Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

The Code Red Worm

This paper focuses on the behavior of the Code Red (CRv2) and Code Red II worms, discussing how the worms propagated and capitalized on known yet unpatched software vulnerabilities. Creating a computer and network security policy is discussed in light of the lessons learned from these worms.

Copyright SANS Institute
Author Retains Full Rights

AD

John C. Dolak
Security Essentials
Version 1.2e
The Code Red Worm

Introduction

On July 12, 2001, a new worm began propagating across the internet. Although the worm did not yet have a name, it was the first incarnation of what was to become known as the "Code Red" worm [1]. This initial version of the worm is commonly referred to as CRv1. On July 19, another variant of the worm, which shared nearly all its code with the first version of the worm, began to spread even more rapidly than its predecessor a week before. The new variant of the Code Red worm was reported to have infected more than 250,000 systems in just nine hours [3]. This variant of the worm is now commonly referred to as CRv2.

The worm scanned the internet, identified vulnerable systems and infected these systems by installing itself. The rate of scanning grew rapidly because each newly installed worm joined others already in existence. Not only did the worm result in defaced web pages on the systems it infected, but its uncontrolled growth in scanning resulted in a decrease of speed across the internet—a denial of service attack—and led to widespread outages among all types of systems, not just the Microsoft Internet Information Server (IIS) systems it infected directly.

On August 4, a new worm exploited the same vulnerability in Microsoft IIS web server as the original Code Red worm [2]. Even though it shared almost no code with the first two versions of the original worm, it was named Code Red II simply because it contained the name in its source code and exploited the same vulnerability in the IIS indexing service. In addition to the original Code Red and the Code Red II worms, there are other possible variants of the worm. However, this paper will focus on the Code Red (CRv2) and Code Red II worms.

Code Red's Affect in Both Private Industry and the Government

As a result of the Code Red worm's rapid spread across the internet, businesses and individuals worldwide experienced disruptions of their internet service. Qwest, the Denver-based telecommunications corporation, which provides DSL services to approximately 360,000 customers throughout the western and midwestern U.S., is being asked to refund fees to customers as a result of service interruptions due to the denial of service caused by the Code Red worm. In addition, the Washington state Attorney General has asked Qwest to pay these customers, some of whom claim the outage cost them thousands of dollars in lost sales. However, Qwest says it has no plans at this time to credit customers who were afflicted by the Code Red worm [15].

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

As reported by the press, the U.S. Department of Defense experienced slightly different but equally disruptive problems as a result of the Code Red Worm [7]. A specific denial of service attack built into the worm prompted the White House to change its IP address. In addition, Code Red Worm's resulting denial of service led DOD to block TCP port 80 traffic originating from non .mil networks destined for the Pentagon and other DOD networks. As a result, non-DOD customers attempting to access DOD web sites were blocked or experienced a severe degradation of service when trying to access government sites. This resulted in numerous customer complaints ranging from the inability to bid on government contracts to difficulties in accessing personnel web sites to apply for government jobs—some complaints even escalating to the congressional level.

Could these attacks and others like them have been prevented? The answer is yes--and perhaps no. Although Microsoft made a patch available nearly a month before the initial outbreak of the Code Red worm, the massive number of infected systems demonstrates the ongoing problem of the failure of system administrators to keep their systems up-to-date with the most recent security patches.

Microsoft's Warning

On June 18, Microsoft released their Security Bulletin MS01-033, Unchecked Buffer in Index Server ISAPI Extensions Could Enable Web Server Compromise [13]. Microsoft recommended that all system administrators of IIS web servers running on Windows NT 4.0 or Windows 2000 immediately patch their systems to prevent a vulnerability that could give an attacker complete control over the server and allow him or her to run any code of choice. An unchecked buffer can lead to a buffer overflow, which occurs when a program or process tries to store more data in a temporary data storage area than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information can overflow into adjacent buffers, corrupting or overwriting the valid data they hold [17].

Several ISAPI extensions are included in the default installation of IIS. An ISAPI, or Internet Services Application Program Interface, extension is a dll (Dynamic Link Library) that provides a set of web functions beyond those natively provided by IIS. The security vulnerability is the result of an unchecked buffer in the idq.dll component of Index Server, or Indexing Service as it is called in Windows 2000. The idq.dll provides support for administrative scripts (.ida files) and Internet Data Queries (.idq files). The unchecked buffer resides in a section of the code that handles input URLs. Potentially, this unchecked buffer could allow an attacker to conduct a buffer overflow attack on a server with the idq.dll installed, and consequently execute his or her choice of code on the vulnerable web server.

Because idq.dll runs in the system context, such an attack could allow the attacker to take complete control over the system and run any action on it. This exploit does not even require that the Index Server/Indexing Service be running, but simply that the

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

script mapping for .idq or .ida files be present. This is true because the buffer overflow occurs prior to any actual request for indexing functionality.

This vulnerability is only exploited if a web session can be established with a vulnerable server. Systems that have Index Server or Indexing Service installed, but not IIS, are not at risk. Default installations of Windows 2000 Server are vulnerable, because IIS 5.0 installs by default as part of Windows 2000 Server products, and idq.dll is installed as part of the IIS 5.0 installation. Default installations of Windows 2000 Professional are not vulnerable because IIS does not install by default. A default installation of Windows NT 4.0 is not vulnerable because IIS 4.0 does not install by default. Any system without IIS is not vulnerable, nor are any flavor of Unix systems [13].

Analysis by eEye Digital Security

The IIS index server vulnerability was first discovered by eEye Digital Security and later announced by Microsoft on June 18. eEye released Advisory AL20010618, All versions of Microsoft Internet Information Services Remote buffer overflow (SYSTEM Level Access) [10] in conjunction with Microsoft's Security Bulletin MS01-033.

On Friday July 13, eEye was contacted by two network administrators who were experiencing significant attacks targeting the index service vulnerability. After reviewing the logs forwarded by the administrators, eEye analysts determined that someone had released a worm into the internet that was now spreading rapidly through Microsoft IIS web servers.

According to eEye's in-depth analysis of the Code Red Worm, the eEye analysts designated the worm the .ida "Code Red" worm in part because the worm is designed to deface web pages with the text "Hacked by Chinese" (hence the reference to "red"), and also because "code red mountain dew was the only thing that kept us awake all night to be able to disassemble this exploit even further [9]."

Based on eEye's analysis, the Code Red worm spreads through IIS systems on the internet by initially attacking one vulnerable system, then setting up one hundred threads of the worm on that system. The first 99 threads are then used to infect other vulnerable web servers. The worm spreads to other servers by using what first appears to be a series of randomly generated IP addresses. However, upon further analysis, the spreading mechanism is found to be, in fact, not random at all. Instead, the worm uses a static seed to generate a list of IP addresses to attack.

A static seed is a starting point for a random number generator [1]. A static seed causes a random number generator to output the same sequence of numbers each time the generator runs. So, even though the resulting numbers themselves have no predictable relationship to each other, they are the same numbers each time. Hence, they cannot be considered entirely random. As opposed to a static seed, a random seed uses an unpredictable starting point, so the sequence of numbers it generates are truly random and not at all predictable.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

As a result of using the static seed number generator, every infected computer will attempt to infect the same list of “random” IP addresses. Consequently, as the worm attempts to re-infect systems, it creates a denial of service affect due to the sheer amount of data transferred between the systems with IP addresses in the “random” sequence. Interestingly enough, had the worm used totally random IP address generation, it would have had the potential to infect a significantly higher number of systems in a much shorter period of time. eEye does not offer a definitive answer as to why the attacker chose the static random number generator. However, they do propose one possible reason: If the attacker’s own IP address was one of the first “few” (i.e. first 100 or 1000) IP addresses to be scanned in the known list of IP addresses, the attacker could then set up a sniffer and by logging all the attempted connections to TCP port 80 to their own IP. By doing so, the attacker would be able to compile a fairly comprehensive list of systems infected by the worm [9].

The 100th thread checks to determine if it is running on an English Windows NT or 2000 system. If it is, the worm defaces the existing web page by replacing it with a message stating “Welcome to http://www.worm.com!, Hacked By Chinese!” The hacked page will remain on the server for ten hours, then disappear completely unless the server is re-infected by the worm from another host. If the system is not running an English version of NT or 2000, the 100th thread will simply be used to infect other vulnerable systems. Each of the 99 or 100 worm threads then checks for a file called c:\notworm. If this file is found, the worm goes into a dormant state. If the file is not found, each thread continues to attempt to infect more systems.

Finally, the worm threads check each infected computers time. If the time is between 20:00 UTC and 23:59 UTC, the worm uses that thread to attack the web site www.whitehouse.gov by sending 100 kilobytes of data to TCP port 80. This results in a potential distributed denial of service on the site. If the time does not fall into this range, the thread simply attempts to find and attack other web servers. eEye estimated that worm had the potential of infecting about a half million IP addresses a day, and they considered this to be a low estimate. eEye also pointed out that when the worm does not execute properly, it will continue to spawn new threads until the infected machine crashes and must be rebooted.

eEye Digital Security Advisory AL20010717 provides an in-depth analysis of the Code Red worm functionality [9]. This analysis explains in detail how the worm code executes at each step of the infection.

Code Red II

On August 4, eEye Digital Security released another security advisory warning of another worm that that exploited the same IIS vulnerability as the original Code Red worm [8]. On August 6, the CERT Coordination Center released an incident note reiterating this warning [2]. The new worm exploited not only Windows NT 4.0 and 2000 systems with IIS installed, but also exploited another unrelated vulnerability in Cisco 600

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Series DSL routers. The new worm was dubbed Code Red II, but it was significantly different from the original Code Red worm.

The Code Red II worm causes system level compromise and leaves a backdoor on Windows 2000 machines running IIS 4.0 or 5.0 with indexing services installed. It also causes crashes to occur on vulnerable NT 4.0 systems running IIS. Finally, unpatched CISCO series DSL routers will process the HTTP request and exploit an other, unrelated vulnerability which will cause the router to stop forwarding packets. In other words, the Code Red II worm is able to exploit not only a Microsoft IIS vulnerability, but also a Cisco vulnerability.

The Code Red II worm attempts to connect to TCP port 80 on a randomly chosen host. When the worm finds a web server and makes a successful TCP port 80 connection, the attacking host sends a crafted HTTP GET request to the victim host in an attempt to exploit the buffer overflow vulnerability. The worm's self-propagating nature causes the same exploit to be sent to each of many randomly chosen hosts. If the exploit is successful, the worm will execute on the victim system. IP addresses to be scanned are determined in a probabilistic manner, such that there is a higher probability that a given thread will scan random IP addresses with the same first one or two bytes as the infected host.

When a system is compromised, the worm verifies the existence of the Code Red II atom. If found, this means the system is already infected, and the worm goes into a permanent sleep state. If not found, the atom is created and the infection process continues. As with the original Code Red worm, Code Red II checks the system language. If it is Chinese (Taiwanese) or Chinese (PRC), the worm spawns 600 threads which will scan the network for 48 hours. If the default language is not Chinese, the worm sets up 300 threads to scan for 24 hours. Unlike the original Code Red worm, the Code Red II worm does not deface the web page.

The worm then copies the %SYSTEM%\cmd.exe file to root.exe into a publicly accessible folder, potentially allowing an intruder to execute arbitrary commands on the victim system. Finally, the worm creates a copy of explorer.exe and places it into C:\ and D:\. The trojan explorer.exe is able to use the real explorer.exe to mask itself, thereby creating a virtual mapping which exposes the C:\and D:\ drives.

To add to the confusion caused by Code Red II, questions have been raised not only by IT professionals but also by the media concerning the Code Red II infection of Microsoft Personal Web Services (PWS) running on Windows 2000 Professional systems [14]. This is purely a misconception, because PWS does not run on Windows 2000 Professional, but only on Windows 95/98/ME and Windows NT Workstation. The misunderstanding seems to have arisen because of confusing documentation in Windows 2000 Professional which refers to the integrated web server as Peer Web Services (also called PWS), rather than by its correct name of IIS 5.0. While Personal Web Server is not vulnerable, a web server in a Windows 2000 Professional system is vulnerable to both the Code Red and Code Red II worms and must be patched.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Although Personal Web Services is similar to IIS 5.0, it is not vulnerable to any variations of the Code Red worm.

How to Patch Your System

Any vulnerable systems should be patched immediately regardless of whether the system has been infected. Of course, in an ideal situation, the patch will be applied before a system gets infected. The following are Microsoft's recommended steps to determine whether or not a system is vulnerable, and if so, how to apply the patch [19].

First, determine if your system is vulnerable by following these steps:

Press Ctrl-Alt-Del and select Task Manager, select the Processes tab, and look in the Image Name column. If you see Inetinfo.exe, you are running IIS. If you are using Windows 95, Windows 98, Windows Me, Windows XP RC1 or later, or Windows .NET Server build 3505 or later, your system is not vulnerable. If your system is vulnerable, apply the patch found on Microsoft's web site [11].

Next, determine if your system has been infected by either the Code Red or Code Red II worms by checking the following: If your server has been infected by the original Code Red worm, your home page will have been defaced with the message "Hacked by Chinese." This message will only be displayed for ten hours, then the home page will revert to its original content. If your server has been infected by the newer Code Red II worm, you will find a file C:/Inetpub/scripts/root.exe on your hard drive. If you have drives other than C:, you should check them as well.

If your system has been infected, but you are confident that there has been no compromise other than the presence of the worm, install and run the Code Red Worm Cleanup tool. This is a small utility designed to "eliminate the obvious effects of the Code Red II worm from infected web servers [19]." In addition to eradicating malicious files installed by the worm, Code Red Cleanup reboots the system to eliminate memory resident code left by the worm. The tool also removes any mappings installed by the worm. Finally, it provides an option to permanently disable IIS on the server. However, it does not install the June 18 Microsoft patch to correct the buffer overflow vulnerability. It is important to note that both Microsoft and CERT recommend that the best way to recover from any system level compromise, such as a Code Red II infection, is to reformat the drive, reinstall the software and apply all the appropriate security patches [2].

Protecting Your Network

It is also important to keep in mind that an attacker's ability to control other machines from a compromised server depends in part on the network configuration itself, regardless of the vulnerabilities present on the individual systems. Machines accessible

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

to the internet face an inherently high risk of attack, and as a result should be protected through measures such as a DMZ, firewalls, access control lists, and running only minimal ports and services.

However, if TCP port 80 is allowed to pass through the firewall, systems inside the network will still be vulnerable to threats like Code Red and Code Red II. One means of protecting web servers that need to access TCP port 80 is to set up a reverse proxy server. All web content is cached on the reverse proxy server, so the attacker never actually reaches inside the network. However, the existence of a reverse proxy device should never serve as a reason for not keeping up with security patches for all servers on the network.

The Potential for More Damaging Worms

Active worms—or programs which replicate themselves by attacking servers on a network, have been around for several years. The Morris worm, which attacked systems in 1988, was one of the first well known worms with a mass effect. The Morris worm caused a major change in the way computer professionals and the public viewed the security of the internet. Since the Morris worm, many worms with minimally damaging effects have propagated across the internet. The Code Red worm was, in a sense, little more damaging than most of it predecessors. Although it caused a significant disruption in network services and forced system administrators to take immediate measures to protect their servers, it destroyed no data on the servers it infected. However, the huge amount of press coverage it received was a major factor in ensuring its notoriety.

In fact, many security professionals insist that Code Red, while posing a real problem, never really lived up to all the publicity it received [12]. The worm's major influence, many claim, was on of inconvenience rather than genuine destruction. They point out that the "paranoia" generated by Code Red could even be useful in preventing the effects of other similar worms—as long as people do not get complacent by assuming that just because Code Red did not get completely out of control, future worms won't either.

Will Code Red be the largest and fastest worm to infect the internet? The answer is no. Code Red is just a shot across the bow. The potential exists for even greater damage from worms that will spread faster and do far more damage that Code Red did. Previously released worms have required at least several hours to spread and become known, giving system and network administrators sufficient time to recognize the potential threat and take measures to mitigate the damage. Imagine a worm that could attack—not just in a matter of hours—but in a matter of minutes, as Nicholas C. Weaver from the University of California at Berkeley Computer Science Department suggests in his scenario and analysis entitled "Warhol Worms," based on Andy Warhol's statement that everyone will have 15 minutes of fame [20].

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Conclusion: Do All That You Can Do

Based on a scenario in which a worm could spread across the internet and inflict serious damage in a matter of just a few minutes, many skeptics – particularly those with little computer experience--might ask "Why bother keeping up with security patches?" Granted, viruses and worms have the potential to be created and propagate so quickly that even keeping up with available patches may never be sufficient protection against all possible attacks.

And, even the patching of most systems will not prevent the degradation of service potentially caused by a worm's denial of service capabilities. Had all systems been patched correctly, the problem could theoretically have been prevented altogether—because the vulnerability must exist on some systems in order to launch a truly successful denial of service attack. However, security professionals realize this is an unreasonable assumption.

But why? Is it because system administrators do not care about patching their systems? Probably not. A valid argument can be made that the information in the vulnerability reports published by vendors is too technical to allow an inexperienced system administrator to determine when his or her system requires a patch. In today's security community, the inexperienced system administrator is a fact of life due to the serious shortage of security-smart personnel working in the information technology field. Several companies including Microsoft are trying to address this issue with tools to assist administrators in identifying and patching potential security holes.

Will vulnerabilities go unpatched? Absolutely. Unfortunately, there will always be systems that remain on-line well after the vendor has released a fix and long after the vendor has ceased to support a particular version of an operating system or application. There is no excuse for not having a good security policy even on the smallest of networks with internet access. Mitigating risks with other strong security measures such as firewalls and web proxy devices can help, but this is no excuse for failing to keep systems current with vendor patches. In the end, companies and system administrators must justify their security position from a risk management perspective. They must attempt to balance the benefits gained by providing web access to employees and customers against the risks inherent in allowing this access both outside and inside their networks.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

References

[1] CAIDA Analysis of Code-Red, Cooperative Association for Internet Data Analysis CAIDA) Web Site, University of California's San Diego Supercomputer Center, 25 Aug 2001
http://www.caida.org/analysis/security/code-red

[2] CERT Advisory CA-2001-23, Continued Threat of the "Code Red" Worm, CERT Coordination Center Web Site, Carnegie Mellon Software Engineering Center, 26 Jul 2001
http://www.cert.org/advisories/CA-2001-23.html

[3] CERT Advisory CA-2001-19, "Code Red" Worm Exploiting Buffer Overflow in IIS Indexing Service DLL, CERT Coordination Center Web Site, Carnegie Mellon Software Engineering Center, 19 Jul 2001
http://www.cert.org/advisories/CA-2001-19.html

[4] CERT Incident Note IN-2001-10, "Code Red" Worm Crashes IIS 4.0 Servers with URL Redirection Enabled, CERT Coordination Center Web Site, Carnegie Mellon Software Engineering Center, 16 Aug 2001
http://www.cert.org/incident_notes/IN-2001-10.html

[5] CERT Incident Note IN-2001-09, "Code Red II": Another Worm Exploiting Buffer Overflow in IIS Indexing Service DLL, CERT Coordination Center Web Site, Carnegie Mellon Software Engineering Center, 6 Aug 2001
http://www.cert.org/incident_notes/IN-2001-09.html

[6] CERT Incident Note IN-2001-08, "Code Red" Worm Exploring Buffer Overflow in IIS Indexing Service DLL, CERT Coordination Center Web Site, Carnegie Mellon Software Engineering Center, 19 Jul 2001
http://www.cert.org/incident_notes/IN-2001-08.html

[7] 'Code Red' Worm Spreads, Pentagon Reacts, CNN Web Site, 1 Aug 2001
http://www.cnn.com/2001/TECH/internet/08/01/code.red/index.html

[8] eEye Advisory AL20010804, Code RedII Worm Analysis, eEye Digital Security Web Site, 4 Aug 2001
http://www.eeye.com/html/Research/Advisories/AL20010804.html

[9] eEye Advisory AL20010717, .ida "Code Red" Worm, eEye Digital Security Web Site, 17 Jul 2001
http://www.eeye.com/html/Research/Advisories/AL20010717.html

[10] eEye Advisory AL20010618, All versions of Microsoft Internet Information Services Remote buffer overflow (SYSTEM Level Access), eEye Digital Security Web Site, 18 Jun 2001

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

http://www.eeye.com/html/Research/Advisories/AD20010618.html

[11] Installing the Patch that Stops the Code Red Worm, Microsoft Web Site, Aug 2001
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/topics/codeptch.asp

[12] Interesting Media Roundup on the Truth of Code Red, Info-sec Web Site, 3 Aug 2001
http://www.info-sec.com/viruses/01/viruses_080301d_j.shtml

[13] Microsoft Security Bulletin MS01-033, Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise, Microsoft Web Site, 18 Jun 2001
http://www.microsoft.com/technet/security/bulletin/MS01-033.asp?frame=true

[14] NIPC Assessment 01-08, National Infrastructure Protection Center, 16 Aug 2001
http://www.nipc.gov/warnings/assessments/2001/01-018.htm

[15] Qwest Communications and Code Red Worm, TechTV Web Site, 23 Aug 2001
http://www.techtv.com/news/story/0,24195,3344033,00.html

[16] SANS Code Red Threat FAQ, SANS Internet Storm Center, Systems Administration, Networking, and Security (SANS) Web Site, 5 Aug 2001
http://www.incidents.org/react/code_red.php

[17] SearchSecurity Buffer Overflow Definition, SearchSecurity Web Site, 1 May 2001
http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci549024,00.html

[18] Symantec Code Red Worm Update, Symantec Antivirus Research Center Web Site, 23 Aug 2001
http://www.symantec.com/avcenter/venc/data/pf/codered.worm.html

[19] A Very Real and Present Threat to the Internet, Microsoft TechNet Web Site, Aug 2001
http://www.microsoft.com/technet/itsolutions/security/topics/codealrt.asp?frame=true

[20] Weaver, Nicholas C., Warhol Worms: The Potential for Very Fast Internet Plagues, Aug 2001
http://brass.cs.berkeley.edu/~nweaver/warhol.html

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 2nd, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced