their dwell time in the system, while denying them outgoing communications [2].

Increasing Cyberthreat Awareness

There are several ways to increase awareness of the cybersecurity threats, and current vulnerabilities. For example, vulnerability information can be found at National Vulnerability Database (NVD) [11] (Figure 2) and Symantec Connect [13] (Figure 3). Vulnerability research can be found at Hackerstorm [7]. Several of these websites also have mailing lists to push timely information to subscribers.


Sponsored by
DHS National Cyber Security Division/US-CERT
NIST
National Institute of Standards and Technology
National Vulnerability Database
automating vulnerability management, security measurement, and compliance checking
Vulnerabilities
Checklists
800-53/800-53A
Product Dictionary
Impact Metrics
Data Feeds
Statistics
Home
SCAP
SCAP Validated Tools
SCAP Events
About
Contact
Vendor Comments
Email List

Mission and Overview

NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance (e.g. FISMA).

Resource Status

NVD contains:

60348	CVE Vulnerabilities
227	Checklists
248	US-CERT Alerts
2818	US-CERT Vuln Notes
10286	OVAL Queries
83734	CPE Names

Last updated: 2/4/2014

CVE Publication rate: 19.2

National Vulnerability Database Version 2.2

NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics.

Federal Desktop Core Configuration settings (FDCC)

NVD contains content (and pointers to tools) for performing configuration checking of systems implementing the FDCC using the Security Content Automation Protocol (SCAP).
FDCC Checklists are available here (to be used with SCAP FDCC capable tools).
SCAP FDCC Capable Tools are available here.

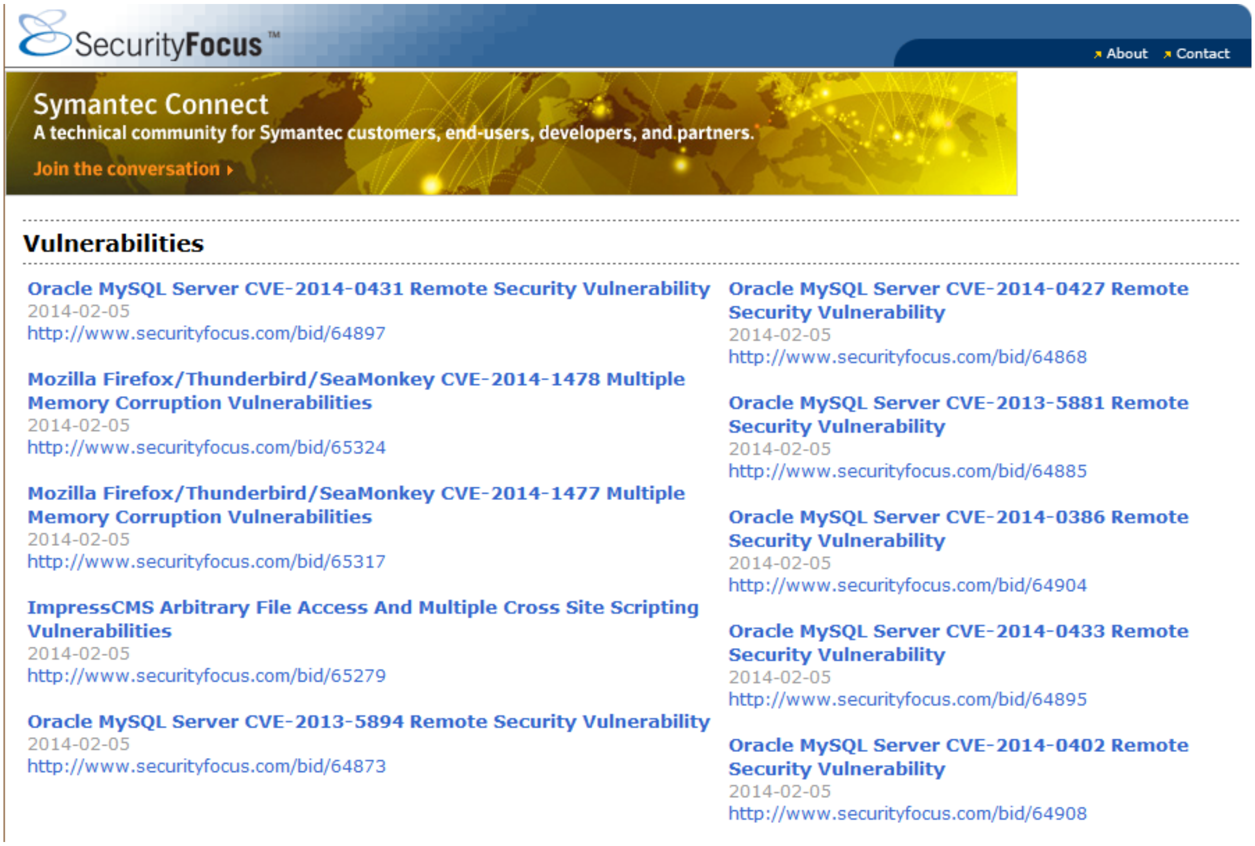
NVD Primary Resources

- Vulnerability Search Engine (CVE software flaws and CCE misconfigurations)
- National Checklist Program (automatable security configuration guidance in XCCDF and OVAL)
- SCAP (program and protocol that NVD supports)
- SCAP Compatible Tools
- SCAP Data Feeds (CVE, CCE, CPE, CVSS, XCCDF, OVAL)
- Product Dictionary (CPE)
- Impact Metrics (CVSS)
- Common Weakness Enumeration (CWE)

NVD/SCAP Recent Activity:

- October 3rd - 5th, 2012: 8th Annual IT Security Automation Conference
- October 31st - November 2nd, 2011: 7th Annual IT Security Automation Conference
- August 29th - 30th, 2011: EMAP Developer Workshop
- September 27th - 29th, 2010: 6th Annual IT Security Automation Conference

Figure 2. The National Vulnerability Database Homepage.


SecurityFocus™
About
Contact
Symantec Connect
A technical community for Symantec customers, end-users, developers, and partners.
Join the conversation ›

Vulnerabilities

Oracle MySQL Server CVE-2014-0431 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64897

Mozilla Firefox/Thunderbird/SeaMonkey CVE-2014-1478 Multiple Memory Corruption Vulnerabilities
2014-02-05
http://www.securityfocus.com/bid/65324

Mozilla Firefox/Thunderbird/SeaMonkey CVE-2014-1477 Multiple Memory Corruption Vulnerabilities
2014-02-05
http://www.securityfocus.com/bid/65317

ImpressCMS Arbitrary File Access And Multiple Cross Site Scripting Vulnerabilities
2014-02-05
http://www.securityfocus.com/bid/65279

Oracle MySQL Server CVE-2013-5894 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64873

Oracle MySQL Server CVE-2014-0427 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64868

Oracle MySQL Server CVE-2013-5881 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64885

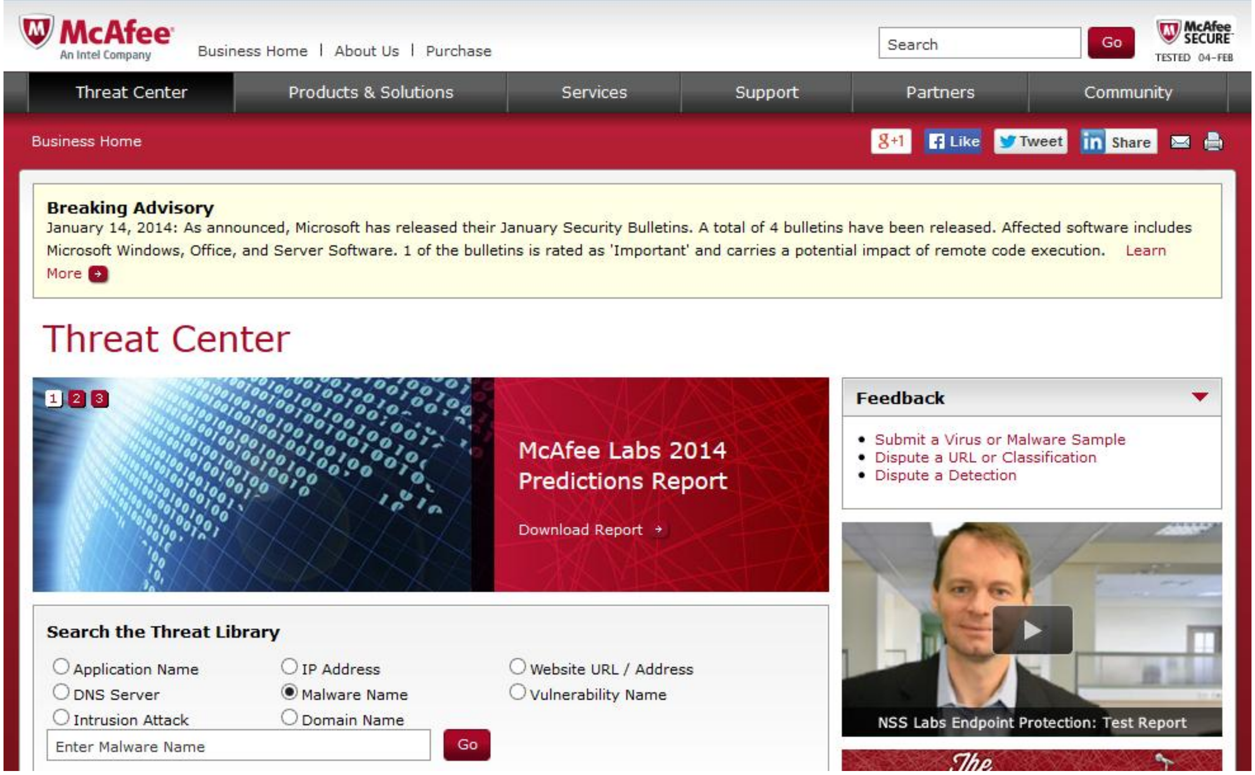
Oracle MySQL Server CVE-2014-0386 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64904

Oracle MySQL Server CVE-2014-0433 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64895

Oracle MySQL Server CVE-2014-0402 Remote Security Vulnerability
2014-02-05
http://www.securityfocus.com/bid/64908

Figure 3. The Security Focus Homepage.

There are several websites that can increase awareness of the state of cyberthreats. One website that provides access to a threat library is McAfee's Threat Center [10] (Figure 4). To learn more about a specific virus or other piece of malware, enter its name in the search box. The information in the database includes the malware's characteristics, method of infection, removal, and variants. It also shows the area of the world where the malware is currently proliferating.



Breaking Advisory

January 14, 2014: As announced, Microsoft has released their January Security Bulletins. A total of 4 bulletins have been released. Affected software includes Microsoft Windows, Office, and Server Software. 1 of the bulletins is rated as 'Important' and carries a potential impact of remote code execution. Learn More

Threat Center

1 2 3
McAfee Labs 2014 Predictions Report
Download Report
Feedback
Submit a Virus or Malware Sample
Dispute a URL or Classification
Dispute a Detection
NSS Labs Endpoint Protection: Test Report

Search the Threat Library

○ Application Name ○ IP Address ○ Website URL / Address
○ DNS Server ● Malware Name ○ Vulnerability Name
○ Intrusion Attack ○ Domain Name

Enter Malware Name

Figure 4. The McAfee Threat Center.

Education is another key piece of expanding awareness and understanding of cyberthreats. To gain depth in understanding malware, there are classes to explore the network delivery methods and the behavior of infected computers. The study of malicious network traffic involves first determining if there has been abnormal network behavior, then working to isolate any malware transmitted over the network. Another area to study to increase fluency in cyberthreats is basic malware analysis, where qualities of different types of malicious software and how it affects an infected computer's files and registry, and the network and process activity is explored.

Conclusions

Users are always the first line of in defending against cybersecurity threats, and not enough can be said about continuing education, and an organization that takes security seriously. By classifying and characterizing malware by type(s), security professionals can more easily describe specific behaviors and determine effective methods to defeat the threat.

Every threat that has ever been conceived is still out there. It can be slightly changed to create a new signature, or combined with other threats for more complex attacks. Worse, the malware is available in kits, similar to the ones software developers use, putting them in the hands of anyone with a desire to attack or experiment.

Malware can be identified by anti-virus (AV) tools only if it is a known form, and its signature exists in the AV databases. AV cannot detect zero-day threats, which are those previous unknown exploits of vulnerabilities; since they are unknown, no counter to them exists.

References

[1] Raed Albuliwi, ANRC Advanced Persistent Threat (APT) Whitepaper, NG Security Summit, Sep 2012, accessed 27 Jan 2014 retrieved from www.ngsecurityeu.com/media/whitepapers/2012/ANRC_AdvancedPersistentThreats.pdf

[2] Brown, Jeff. “A National Model for Cyber Protection Through Disrupting Attacker Command And Control Channels” accessed 8 Apr 2014 retrieved from http://www.whitehouse.gov/files/documents/cyber/ISA%20-%20A%20National%20Model%20for%20Cyber%20Protection%20Through%20Disrupting%20Attacker%20Command%20and%20Control%20Channels.pdf

[3] Burton, Kelly. “The Conficker Worm”, accessed 8 April 2014 retrieved from http://www.sans.org/security-resources/malwarefaq/conficker-worm.php

[4] Carpenter, Dan. An Approach to Command and Control Using Emerging Technologies. 18th International Command & Control Research & Technology Symposium (ICCRTS), 19-21 June, 2013.

[5] Cranor, Lorrie Faith “Can Phishing Be Foiled?”, Scientific American; Dec2008, Vol. 299 Issue 6, p104-110.

[6] Dittrich, David, and Sven Dietrich. "New directions in peer-to-peer malware", Sarnoff Symposium, 2008 IEEE, pp. 1-5. IEEE, 2008.

[7] Hackerstorm, n.d., accessed http://www.hackerstorm.co.uk

[8] Incapsula “Report: Bot Traffic is up to 61.5% of all website traffic”, The Incapsula Blog, 9 Dec 2013, accessed 24 Jan 2014, retrieved from http://www.incapsula.com/the-incapsula-blog/item/820-bot-traffic-report-2013

[9] Mandiant, APT1, n.d., accessed 27 Jan 2014, retrieved from http://intelreport.madiant.com/Mandiant_APT1_Report.pdf.

[10] McAfee Threat Center, n.d., accessed 27 Jan 2014, retrieved from http://www.mcafee.com/us/threat-center.aspx

[11] NIST, National Vulnerability Database, v2.2, n.d., accessed 4 Feb 2014, retrieved from http://nvd.nist.gov/

[12] New York Times, “No Extra Jail Time For Man Sentenced In Melissa Virus”, 4 May 2002.

[13] Connect, n.d. accessed 4 Feb 2014, retrieved from www.securityfocus.com.

[14] Sikorski, Michael, and Andrew Honig. Practical Malware Analysis, No Starch Press, 2012.

[15] Spafford, Eugene H. The Internet Worm Program: An Analysis, Purdue Technical Report CSD-TR-823, 1988, retrieved from http://spaf.cerias.purdue.edu/tech-reps/823.pdf

[16] Swain, Nathan. NMAP Scanning: How a Simple Tool STILL Makes Dramatic Impact, Hackin9, Vol. 8, No. 4 Issue 4/2103 (64), pp 20-25.