Global Information Assurance Certification Paper

Classic Attacks
Lessons from the Past

GSEC Practical Version 1.4b Option 1

By Brian Johnson

7/12/04

Abstract

Virus attacks seem to come out of nowhere. Systems fail and resources become unavailable. Business processes come to a halt and revenue slows. The Executives want answers, and System Administrators find themselves scrambling to fix the problem. Many times the situation is avoidable, but systems aren't patched and signature files are out of date. This situation is all too familiar. After two decades of fighting viruses and other attacks, we still find ourselves vulnerable to these threats? Why are we always playing catch-up?

This paper takes a close look at four classic virus attacks. By learning how these attacks work, we can better prepare ourselves for the future. This will be done by dissecting each attack and looking at its important attributes. Key areas of concern are the vulnerabilities exploited, the propagation methods, and the payloads released. Based on the threats of each attack, defense strategies will be discussed and recommended.

Terminology

The tech industry has transformed the term "virus" into a generic phrase, basically meaning any malicious code that attacks computer systems. In part, this is due to the hybrid nature of many attacks. This paper will follow the industry's lead, and use the term "virus" loosely when discussing attacks in general. However, when discussing specific threats they will be identified in true form as viruses, trojans, worms, etc.

History

In 1984 Fred Cohen wrote a research paper in which he described self-propagating programs. He later coined the termed "Virus." Three years after his initial research, Cohen requested funding from the National Science Foundation so that he could research countermeasures. He was denied funding because the topic was not relevant to current interest.[1]

It wasn't long after Cohen's research that we started seeing viruses in the wild. When viruses first emerged they relied on floppy disks and tapes to propagate. The Brain and Stoned viruses were two boot sector viruses that relied on such techniques. Early viruses were more like teenage pranks, more annoying than destructive. At the time, a simple desktop virus scanner was considered sufficient protection.

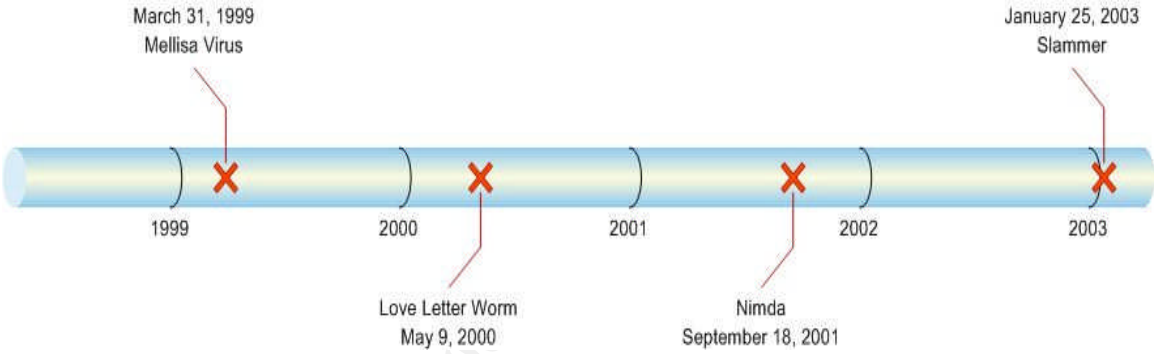
In 1998, Robert Morris Jr. created a first-of-its-kind worm. The worm, later to become known as the "Morris" worm, used ARPANET as its distribution channel.

[1] Lemos, Robert. "Decades after creation, viruses, defy cure." A 20-year Plague. 25 November 2003 URL: http://news.com.com/2009-7349_3-5111410.html (May 2004).

It is estimated that 10% of the then 88,000 systems on ARPANET were infected.[2] Stories differ on his intent, but one thing is for sure. A new breed of virus had arrived that propagated faster and further than any virus before, thus setting the stage for viruses and worms to come.

4 Classic Attacks

The following are four classic examples of malicious attacks. The threats range from viruses and worms to hybrid techniques. Systems are compromised by utilizing macros, scripting languages, buffer overflows and unneeded services. These attacks show the many ways in which malicious code can take advantage of vulnerable systems.



Melissa

Melissa was a macro virus that infected Microsoft Word documents. The virus propagated rapidly due to its use of e-mail. Replication via e-mail was only successful if Outlook was installed on the system. Messages appeared in users mailboxes with a subject of "Important Message from *[User Name]*." The message contained an attached file that, when opened, would release the payload. By reading the Outlook address book, the virus would find new victims to which the infected attachment would be sent. In addition, the virus altered the normal.dat file, the default template used by Word. This caused all newly created documents to be infected.

The CERT advisory for Melissa pointed out that "human action (in the form of a user opening an infected Word document) is required for the virus to propagate."[3] Today users know not to open attachments in e-mail. In part, this is due to

[2] "Security of the Internet" 1998. URL: http://www.cert.org/encyc_article/tocencyc.html (June 2004)
[3] "CERT Advisory CA-1999-04 Melissa Macro Virus." 27 March 1999. URL: http://www.cert.org/advisories/CA-1999-04.html (June 2004)

Melissa. Melissa was the first macro virus to have far reaching effects on both corporations and government agencies. At the time the Internet was booming and this was their first taste of malicious intent. For the first time the FBI National Infrastructure Protection Center (NIPC) issued a warning about a virus. The overall message was simple: "Do not open e-mail attachments."[4]

Melissa took advantage of a default configuration for an often unused feature. At the time all Microsoft Office applications automatically ran macros when a document was opened. Macro viruses were not new, but they had stayed under the radar without notice until the introduction of Melissa. Macro viruses are not limited to Microsoft Office, but to all products with macro capabilities. By changing the macro security level to high or medium, the threats of a macro virus can greatly be reduced.

Love Letter

Love Letter, also known as Love Bug, was a VBScript with both virus and worm characteristics. It used Outlook to propagate and VBScript to release its pay load. Love Letter arrived via e-mail as an attachment named LOVE-LETTER-FOR-YOU.TXT.vbs and a subject of "I love you." Once executed, it would search out and infect files by inserting its own code. It would then rename the file with a .vbs extension.

A critical attribute of this threat is the ".TXT.vbs" double extension of the attachment. In Windows operating systems the last extension is the one that Windows will act on. By default Windows hides file extensions of known types. By hiding the true .vbs nature of the attachment, it appeared as a harmless .txt file.

Windows 9x and NT 4.0 were not vulnerable to Love Letter because no scripting engine was available to execute the code. WSH, the Windows Scripting Host program, was first included with Windows 2000. It provided a means for executing different scripting languages. WSH should be turned off on systems that do not need scripting.

The most interesting attribute of Love Letter is its use of social engineering. Social engineering often conjures up visions of human-based attacks where hackers con unsuspecting users out of their passwords. Love Letter is an example of computer-based social engineering. Kevin Mitnick, among the most notorious social engineers to date, calls it "Cracking the human firewall."[5] Love Letter does this by earning our trust and peaking our interest. It earns trust

[4] "FBI Warns of Melissa Virus." Wired News. 20 March 1999. URL: http://www.wired.com/news/technology/0,1282,18790,00.html (June 2004)
[5] Mitnick, Kevin and Simon, William L. Art of Deception: Controlling the Human Element of Security. Indianapolis: Wiley Publishing, Inc, 2002. 4.

because it comes from a known sender. Its subject, “I Love You,” peaks interest. The irony is this e-mail comes from the people least likely to say “I Love You:” our bosses, business associates, and beer drinking buddies, but yet the recipient cannot resist the urge to open the attachment.

Nimda

Nimda is a classic example of a hybrid attack that used a combination of worms, trojans, and backdoors to accomplish its misdeeds. Like many viruses, Nimda used e-mail to propagate. Unlike other viruses which used Outlook, Nimda installed its own SMTP server. When an Outlook client received the malicious code it would auto execute due to a known vulnerability. Once executed, it would create trojan horse copies of itself by infecting existing .exe files on the computer. In addition, this worm installed a TFTP server which it used for propagation. It also attempted to use back doors left behind by Code Red.

Unpatched systems greatly increased the effects of Nimda. Nimda took advantage of a known flaw in Outlook that allowed for the automatic execution of embedded mime attachments. It also used a known directory traversal tactic to compromise IIS web servers. Many of these IIS servers had already been compromised by Code Red. Nimda used to its advantage backdoors that had been left behind by Code Red. This leads to the question: Why had administrators not cleaned and patched their IIS servers?

Nimda propagated through e-mail as an attachment named readme.exe. If executable attachment blocking had been in effect, one of Nimda’s propagation methods would have been shutdown. An effective practice is to block .scr, .bat, .pif, .vbs, .com, .exe and any other executable file attachments. Very few businesses have a legitimate business reason to receive these types of files through e-mail.

The effects of Nimda were so great that it often took a complete rebuild of the system to safely recover.

Slammer

Slammer hit like a thief in the night compromising most of the Internet in a matter of minutes. According to CAIDA, the Cooperative Association for Internet Data Analysis, slammer compromised 90% of its victims in 10 minutes.[6] Using port 1434 UDP, it took advantage of a buffer overflow in Microsoft SQL and MSDE. Once a system was compromised, it would then continuously search for vulnerable systems which resulted in a denial of service.

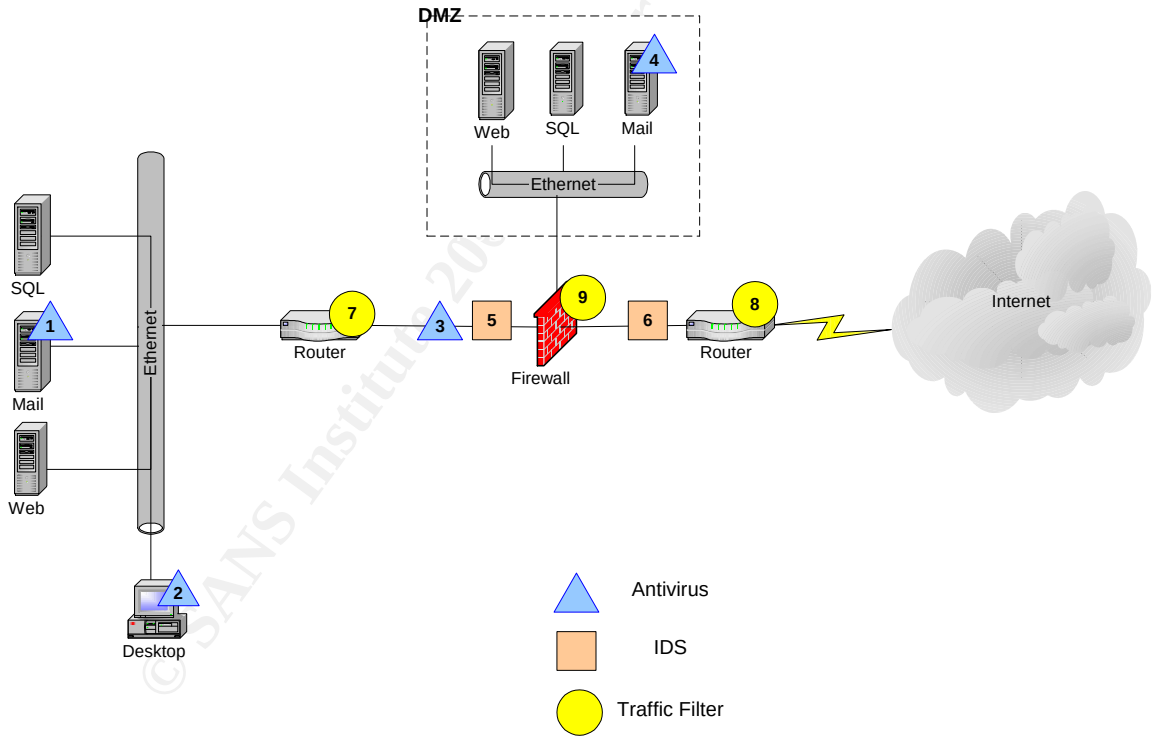
[6] More, Paxson, Savage, Shannon, Staniford, and Weaver. “Spread of the Sapphire/Slammer Worm.” URL: http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html (June 2004).

Most network administrators became aware of Slammer because of a sudden decrease in network performance. Before it became obvious why this traffic was being generated, administrators looked to their firewalls, routers, and other perimeter devices to block the traffic. This gave them time to hunt down the root of the problem[7].

Slammer is a prime example of an event that could have been avoided. A patch that fixed the vulnerabilities used by Slammer had been available for five months. Microsoft Bulletin MS02-039 had been available since July 24, 2002. It prescribed installing a patch to fix three vulnerabilities that could ultimately lead to an attacker gaining control of an affected system. [8]

Defending the Attack

A number of defense mechanisms exist that can protect against these threats. The following diagram demonstrates three popular methods: antivirus, IDS, and traffic filtering. It shows strategic points of placement within a typical network. By using multiple methods of defense we are providing a layered strategy.



[7] Shah, Sam. "How we slammed the door on Slammer." 11 Mar 2003. URL: http://techrepublic.com.com/5100-6264-1058233.html (May 2004).
[8] "Microsoft Security Bulletin MS02-039. 31 January 2003. URL: http://www.microsoft.com/technet/security/bulletin/MS02-039.mspx (June 2004).

Antivirus

At the heart of every defense strategy we find antivirus software. Our example network shows three primary points of placement: at the mail server, on the desktop, and at the gateway. Multiple layers are in place to provide maximum protection. If one layer fails, another will succeed.

The desktop scanner (object 2) will detect infected files as they are executed. Successful detection will clean, quarantine, or delete the file depending on the configuration. Many antivirus products hook into various e-mail clients to provide extra protection by scanning e-mail attachments as they arrive. Due to the growing trend of html based e-mails, scanners now look for malicious Active X controls that are embedded in the body of the message.

At the mail server (object 1 and 4), e-mail will be scanned as it enters the system for processing. Most solutions will scan both attachments and content of messages. In addition, attachments of certain types can be blocked.

At the gateway (object 3) traffic will be scanned as it enters or leaves the network. This method is made up of both software and hardware solutions. Software can sit on a dedicated server or a firewall. Most antivirus vendors provide a plug-in for popular firewalls such as Checkpoint and Microsoft ISA. Hardware solutions are appliance type devices that will include one of the popular vendors' software embedded into the system. One benefit of a hardware solution is the speed at which it scans.

IDS (Intrusion Detection Systems)

IDS devices monitor traffic for unauthorized access of a network. Two types of IDS are network and host based systems. Network-based solutions monitor all traffic as it passes a point on the network. Host-based solutions are only concerned with traffic that enters or leaves a specified host. It is not uncommon to see both methods implemented in an IDS strategy.

Our example shows the placement of two network IDS points (objects 5 and 6.) These can be hardware based, software based, or a combination of both. The placement of IDS devices will greatly depend on the design of the network. The ideal place to scan is at the entry/exit point of network segments.

IDS function in a manner similar to the signature-based characteristics of antivirus software. IDS compare traffic to a known set of attack signatures. Both Nimda and Slammer left identifiable patterns that would have been picked up by an IDS system. The first patterns were related to the back doors left behind by Code Red. The second, unique to Nimda, was a result of the directory traversal.

```
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
```

```
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

CERT® Advisory CA-2001-26 Nimda Worm
http://www.cert.org/advisories/CA-2001-26.html

These known attack signatures when entering the network should alert the administrator that the network is under attack. If these signatures are seen leaving the network, this indicates the systems have been compromised and are now doing the attacking.

In addition, IDS can be used for recognizing reconnaissance scans, enforce policy, and map normal traffic patterns. Using the example network, let's assume that the mail server in the DMZ is the only system authorized to send and receive e-mail to the Internet. Nimda installed an SMTP server on each infected host which would then produce port 25 traffic. A properly configure IDS would recognize the unauthorized traffic.

Traffic Filtering

Traffic filtering involves the filtering of traffic as it enters or leaves different network segments. Ingress filtering is applied as traffic enters a network while egress filtering is applied as traffic leaves a network. The most logical points for filtration are the firewall and the routers. Routers are devices that operate at layer three, the network layer. Filtering is based on header information. This comprises the source and destination addresses along with the service ports. This type of packet filter is fast, but limited in scope. A more advanced filtering technique performed on firewalls is stateful inspection. Stateful inspection delves deeper into the packet, beyond the header. Inspection can be performed all the way to the application layer. This deeper inspection allows for filtering decisions to be made on both content and context.[9] This makes for slower, but more powerful filtering.

Lessons Learned

In many environments the first line of defense to threats is Antivirus software. The problem with antivirus software is its reactive nature. The attacks are

[9] Welch-Abernathy, Demeon D. Essential Check Point Firewall-1. Indianapolis: Addison-Wesley, 2001. 4.

thwarted only when there is a known signature. Systems were only protected from Melissa and Love Letter after the antivirus vendors were able to properly identify its signature. This demonstrates the need to update signature files daily, if not hourly. Look for products that deploy signatures from a central console. Mobile workforces are expanding the perimeters of our networks to limitless boundaries. Extra measures need to be in place to ensure protection when disconnected from the corporate environment.

IDS, like antivirus, are reactive in nature. They are only successful when there is a known traffic pattern. Nimda left two identifiable signatures. One generated as it tried to access the backdoors left behind by Code Red. The second, unique to Nimda, was created by an IIS directory traversal. Administrators often don't know where to monitor or how to properly configure their IDS. In many cases the attack signatures left by Nimda and Slammer were hidden by large numbers of false positive alerts. False positives are common among IDSs after a default installation. It's imperative that administrators configure these systems to match the unique needs of their environment.

Administrators need to find vulnerabilities before the bad guys do. Commercial scanners like Retina Network Security Scanner, or free scanners, such as Nessus, can help discover what services systems are running. In the Nimda and Slammer attacks, administrators found they were running SQL and IIS servers of which they were unaware. MSDE, also vulnerable, is installed by many third party solutions. IIS was preinstalled on Windows 2000 systems. Had these unnecessary services been turned off the effects of Nimda and Slammer would have been lowered dramatically.

The Melissa, Love Letter and Nimda attacks all benefited from user interaction. It is important that users be educated on the destructive nature of viruses and the means by which they propagate. The user needs to know the measures that are in place and their roles in the protective process. This can be accomplished with policy and procedure statements. A sample statement might read, "It is the company's policy to use XYZ antivirus software. The end user is not to disable or tamper with the scanner in anyway. Please report suspicious activity to helpdesk@companyXYZ.com.[10]" For additional help on developing a policy, consult the SANS "Guidelines for Anti-Virus Process" at http://www.sans.org/resources/policies/Anti-virus_Guidelines.pdf and review their sample policy at http://www.sans.org/resources/policies/Lab_Anti-Virus_Policy.pdf.

From the outset, applications are flawed. It is the administrator's responsibility to make them more secure. Applications come with powerfull features that are not used and operating systems included unnecessary services. Melissa and Love Letter both took advantage of default configurations that were less than

[10] Walton, Mike. "Antivirus policies should educate users and outline responsibilities" 15 Apr 2002. URL: http://techrepublic.com.com/5100-6263-1043860.html (May 2004).

desirable. Nimda attacked IIS servers. Slammer attacked SQL servers. All four of our classic attacks damaged the integrity of data files or services provided by the compromised systems. The use of change monitoring systems, like Tripwire, can ensure the integrity of systems.

Administrators need to implement better ingress and egress filtering. It is all too common to have a hard perimeter blocking everything only to leave the internal network unmonitored. In addition, ingress filtering is often rigid while egress filtering is left non existent, allowing all traffic out. The problem often exists because administrators don't know the traffic patterns of their networks. Slammer used port 1434 UDP. A port used only by SQL servers. Using the example network, assume the SQL server located in the DMZ is the only SQL server that talks directly to the Internet. A properly configured ingress filter would prevent unauthorized port 1434 traffic from entering our network. If the internal SQL server became infected with Slammer, it would start sending 1434 UDP traffic to random IP addresses on the Internet. Knowing that our internal SQL server does not talk to the Internet, a properly configured egress filter would stop further propagation of Slammer.

Companies need to develop more rigid patching habits. Both Nimda and Slammer took advantage of know vulnerabilities that had patches readily available. By properly patching these systems, risk is dramatically reduced. You can learn about vulnerabilities by contacting your vendor directly, or by checking their website. Most vendors now have a dedicated section to security. You can also sign up for newsletters like @Risk from sans.org or moderated mailing lists like Bugtraq at securityfocus.com. We are starting to see an increase in vendors providing built-in mechanisms for updating and patching systems. Linux distributions like Redhat, provide RHN, while Microsoft provides SUS for Windows systems. There are also many 3rd party products that fill in the void.

There is a need for developing proactive security software. Many of the current products are reactive in nature. This is most often seen with our antivirus and IDS systems. Antivirus systems need to use heuristic and behavioral algorithms so that new virus mutations will be detected. Administrators need to monitor network traffic so that when the abnormal happens IDS systems recognize the patterns and take protective measures. Fortunately, we are starting to see this as IDS systems evolve into Intrusion Prevention systems (IPS) that not only detect, but take a more active role in blocking the threat.

That which is designed for good, too often, is used for bad. In the Love Letter example, VBS code was used maliciously. VBS in combination with WSH was intended to bring a much needed functionality to the Windows environment. The idea was to give administrators the scripting power that had been available in UNIX for years. Unfortunately, this gave virus writers a new medium to use as well.

Bruce Schneier, acclaimed author and CTO of CounterPane, calls security "a process, not a product." Often we purchase a suite of products, install it, and think we're protected.[11] Network administrators must go beyond the product and create policies, procedures, and strategies to increase the effectiveness of the product. The individual pieces must compliment one another to create a continuous operation. If one piece fails, the operational process must continue. There is always room for improvement and we should continually be educating ourselves about new threats. Historically our security measures have been reactive. Many of the tactics used in the four classic attacks discussed are still used today. We still see e-mail attachments used as a propagation method, macro viruses, buffer overflows, and unneeded services compromised by the latest malicious code circumventing the Internet. Only diligence in our preparation to confront the daily threat of these attacks will allow us to capitalize on what we've learned from the past to prepare for a more secure future.

[11] Schneier, Bruce. Secrets & lies. New York: John Wiley & Sons, Inc, 2000. 84.

Works Cited

"CERT/CC. Advisory CA-1999-04 Melissa Macro Virus." 27 March 1999. URL: http://www.cert.org/advisories/CA-1999-04.html (June 2004)

"CERT/CC. Advisory CA-2001-26 Nimda Worm. 18 September 2001. URL: http://www.cert.org/advisories/CA-2001-26.html (May 2004)

CERT/CC. "Security of the Internet" 1997. URL: http://www.cert.org/encyc_article/tocencyc.html (June 2004)

Lemos, Robert. "Decades after creation, viruses, defy cure." A 20-year Plague. 25 November 2003 URL: http://news.com.com/2009-7349_3-5111410.html (May 2004).

Microsoft. "Microsoft Security Bulletin MS02-039. 31 January 2003. URL: http://www.microsoft.com/technet/security/bulletin/MS02-039.mspx (June 2004).

Mitnick, Kevin and Simon, William L. Art of Deception: Controlling the Human Element of Security. Indianapolis: Wiley Publishing, Inc, 2002. 4.

More, Paxson, Savage, Shannon, Staniford, and Weaver. "Spread of the Sapphire/Slammer Worm." URL: http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html. (June 2004).

Schneier, Bruce. Secrets & lies. New York: John Wiley & Sons, Inc, 2000. 84.

Shah, Sam. "How we slammed the door on Slammer." 11 Mar 2003. URL: http://techrepublic.com.com/5100-6264-1058233.html (May 2004).

Walton, Mike. "Antivirus policies should educate users and outline responsibilities" 15 May 2002. URL: http://techrepublic.com.com/5100-6263-1043860.html (May 2004)

Welch-Abernathy, Dameon D. Essential Check Point Firewall-1. Indianapolis: Addison-Wesley, 2001. 4.

Wired. "FBI Warns of Melissa Virus." 20 March 1999. URL: http://www.wired.com/news/technology/0,1282,18790,00.html (June 2004)

Last Updated: July 27th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event