Use offense to inform defense.
Find flaws before the bad guys do.

Incident report for GCIH course
Russell Hall
Sept. 24, 2000

Executive Summary

This is an incident report for an event that happened in Jan 1999 – May 1999. I was not a part of the incident handling team and will only have the information from the incident report and interviews to go from. There are a few parts of this incident that are different then most incidents. The system attacked was a temporary system placed outside of a firewall and was running intrusion detection software in a testbed capability. This system sent information it collected to another system behind the firewall that is a research and development system. The outside system was setup and installed with minimal software of only the operating system and the IDS system. There are a few unique items for this incident that show this was no "Script Kiddy". They broke into six different companies computer networks and was using the system broken into to hop to another company. The attacker ran buffer overflows to gain root access and installed Secure Shell (SSH) software to encrypt and hide their further activities on the network.

In this company (Company A) they exploited the outside firewall system to hop to other companies systems. They utilized information found on the other companies exploited systems to find accounts, usernames and passwords on a different companies system. They would then attack the systems at the company that had accounts on the previous system. They would scan the IP range of the user's account subnet's to find open service ports. Unfortunately they found an opening in the system AJAX at company A that enabled them to gain root access. The main problem with their attack, exploit and the transverse across the networks was the long time before detection. The discovery occurred in May 1999 with the exploit happening in January 1999. This leads to the concern of systems hacked and not discovering changes in the main software. By further evaluating the systems behavior after the discovery and checking connection logs there was minimal damage here. Other companies' system damage is unknown to me at this point. Another unique occurrence of the discovery of the exploit was when a person who works for company C and assists on the Research project at company A was called by company B and told that company C's system was a target for the hacker. When the company B representative explained the situation to him and named the systems from company B and A that were in the chain of attacked systems he recognized the AJAX system as this one from company A. The following diagram will show the hackers web of exploits.

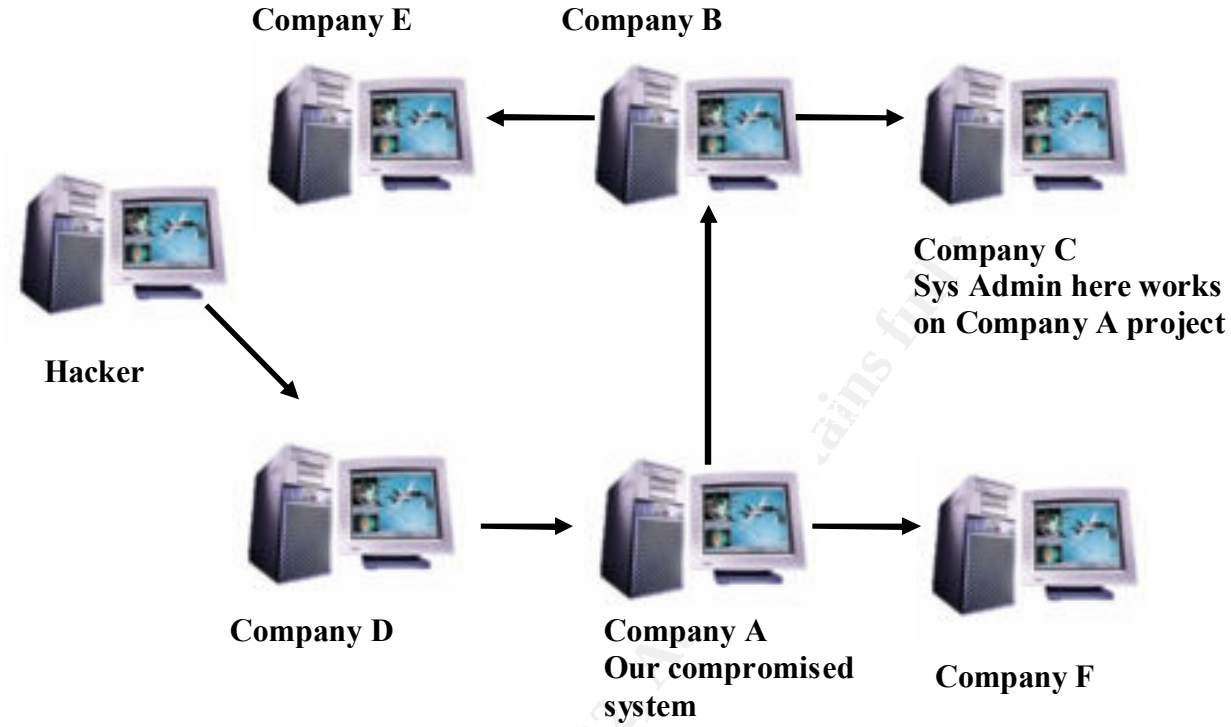


Figure 1. Topography of Hacker attack on Company A and total compromises by this incident.

The six stages of an incident are preparation, identification, containment, eradication, recovery and follow-up. The following will discuss this incident utilizing these six steps.

Preparation

Preparation to prevent an incident include the warning banner on all company systems that warns all users of the consent to monitoring, unauthorized use is prohibited and all activity can be used to prosecute any offenders. The Company A computer had the following warning banner:

This computer system, including all related equipment, networks and network devices (specifically including internet access), are provided only for authorized company use. Computer systems may be monitored for all lawful purposes, including ensuring that their use is authorized, for management of the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability and operational security. Monitoring includes active attacks by authorized entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied and used for authorized purposes. All information, including personal information, placed on or sent over this system may be monitored. Use of this computer system, authorized or unauthorized, constitutes consent to monitoring of this system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use

collected during monitoring may be used for administrative, criminal or other adverse action. Use of this system constitutes consent to monitoring for these purposes.

I interviewed the incident handler (Mr. D.) and asked about whether they had a "jump kit" as explained in the incident handlers course. Mr. D. stated for attacks they have a smart hub, cables, CD-ROM disk with trusted binaries including static linked tools and use these to check for contamination and capture keystrokes for prosecution if possible.

Identification

The identification of attack was discovered by another company and called to Mr. K at Company C although he also works at computer systems for Company A. I interviewed Mr. K. and asked how he became aware of the incident on AJAX @ Company A. He was notified by the network security person at Company B and was concerned over traffic from his company to Company C's computer network. Upon further analysis of where Company B's system was trying to connect at Company C they also noticed connections to systems at Company A. When the network security person mentioned the names of the computers to Mr. K he recognized the AJAX system as the one used in the research and development project. Then Mr. K and the network security person from Company B notified Company A's security personnel.

A system administrator for the machine at 256.256.256.213 was notified on May 12, 1999 that his system AJAX was hacked by another machine from a Company B on 12 January 1999. Four attempts were made to gain access to his machines; three were unsuccessful, one succeeded. The hacker then telneted from the compromised machine to the other organization and did an "rdist" buffer overflow. Then they obtained root access and using a "ToolTalk" buffer overflow installed Secure Shell (SSH) to hop to other systems encrypted and bypass network sniffers.

Buffer overflows happen when the software is written there isn't error-checking code to limit the amount of data entered into a variable's assigned space. When more data than space available is entered then the data is overwritten into the next-door variable's space and eventually can fill up the pointer space. This new code contains executable code along with a new return pointer address. This new address points to the new executable code and allows the code to be run a service at the root level. When they connected to the ToolTalk RPC service number and padded the buffer with No Operational (NOP) instructions they easily figure out how many addresses until the return pointer address space. The ToolTalk exploit can be seen at http://www.rootshell.com.

The following is from the incident log written by Company A after discovering the exploit.

16 January 1999

256.256.256.213 (Ajax) Company A telnet to 256.256.256.113 (lion) Company B

/tmp/.t is the buffer overflow program.
/tmp/.t was buffer overflow file.
nscfd - sniffer file - sniffer file stopped growing on 3 Feb 99.
ToolTalk used exploit to insert ssh key in /.ssh/authorized keys to allow root login with ssh.
Files hidden in /tmp/.t (buffer overflow program).
Sniffer program installed: /usr/sbin/nscfd.
Creating the following sniffer file: /usr/include/sys/sys_nu-t.h

We have seen him use the following usernames and passwords

UID/PID
cache/cache
visitor/novisitor
bptesta/2damOOn
bednarj/Sunbird
update/brickwall

Company B sent Company A the sniffer log from the Company B system for activity on 23 Dec 98 and 3 Jan 99.

USER visitor
PASS novisitor
PORT 137,246,34,24,138,0
NLST
CVI]D /tmp
MKD.T
CVVTD t
TYPE I
PORT 137,246,34,24,138,1
STOR info
PORT 137,246,34,24,138,2
STOR atmp
[TIMEOUT]

AJAX had hosts.deny empty file- running statd, rstat, telnet, ftp.
Note- AJAX had TCPWrappers installed and should have had hosts.deny set to deny all.

The following were recommendations from the incident team.

Use "ssh" on all DMZ systems.
Review sensors for activity from outside organization.
Identify all ssh connections on sensor as far back as we can.
Look at all activity going to outside organization.
Look for traffic on ports 33303 as far back as we can.
Document "forensics" or any changes in operating system software in the near past.

Call in sys administrators to look for "forensics"
Get 3 backups of AJAX, the compromised system.

According to the Mitre group Common Vulnerabilities and exposures listing at www.cve.mitre.org the following is an “rdist” buffer overflow.

CVE-1999-0022	Local user gains root privileges via buffer overflow in rdist, via expstr () function.
Brief Description	Inadequate boundary check allows stack data to be overwritten by user data.
References	CERT: CA-97.23.rdist, SUN:00179,XF: rdist-bo3, XF: rdist-sept97

The following is the actual SUN Advisory from The CIAC Computer Incident Advisory Capability bulletin board @ www.ciac.llnl.gov

CERT* Advisory CA-97.23
Original issue date: September 16, 1997
Topic: Buffer Overflow Problem in rdist

The CERT Coordination Center has received reports of a vulnerability in rdist that enables anyone with access to a local account to gain root privileges. This is not the same vulnerability as the one discussed in CA-96.14.

Section III.A contains instructions on how to determine if your site is vulnerable. If your implementation of rdist is vulnerable, the CERT/CC team encourages you to follow your vendor's instructions (Sec. III.B and Appendix A) or install a freely available version of the rdist program that is not installed as set-user-id root and is, therefore, not susceptible to the exploitation described in this advisory (Sec. III.C).

For information on the earlier problem with rdist, see ftp://info.cert.org/pub/cert_advisories/CA-96.14.rdist_vul

We will update this advisory as we receive additional information. Please check our advisory files regularly for updates that relate to your site.

I. Description

The rdist program is a UNIX Operating System utility used to distribute files from one host to another. On some systems, rdist opens network connections using a privileged port as the source port. This requires root privileges, and to attain these privileges rdist on such systems are installed set-user-id root.

A new vulnerability has been found in some set-user-id root implementations of rdist. The vulnerability lies in the function expstr(), where macros supplied as arguments are expanded using sprintf(). It is possible to overwrite stack frames and call especially pre-crafted native machine code. If the appropriate machine code is supplied, an attacker can execute arbitrary programs (such as the shell) with set-user-id root privileges.

Note that this vulnerability is distinct from that discussed in CERT advisory CA-96.14.

II. Impact

On systems with a vulnerable copy of rdist, anyone with access to a local account can gain root access.

III. Solution

We urge you to follow the steps in Section A to determine if your system is vulnerable and, if it is, to turn off rdist while you decide how to proceed.

If your system is vulnerable and you need the functionality that rdist provides, you should install a vendor patch (Section B). Until you can do so, you may want to use a freely available version of rdist that does not need to be installed as set-user-id root and is, therefore, not susceptible to the exploitation described in this advisory (Section C).

IV. How to check for set-user-id root versions of rdist

To find set-user-id root versions of rdist and to disable the programs that are possibly vulnerable, use the following find command or a variant. Consult your local system documentation to determine how to tailor the find program on your system.

You will need to run the find command on each system you maintain because the command examines files on the local disk only. Substitute the names of your local file systems for FILE_SYSTEM_NAMES in the example. Example local file system names are /, /usr, and /var. You must do this as root.

Note that this is one long command, though we have separated it onto three lines using backslashes.

```
find FILE_SYSTEM_NAMES -xdev -type f -user root \
        -name '*rdist*' -perm -04000 -exec ls -l '{}' \; \
        -ok chmod 0500 '{}' \;
```

This command will find all files on a system that
- are only in the file system you name (FILE_SYSTEM_NAMES -xdev)
- are regular files (-type f)
- are owned by root (-user root)
- have "rdist" as a component of the name (-name '*rdist*')

- are setuid (-perm -04000)

Once found, those files will
- have their names and details printed (-exec ls -l '{}')
- have the setuid mode removed (making the file available only to root) but only if you type `y' in response to the prompt (-ok chmod 0500 '{}' \;)

B. Obtain and install the appropriate patch

Below is a list of vendors who have provided information for this advisory. Details are in Appendix A, and we will update the appendix as we receive more information.

Berkeley Software Design, Inc. (BSDI)
Digital Equipment Corp.
FreeBSD, Inc.
Hewlett-Packard Company
IBM Corporation
NEC Corporation
The Santa Cruz Operation, Inc. (SCO)
Siemens-Nixdorf
Silicon Graphics Inc. (SGI)
Sun Microsystems, Inc.

If your vendor's name is not on this list, please contact the vendor directly.

C. If you need the functionality that rdist provides but a patched version is not yet available from your vendor, consider installing rdist-6.1.3, which is freely available from

ftp://usc.edu/pub/rdist/rdist-6.1.3.tar.gz

MD5 (rdist-6.1.3.tar.gz) = 8a76b880b023c5e648b7cb77b9608b9f

The README file in the distribution explains how to configure and install this version of rdist.

We recommend that you configure this version of rdist to use rsh instead of rcmd. Here is the relevant text from the README:

By default rdist uses rsh(1c) to make connections to remote hosts. This has the advantage that rdist does not need to be setuid to "root". This eliminates most potential security holes. It has the disadvantage that it takes slightly more time for rdist to connect to a remote host due to the added overhead of doing a fork() and then running the rsh(1c) command.

Some sites with sufficient expertise use the ssh program in conjunction with rdist, instead of using rcmd or rsh. If you have the expertise, you may want to implement this configuration.
For further details on this option see "Ssh (Secure Shell) FAQ - Frequently asked questions," Section 4.4, "Can I use rdist with ssh?" It is available from
http://www.uni-karlsruhe.de/~ig25/ssh-faq/ssh-faq-4.html

For details on how to obtain ssh, see FAQ Section 3.4, "Where can I obtain ssh?" This section can be found in
http://www.uni-karlsruhe.de/~ig25/ssh-faq/ssh-faq-3.html

According to the Mitre group Common Vulnerabilities and exposures listing at www.cve.mitre.org the following is a "ToolTalk" buffer overflow.

CVE-1999-0003	Execute commands as root via buffer overflow in ToolTalk database server (rpc.ttdbserverd)
Brief Description	Inadequate boundary check allows stack data to be overwritten by user data.
References	NAI: NAI-29, CERT: CA-98.11.tooltalk, SGI: 19981101-01-A, SGI: 19981101-01-PX, XF: aix-ttdbserver, XF: tooltalk, BID: 122

The following is the actual NAI Advisory from The CIAC Computer Incident Advisory Capability bulletin board @ www.ciac.llnl.gov

Stack Overflow in ToolTalk RPC Service

NAI Advisory 29
Network Associates, Inc.
SECURITY ADVISORY
August 31, 1998

SYNOPSIS

An implementation fault in the ToolTalk object database server allows a remote attacker to run arbitrary code as the superuser on hosts supporting the ToolTalk service. The affected program runs on many popular UNIX operating systems supporting CDE and some Open Windows installs. Attackers on the Internet are actively exploiting this vulnerability.

Confirmed Vulnerable Operating Systems and Third Party Vendors

Sun Microsystems

SunOS 5.6, 5.6_x86
SunOS 5.5.1, 5.5.1_x86
SunOS 5.5, 5.5_x86

SunOS 5.4, 5.4_x86
SunOS 5.3
SunOS 4.1.
SunOS 4.1.3_U1

Hewlett Packard

HP-UX release 10.10
HP-UX release 10.20
HP-UX release 10.30
HP-UX release 11.00

SGI

IRIX 5.3
IRIX 5.4
IRIX 6.2
IRIX 6.3
IRIX 6.4

IBM

AIX 4.1.X
AIX 4.2.X
AIX 4.3.X

TriTeal

TriTeal CDE - TED versions 4.3 and previous.

Xi Graphics

Xi Graphics Maximum CDE v1.2.3

It should be noted here that this not an exhaustive list of vulnerable vendors. These are only the *confirmed vulnerable* vendors. Also, any OS installation that is not configured to use or start up the ToolTalk service is not vulnerable to this problem. To determine whether the ToolTalk database server is running on a host, use the "rpcinfo" command to print a list of the RPC services running on it, as:

$ rpcinfo -p hostname

Because many operating systems do not include an entry for the ToolTalk database service in the RPC mapping table ("/etc/rpc" on most Unix platforms), the vulnerable service may not appear by name in the listing. The RPC program number for the ToolTalk database service is 100083. If an entry exists for this program, such as,

100083 1 tcp 692

then the service is running on the host. Until additional information is made available from the OS vendor, it should be assumed that the system is vulnerable to the attack described in this advisory.

DETAILS

The ToolTalk service allows independently developed applications to communicate with each other by exchanging ToolTalk messages. Using ToolTalk, applications can create open protocols, which allow different programs to be interchanged, and new programs to be plugged into the system with minimal reconfiguration.

The ToolTalk database server (rpc.ttdbserverd) is an ONC RPC service, which manages objects needed for the operation of the ToolTalk service. ToolTalk-enabled processes communicate with each other using RPC calls to this program, which runs on each ToolTalk-enabled host. This program is a standard component of the ToolTalk system, which ships as a standard component of many commercial Unix operating systems. The ToolTalk database server runs as root.

Due to an implementation fault in rpc.ttdbserverd, it is possible for a malicious remote client to formulate an RPC message that will cause the server to overflow an automatic variable on the stack. By overwriting activation records stored on the stack, it is possible to force a transfer of control into arbitrary instructions provided by the attacker in the RPC message, and thus gain total control of the server process.

TECHNICAL DETAILS

Source code and XDR specifications for the ToolTalk database protocol and server were not available at the time this advisory was drafted. What follows is information based on analysis of the rpc.ttdbserverd binary and a captured attack trace from a network on which an exploitation script for this problem was run.

The observed attack utilized the ToolTalk Database (TTDB) RPC procedure number 7, with an XDR-encoded string as its sole argument. TTDB procedure 7 corresponds to the _tt_iserase_1() function symbol in the Solaris binary (/usr/openwin/bin/rpc.ttdbserverd). This function implements an RPC procedure which takes an ASCII string as an argument, which is treated as a pathname.

The pathname string is passed to the function isopen(), which in turn passes it to _am_open(), then to _amopen(), _openfcb(), _isfcb_open(), and finally to _open_datfile(), where it, as the first argument to the function, is passed directly to a strcpy() to a pointer on the stack. If the pathname string is suitably large, the string overflows the stack buffer and overwrites an activation record, allowing control to transfer into instructions stored in the pathname string.

RESOLUTION

This is an implementation problem and can only be resolved completely by applying patches to or replacing affected software. As a temporary workaround, it is possible to eliminate vulnerability to this problem by disabling the ToolTalk database service. This can be done by killing the "rpc.ttdbserverd" process and removing it from any OS startup scripts. It should be noted that this might impair system functionality.

The following vendors have been confirmed vulnerable, contacted, and have responded with repair information:

Sun Microsystems

Sun plans to release patches this week that relate to the ToolTalk vulnerability for SunOS 5.6, 5.6_x86, 5.5.1, 5.5.1_x86, 5.5 and 5.5_x86.

Patches for SunOS 5.4, 5.4_x86, 5.3, 4.1.4 and 4.1.3_U1 will be released in about 4 weeks.

Sun recommended security patches (including checksums) are available from:
http://sunsolve.sun.com/sunsolve/pubpatches/patches.html

The reason that I placed the actual bulletins from SUN and the CIAC is to how involved the exploit is and even after the exploit is discovered there may not be a patch available. The practice of running scripts that downloads current patches and loads them to all systems at night still might not be enough if the hacker runs exploits before patches are available.

Containment

Mr. D. explained his method for containment of a confirmed exploit on a computer system. First he pulls the network connection of the infected system to prevent further use of the exploited software. He captures an image of the hard drives and creates a tape of the image to use as evidence if necessary. He places this tape in a box with tape on the outside with his, along with his supervisors' signature. This tape is placed in a safe with limited access for network security personnel only having the combination. Then he uses the hub in the jump kit and a "fishbowl" system that has the same OS and on the same subnet as the infected system and adds the message of the day "motd" service to state that there are network problems and connectivity can be problematic. Then by monitoring this system he can see if the hacker is still scanning for vulnerable systems. This may or may not lead to further activity from the intruder. Prosecution is determined by the extent of the damage and length of time from exploit to identification to containment.

Recovery

After it is determined that the hacker hasn't infected any other systems at Company A and there is no further attempts from them to scan the network then restoring the system is required. Since this attack happened over a long time frame and it was a research and development system we reformatted the hard drives, reinstalled the Operating System with the current patches and reinstalled the IDS system required for the project. The companies affected also restored their systems from previous backups prior to December 1998. They have lost data, as there is no way to ensure the files on more recent backups don't have Trojans installed. As far as we at Company A we lost little data. There is no way to measure the amount of data lost at other companies and the loss of trust for our systems. It will take awhile for that to be restored. Also on Friday, 28 May 99, 1600 Company A installs filters to stop all IP traffic access to AJAX.

Follow-up

The follow-up included patching all systems to the current patch cluster from SUN Microsystems and other UNIX vendors. Keeping current all patch clusters even on temporary systems will prevent further incidents from occurring. The incident report was written and forwarded to all department heads involved along with letters to Company B informing them of corrective actions. This incident was stressed to all system administrators to take the time to ensure systems are completely patched with software and keeping them current is a high priority.

Lessons Learned

The workload of the system administrators has increased exponentially and extra work requests must take a backseat to network security. This incident is a result of an overloaded administrator that always gets priority one requests that must be accomplished yesterday. Everyone must remember that computer network security is extremely important keeping up with all of the exploits is time consuming. Spending money on software or hardware that assists the security personnel in preventing exploits will save at least ten times the money spent. There have been incidents in the newspaper that have cost companies millions and when the hacker was caught the sentence was small and fine was about one hundredth of the damage. Prevention is more important then wasting time to restore systems that could have had not been exploited if a company had better Intrusion Detection Systems. Firewalls, network and host based Intrusion Detection Systems and constant monitoring and reconfiguring as exploits are identified will help prevent contamination. Adding additional personnel and training to the computer network security department will pay better then a hot stock on Wall Street.

The hackers use the Internet to spread the word of any vulnerable systems and one isolated incident can increase quickly. A newspaper article recently mentioned that the hackers that deface web sites aren't the problem. The real problem is corporate espionage that has cost companies Millions in lost contracts when outbid by a couple of percent, stolen credit card numbers and using systems to hop around to cover their tracks.

On MSNBC the following article was posted. A teen-ager was sentenced to six months in jail Thursday after pleading guilty to federal charges of hacking into NASA computers that support the International Space Station. The teen, now 16, also admitted he had illegally entered a Pentagon computer system, intercepted 3,300 e-mail transmissions and stolen passwords. "Breaking into someone else's property, whether it's a robbery or a computer intrusion, is a serious crime," Attorney General Janet Reno said. In a plea bargain, the young hacker admitted to entering 13 computers at the Marshall Space Flight Center in Huntsville, Ala., for two days in June 1999 and downloading $1.7 million in NASA proprietary software that supports the space station's environmental systems. NASA said it cost $41,000 to check and repair the system during the three-week shutdown after the illegal entry was discovered.

References:

Mitre Corporation - Common Vulnerabilities and exposures listing at www.cve.mitre.org

The CIAC Computer Incident Advisory Capability bulletin board @ www.ciac.llnl.gov

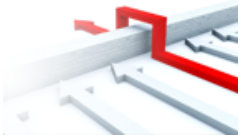
MSNBC web site for article http://www.msnbc.com/news/464929.asp

SANS Computer and network hacker exploits: Step-by-Step, Part 1, Eric Cole, Book 4.2, GIAC GCIH Certification program.

SANS Hacker Exploits Workshop, Eric Cole, Book 4.4, Track 4, GIAC GCIH Certification program.

Last Updated: October 20th, 2018

Upcoming SANS Penetration Testing



Community SANS Ottawa SEC560	Ottawa, ON	Oct 22, 2018 - Oct 27, 2018	Community SANS
SANS vLive - SEC660: Advanced Penetration Testing, Exploit Writing, and Ethical Hacking	SEC660 - 201810,	Oct 23, 2018 - Nov 29, 2018	vLive
Houston 2018 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Houston, TX	Oct 29, 2018 - Nov 03, 2018	vLive
Community SANS Kansas City SEC560	Kansas City, KS	Oct 29, 2018 - Nov 03, 2018	Community SANS
SANS Houston 2018	Houston, TX	Oct 29, 2018 - Nov 03, 2018	Live Event
Mentor Session - SEC504	Oklahoma City, OK	Nov 03, 2018 - Dec 08, 2018	Mentor
SANS Gulf Region 2018	Dubai, United Arab Emirates	Nov 03, 2018 - Nov 15, 2018	Live Event
SANS Dallas Fall 2018	Dallas, TX	Nov 05, 2018 - Nov 10, 2018	Live Event
Community SANS Omaha SEC504	Omaha, NE	Nov 05, 2018 - Nov 10, 2018	Community SANS
SANS DFIRCON Miami 2018	Miami, FL	Nov 05, 2018 - Nov 10, 2018	Live Event
Mentor Session - SEC560	Des Moines, IA	Nov 05, 2018 - Dec 08, 2018	Mentor
SANS London November 2018	London, United Kingdom	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS Sydney 2018	Sydney, Australia	Nov 05, 2018 - Nov 17, 2018	Live Event
Mentor Session - SEC504	Cincinnati, OH	Nov 06, 2018 - Dec 18, 2018	Mentor
SANS Osaka 2018	Osaka, Japan	Nov 12, 2018 - Nov 17, 2018	Live Event
Pen Test HackFest Summit & Training 2018	Bethesda, MD	Nov 12, 2018 - Nov 19, 2018	Live Event
Mentor Session - SEC504	Vancouver, BC	Nov 17, 2018 - Dec 15, 2018	Mentor
Mentor Session AW - SEC504	Hong Kong, China	Nov 25, 2018 - Dec 08, 2018	Mentor
SANS Stockholm 2018	Stockholm, Sweden	Nov 26, 2018 - Dec 01, 2018	Live Event
Community SANS Reno SEC504	Reno, NV	Nov 26, 2018 - Dec 01, 2018	Community SANS
Austin 2018 - SEC542: Web App Penetration Testing and Ethical Hacking	Austin, TX	Nov 26, 2018 - Dec 01, 2018	vLive
SANS San Francisco Fall 2018	San Francisco, CA	Nov 26, 2018 - Dec 01, 2018	Live Event
Austin 2018 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Austin, TX	Nov 26, 2018 - Dec 01, 2018	vLive
SANS Austin 2018	Austin, TX	Nov 26, 2018 - Dec 01, 2018	Live Event
Mentor Session AW - SEC560	Colorado Springs, CO	Nov 28, 2018 - Dec 07, 2018	Mentor
SANS Nashville 2018	Nashville, TN	Dec 03, 2018 - Dec 08, 2018	Live Event
SANS Santa Monica 2018	Santa Monica, CA	Dec 03, 2018 - Dec 08, 2018	Live Event
SANS Dublin 2018	Dublin, Ireland	Dec 03, 2018 - Dec 08, 2018	Live Event
Community SANS Falls Church SEC560	Falls Church, VA	Dec 03, 2018 - Dec 08, 2018	Community SANS
Mentor Session AW - SEC504	St. Petersburg, FL	Dec 05, 2018 - Dec 14, 2018	Mentor
Community SANS Portland SEC504	Portland, OR	Dec 10, 2018 - Dec 15, 2018	Community SANS