Interested in learning more about security?

SANS Institute InfoSec Reading Room

Nessus Primer with the NessusWX Client

The process of securing information takes on many forms. Some forms are proactive and some are reactive. Many methods of securing data and the systems that hold the data can actually be both proactive and reactive. Case in point, the installation of virus scanning software and the updating of virus signatures are proactive in that it protects a system or a network from known viruses, worms, and Trojans. However, this same security measure can be reactive in dealing with the damage caused by the new and unknown variety ...

Nessus Primer with the NessusWX Client
Cecil Stoll
GIAC Security Essentials Certification (GSEC)
Practical Assignment 1.4b Option 1
June 26, 2004

Abstract:

In the ever changing world of technology, it has become imperative to secure information both in transit across the network wire and on the end systems where information data is stored. Nessus is a public domain free program designed to help system and network administrators scan networks for known vulnerabilities and thus assist in closing these vulnerabilities before they can be exploited.

The beginning of this paper will be introductory material. Following this introduction will be discussions as to what Nessus is, and what it can do. A simple installation of the Nessus server from sources onto a test machine running FreeBSD 4.9 will be conducted. The client portion of Nessus will be installed onto a Windows XP machine. Although it will be impossible in the brevity of this assignment to cover all aspects of Nessus, test scans will be run on a few varying systems running Windows and FreeBSD operating systems. Each will be running a number of different services and processes which will be tested. The end result of these scans will be discussed and some reports generated. Concluding remarks and a discussion of some of the limitations of Nessus will finish out the document.

Introduction:

The process of securing information takes on many forms. Some forms are proactive and some are reactive. Many methods of securing data and the systems that hold the data can actually be both proactive and reactive. Case in point, the installation of virus scanning software and the updating of virus signatures are proactive in that it protects a system or a network from known viruses, worms, and Trojans. However, this same security measure can be reactive in dealing with the damage caused by the new and unknown variety of viruses that have been so prevalent as of late. Proactive measures can include the installation of firewalls, the encryption of data while in-transit, the encryption of data while stored on an end system, the securing of these same end systems, and the processes that run on them just to name a few.

The focus of this paper will be to proactively seek out known vulnerabilities on the end systems and the processes running on them. In my current employment, I work in the IT department for a very small community college. This presents a challenge in securing systems on our campus, not only from a technical aspect, but also from a financial one as well. Our state has experienced a financial crunch much the same as many states, and therefore we do not have large

budgets to invest in expensive vulnerability testing software or equipment. Fortunately, with the development and proliferation of open source and public domain software, we do not have to. Nessus is an excellent tool for seeking out known vulnerabilities and at the same time, as it is public domain software, will not break the budget. In support of this statement, Harry Anderson in his article "Introduction to Nessus" on the Security Focus web site states:

> Historically, many in the corporate world have ridiculed such public domain software as being a waste of time, instead choosing supported products developed by established companies. Typically these packages cost hundreds or thousands of dollars, and are often purchased using the logic that you get what you pay for. Some people are starting to realize that public domain software, such as Nessus, isn't always inferior and sometimes it is actually superior.[1]

Features

Nessus offers a very long list of features which allows for multiple scanning options. There are two parts to the Nessus software, the server side and the client side. Since the client/server architecture is followed, a great deal of flexibility is provided. One Nessus client can access a number of different Nessus servers placed at various points in the network. Likewise one Nessus server can be accessed from a number of Nessus clients allowing different administrators to setup and perform scans.

One of the prominent features of Nessus is intelligent scanning. An approach is taken of assume nothing and take nothing for granted. No assumptions are made as to what services are running on which ports. For example, an ftp service would normally be assigned to and possibly running on the well-known port 21. Scans using Nessus do not make this assumption and do not assume the IANA port numbers will be honored. If port 21 is open, checks will be made to assure it is an ftp service, if it is not, scans will be run according to the service that is running.

When the correct service application and version are discovered, no assumptions are made in regards to whether security risks are present or not. The scan will continue and attempt to exploit the service. This is done in case software patches or other fixes have been applied to the software, which may not have shown a version change. In doing so, a more accurate scan and thus much more accurate results will be provided.

Scans can be run in two modes, a full intensive mode will aggressively seek out vulnerabilities and actually try and exploit them. This type of scanning can bring services to a complete halt and crash the system. It is the more accurate of the

[1] Anderson, Harry. "Introduction to Nessus." SecurityFocus. 28 Oct. 2003. URL: http://www.securityfocus.com/infocus/1741 (24 Jun 2004)

two modes of scanning. The second scan is a non-destructive scan. It relies on software banners to determine if vulnerabilities exist. Non-destructive scanning does NOT take an assume nothing approach. This is a safer method since vulnerabilities based on software versions are assumed. No attempt to exploit the vulnerabilities is made in this method making a DoS or system crash much less likely.

Network scans can be run on a single machine or can be run on an entire host of machines at once. The robustness of the network along with the processing power, speed, and memory of the server will be factors used to determine the number of hosts to scan at any given time. The test server used for the scans later in this paper is a Pentium II 450 MHz machine with 128 MB of RAM memory. Given the simplicity of the test scans, this proved to be enough power to run and finish the scans within a time frame of 5 to 25 minutes.

Full SSL support is built into the Nessus program, which gives it the ability to scan and test services with SSL support. A complete list of security checks Nessus can perform may be found at http://cgi.nessus.org/plugins/. However, given that security risks are increasing almost on a daily basis or that a network may have special circumstances requiring special needs, the list of checks Nessus provides may fall short. The creators of Nessus have taken this into account and have created NASL for these situations. NASL stands for Nessus Attack Scripting Language. NASL allows system administrators to write/create their own security tests, thus adding even more to the flexibility and scope of Nessus as a complete vulnerability scanner. As an option to NASL, security tests may be written in the C programming language.

When it comes to vulnerability scanning, it is important to ensure the integrity of the program and the scans. To that end and to insure its integrity, Nessus has been written and developed by people who are independent from the rest of the world. Autonomous programmers and developers help to insure vulnerabilities will not be hidden or glossed over because of contracts or other financial obligations between the developers and commercial entities.

In an effort to reduce the bandwidth of a Nessus scan, the **Knowledge base** feature has been added. This option stores the list of open ports, the type of system scanned, and much more information about the scan of a host or network. This is done so information discovered by one plugin may be used or shared by another plugin thus eliminating the repetitions of tests, resulting in a savings of time and bandwidth for scans. For more complete documentation and description of this feature, point your web browser to http://www.nessus.org/doc/kb_saving.html.

The output from a Nessus scan gives the user a variety of information. If a port scan is taken, open ports are listed. Vulnerabilities and security holes are listed along with security warnings. Nessus will also provide possible solutions to the

security issues listed. For an example, the following screenshot shows a version of OpenSSH which, as is stated, is vulnerable to a flaw in the buffer management which can allow an attacker to execute arbitrary commands on the suspect server.

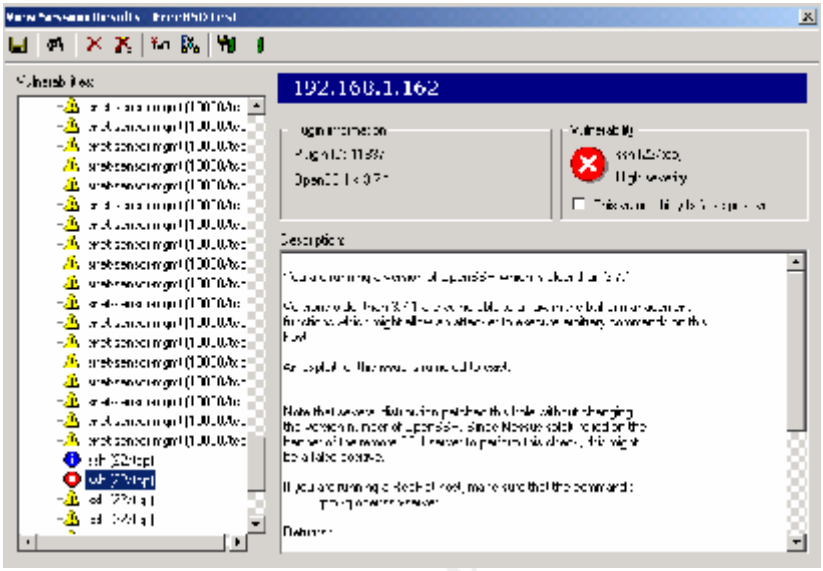


Figure 1 Example of Nessus results showing SSH vulnerability.

Nessus continues by giving examples on how to resolve this issue by upgrading the vulnerable program with an update. The examples given in this situation are for Linux users but never the less, the information provided is what is important. In this situation, the scan result shown is from a FreeBSD system. Obviously, trying to install a Linux RPM on a *BSD box is not going to work. In the instance of FreeBSD, several options are available to the user for updating OpenSSH on the system. These include installing from the ports collection, installing a package which is similar to a Linux RPM, and of course a system administrator can always compile and install from the applications source code.

Installation

The installation of Nessus is a two-fold process. Since the program is a client/server model, the server program and the client must be installed. The Nessus website at http://www.nessus.org/install.html gives a very brief, step by step installation of the Nessus server. Installation of the server from the source code is relatively straightforward. As explained in the web document, be sure to do the installation in the order listed or problems will be encountered. To ease the download and installation, a Nessus install script is available for UNIX systems. The information and download of the installer script may be found on the Nessus web site at http://www.nessus.org/nessus_2_0.html. Installation of Nessus can be accomplished by downloading the script and executing it from the command line as:

sh ./nessus-installer.sh

This command will configure, compile, and install Nessus in one step. Compiling from the source code may be the best alternative if special installation or setup needs are required.

A Nessus client is installed along with the server, as long as you do have the GIMP toolkit installed. If GIMP is not installed, a stripped down version of the client will be created. This stripped down client may be run from the command line. The Edgeos website has documentation regarding the options available when running either the Unix GUI or command line clients. The address for this information is http://www.edgeos.com/nessuskb/. According to Edgeos, their Nessus knowledge base has information about every option and configuration variable available to the Nessus scanner. A Windows version of the client is available for download at http://nessuswx.nessus.org/ . The NessusWX client is currently at version 1.4.4, and the download does include a Windows installer program to ease the setup process.

After the installation is complete, a few more steps remain before scans may be conducted. Users must be added to the Nessus system with the **nesssus-adduser** command. Nessus keeps a database of users allowed to attach to the server.

Nessus encrypts all of the communications between the Nessus server and the Nessus client. SSL is used for the encryption process and an SSL certificate must be created. Creation of the server certificate and certificate authority are accomplished with the **nessus-mkcert** command. Although this is not a mandatory step, it is highly recommended. After all, the purpose here is to make security, of the network and systems therein, as tight as possible.

Running Nessus Scans.

At this point, we are ready to begin using Nessus for some test scans. Before scanning any network, whether on test equipment or not, it is important to have permission to do so. This is extremely important, as stated earlier, Nessus scans can and will at times cause processes to stop or crash resulting in a denial of service(DOS), or worse yet, cause the entire system to crash and become unusable. As related in the SANS GSEC course, "the difference between a penetration tester and an attacker is **PERMISSION**"[2]. Therefore, scans must not be run without permission or be prepared to deal with the any consequences of doing so.

The first step in performing a scan is to attach the Nessus server and to setup the scan. In these exercises, the Windows client will be used to attach to the

[2] Cole, Eric et. al. Internet Security Technologies. SANS Institute Jan. 2004, 204

server unless otherwise indicated. To start the client in Windows, double click the **NessusWX** icon. The initial Windows client screen appears and should be very similar to Figure 2 below.

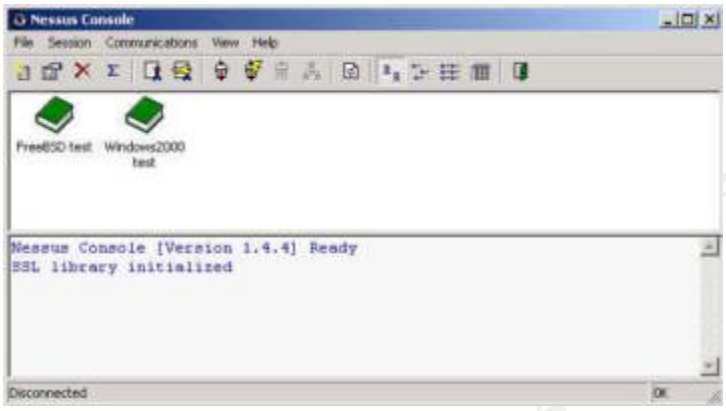


Figure 2 Nessus Console screen.

A status window shows the Nessus console is ready and the SSL library has been initialized. At this point, a connection to a Nessus server has NOT been established as shown in the lower left hand corner indicating the client is in the disconnected state. To connect to a server, click on the **Communications** menu, and then click on the **Connect** option. The connect window will appear and will be similar to the screenshot in Figure #3.

Figure 3 Windows client Connect screen.

The **Connect** button may be clicked after typing in the server name or IP address and the Nessus username. If this is the first time to log into the specified server the New Server Certificate windows will appear and ask for acceptance or rejection of the SSL certificate provided by the server. As can be seen in the Figure #4 below, there are three options in this window. An administrator may **Reject**, **Accept Once** or **Accept and Save** the specified certificate. To continue the connection process, either the **Accept Once** or the **Accept and Save** buttons must be clicked. If the **Reject** button is clicked, the connection the server will be aborted and the administrator will be returned to the Nessus console

screen. If the **Accept Once** button is clicked, the SSL certificate will be accepted and used only for the current session. The next time the Nessus client is run and attached to the same server, the user will be prompted for acceptance of the same certificate again. And obviously, if the **Accept & Save** button is clicked the certificate will be accepted and then saved with the client. With this last option, the next time this same client accesses the specified server, the administrator will not be prompted to accept the server certificate.

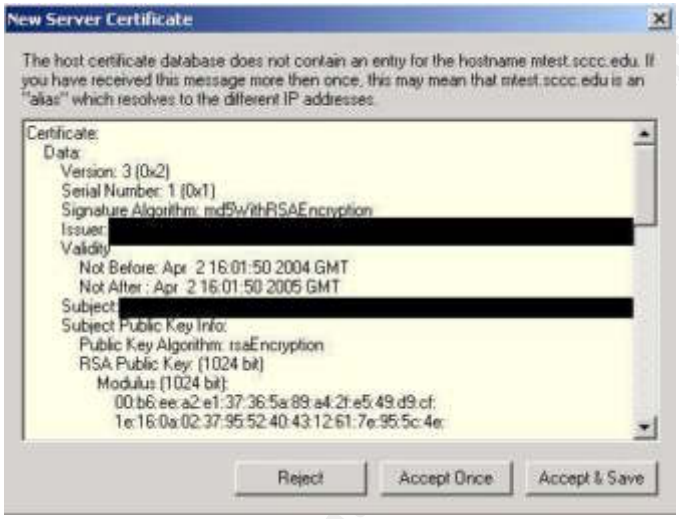


Figure 4 New Server Certificate screen.

After choosing either **Accept Once** or **Accept & Save**, the administrator will be prompted for a password. Upon entering the correct password, a connection with the server will be complete and the server to which the client is connected will be indicated in the status portion of the Nessus Console screen.

As mentioned in the Features section of this paper, Nessus has the ability to do a non-destructive scan. This will only check software banners to decide if a specific vulnerability exists. The first test scan will be non-destructive, followed by the same scan with the non-destructive safe checks disabled. To perform this non-destructive scan the **Safe checks** box must be selected when setting up the scan. To create a new scan, click on the **Session** menu option, and then click on **New**. Type in a descriptive name of the scan to be performed in the text box provided.

After typing in the session name, the scan configuration screen appears. This screen should be similar to the one in Figure #5 below. The **Targets** tab should already be selected. It is here that an administrator chooses what to scan. At this point, click on the **Add** button to choose what to scan. As Figure #6 shows, a single host, an entire subnet, or a range of IP addresses may be chosen. For the purpose of this paper, only a single host will be selected as the “guinea pig” for testing. After selecting the target machine(s), clicking on the **Options** tab, will open a window shown in Figure #7. In this screen, an administrator may

choose whether to run the scan with safe checks (non-destructive) or not. The default in the Windows NessusWX client is to have the **Safe checks** button on. For the first scan safe checks will be turned on. Figure #8 is the Port scan configuration screen. Here one may choose what ports to scan and which port scanners applications to use. For these scans, NMAP will be the only port scanner selected.

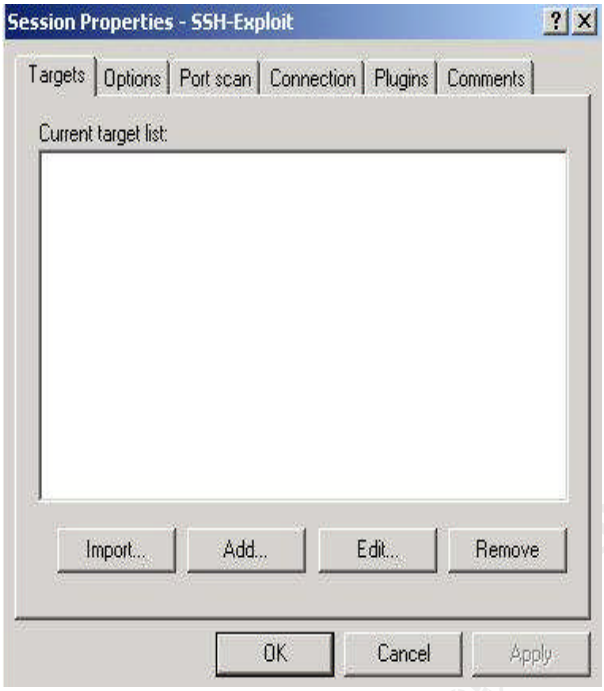


Figure 5 Scan Session configuration screen.

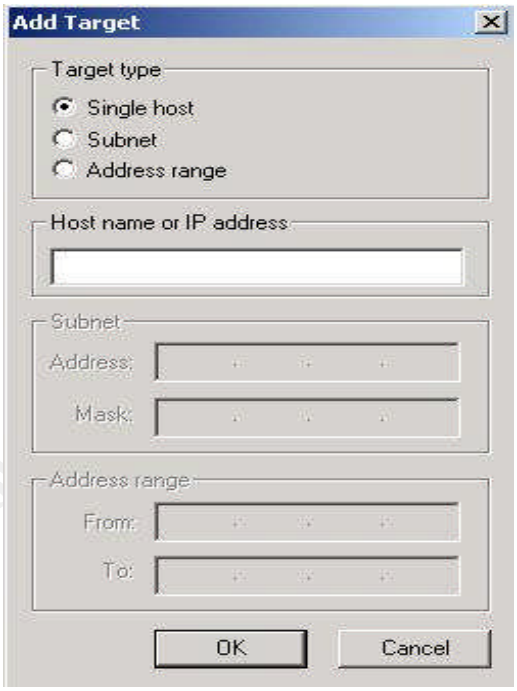


Figure 6 Add Target-select what to scan.

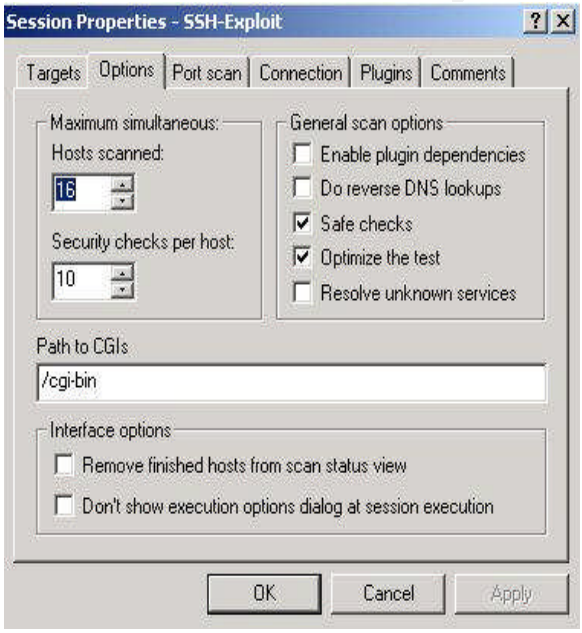


Figure 7 Scan Session Options screen.

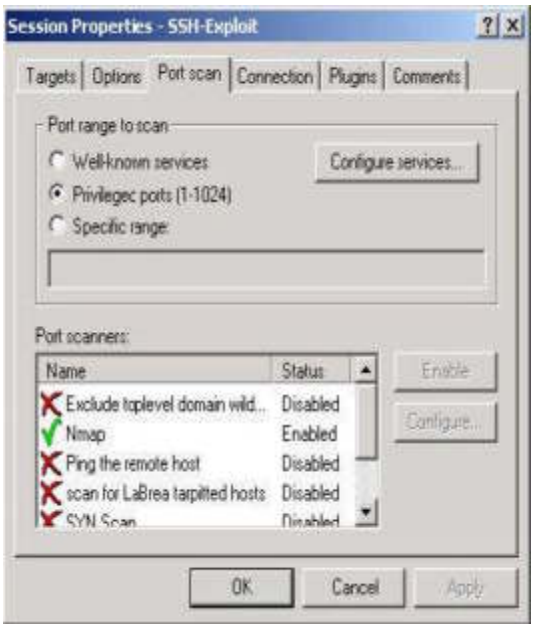


Figure 8 Session Port scan configuration.

Probably the most important tab in creating a new session is the Plugins tab. In the Plugins window, the administrator has the option to pick and choose what tests to run. Figure #9 below shows the main Plugins window. The **Select Plugins** option will need to be checked before performing a scan. Some plugins may or may not be relevant to the system being scanned. Such it is fruitless to run plugins checking for Windows vulnerabilities if the system being tested is UNIX based machine and vice-versa.

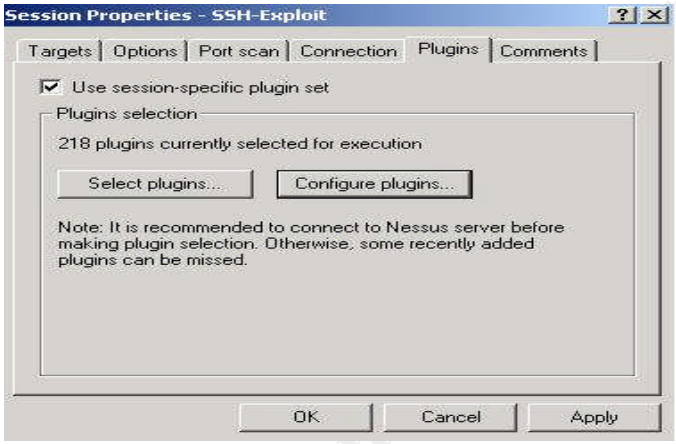


Figure 9 Session scan main Plugins window.

Clicking on the **Select plugins** button will display a list of plugins grouped by family. Figure #10 below shows the plugin groupings. Each group may be expanded and individual plugins may be enabled or disabled as the administrator sees fit.

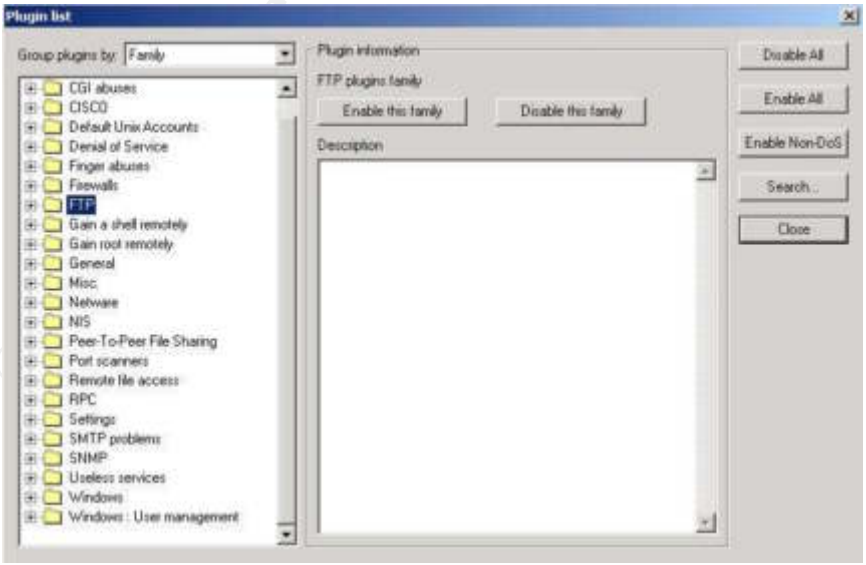


Figure 10 Plugins list grouped by family.

An entire family of plugins may be enabled or disabled with the click of a button. All plugins may be enabled with the **Enable All** button. Another precaution

Nessus adds is the **Enable Non-DoS** button, thus preventing Nessus from running scans which will instigate a Denial of Service against a system or network. This is a precaution as with any vulnerability scanning, it is not a guarantee something will not happen to crash a service or system.

Now that we have gone through the basic screens in setting up a scan using the Windows client, a set of test scans will be conducted first with the safe checks activated and the second without the safe checks. For the sake of this test, only the FTP family plugins will be enabled and the port scanners except for NMAP will be disabled. In an effort to improve the accuracy of this scan, a username and password will be provided to Nessus in order for the Nessus server to be able to login to the FTP server. This user has the ability to upload to a test folder on the FTP server. To provide this username and password, the administrator must click on the **Configure Plugins** button in Figure #9 above. The resulting screen is shown in Figure #11 below. Click on the Login configurations to expand the list of options. In our example, an FTP account name of testftp is typed in along with the password for this account.

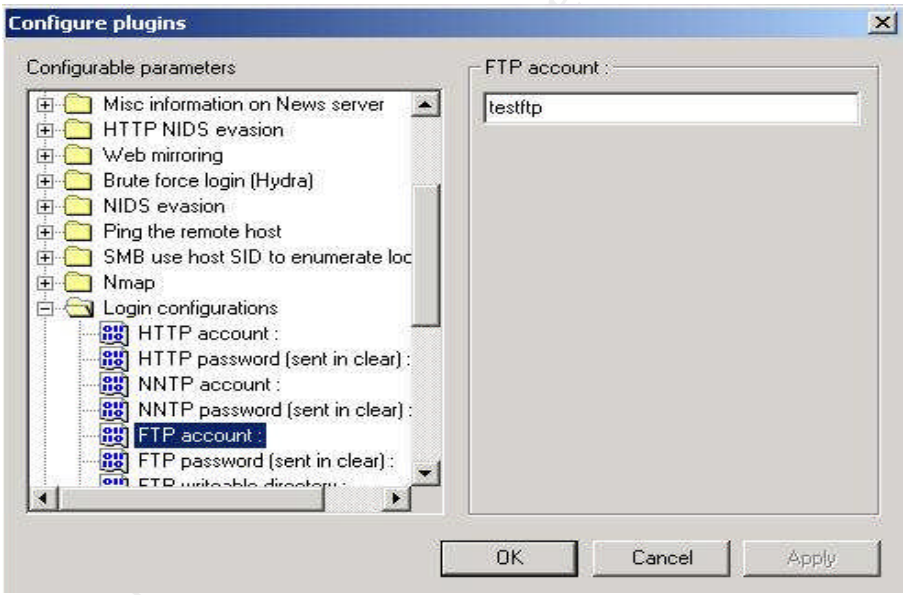


Figure 11 Configure Plugins screen.

The results of the first check are shown in Figure #12. According to this scan, two vulnerabilities are displayed. The high severity vulnerability is shown to be from plugin #10821 and listed as the FTPD glob Heap Corruption flaw. As listed in the description, this flaw can allow an attacker to execute arbitrary commands on the vulnerable server. However, since safe checks are enabled, the Nessus server does not try and exploit the flaw to ensure it is a risk. With safe checks enabled, false positives become more of a factor in the vulnerability scans.

Figure 12 Results from Safe checks FTP scans.

In an effort to compare apples to apples this same scan is now repeated on the same server under the same conditions, except the safe checks are now disabled. Again no other port scanners are enabled. The same plugins (FTP family) are the only ones enabled. After re-running the scan, the second results differ from the first scan. Both scans show anonymous ftp logins are allowed. However, the second scan shows a high severity vulnerability that is different from the first scan. The vulnerability listed can allow an attacker to exhaust the memory of the server resulting in a Denial of Service and very possibly an entire system crash. This is detailed in Figure #13 below.

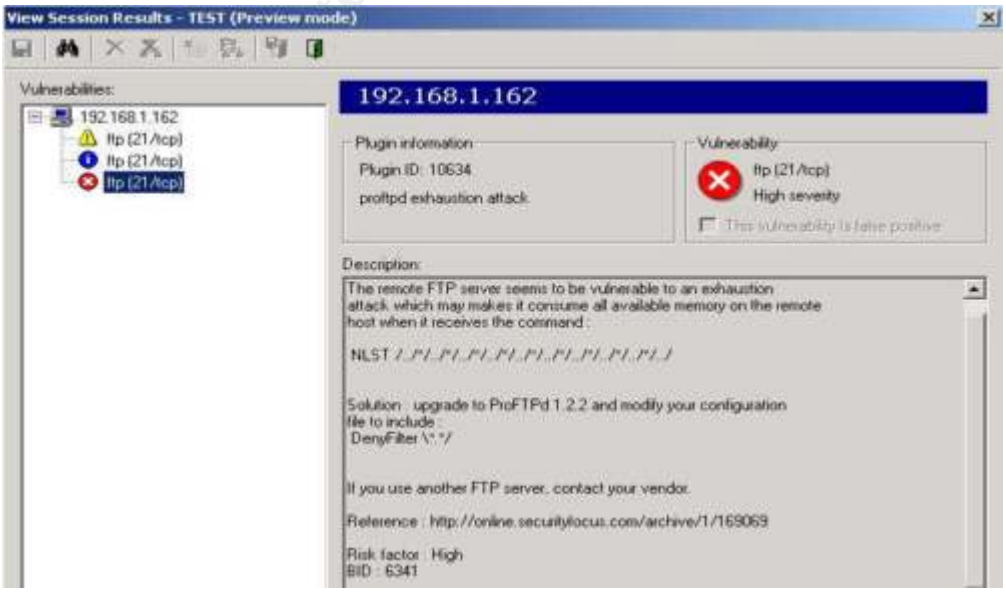


Figure 13 Results from destructive FTP scans.

These two simple scans magnify the importance of running scans with the safe checks disabled. This allows for a more accurate scan from the Nessus server. When a hole is discovered such as the one listed in Figure #13 hints or advice are given on how to close the security hole. In this instance, Nessus suggests downloading and installing ProFTPD 1.2.2. To test the accuracy of the advice given, the latest version of ProFTPD was downloaded. A basic installation of ProFTPD was made by compiling from source code using the standard command set of:

1. ./configure
2. make
3. make install

Then to run the new FTP server out of the internet super server, the following line was added to the /etc/inetd.conf file.

```
ftp          stream  tcp     nowait  root    /usr/local/sbin/proftpd proftpd
```

The HangUP process command is sent to the inetd daemon with the kill –HUP 'inetd.pid' in order to have the new inetd configuration to be in effect. To test the new changes, Nessus FTP scans are run against the server again which is now using ProFTPD version 1.2.10 instead of the default FreeBSD 4.9 FTP server. The results show anonymous logins are still allowed, but that the severe security holes of the previous scans are now closed, as displayed in Figure #14 below.

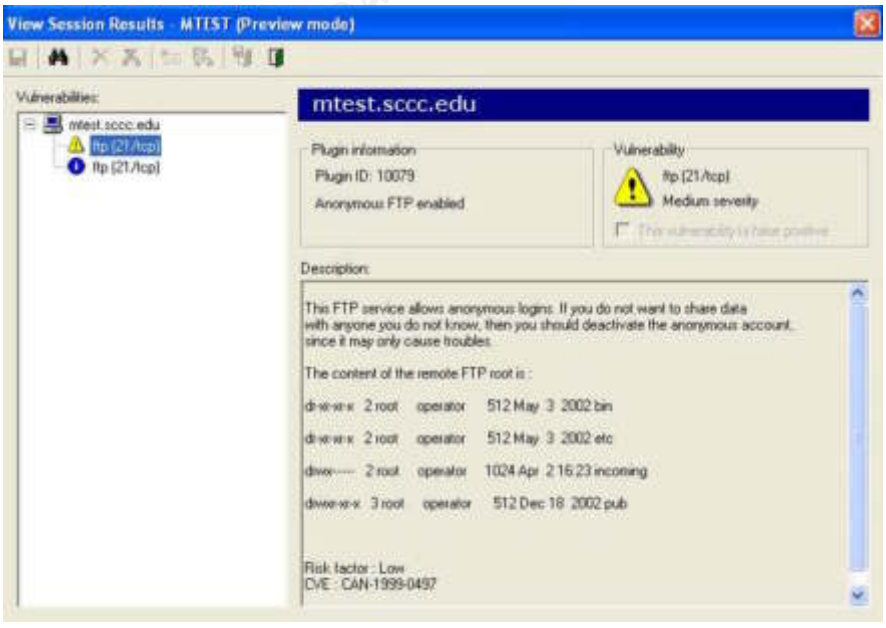


Figure 14 FTP scan with critical holes now closed.

A few very important conclusions can be drawn from these examples. Nessus is a very good tool for identifying and suggesting fixes for open security holes in an

end system. It is important to run a thorough scan to ensure the correct security flaw is identified and so the breech may be properly closed.

Reporting

The previous scans were obviously accomplished targeting a UNIX operating system. Since many end workstations are running Windows operating systems, a sample scan of a Windows XP machine will be included with the results of the scan exported to a file in PDF format. This report will be easier to read on screen and when printed out than the results boxes shown in earlier test scans.

The sample scan of the Windows XP machine will have all plugins and the NMAP port scanner enabled. This first scan will be conducted on the XP machine with no patches or service packs installed. Windows XP service pack 1a and all other security patches available will be installed on the machine. A second scan will then be conducted.

The **Manage Session Results** screen from this initial scan is show in Figure #15 below.

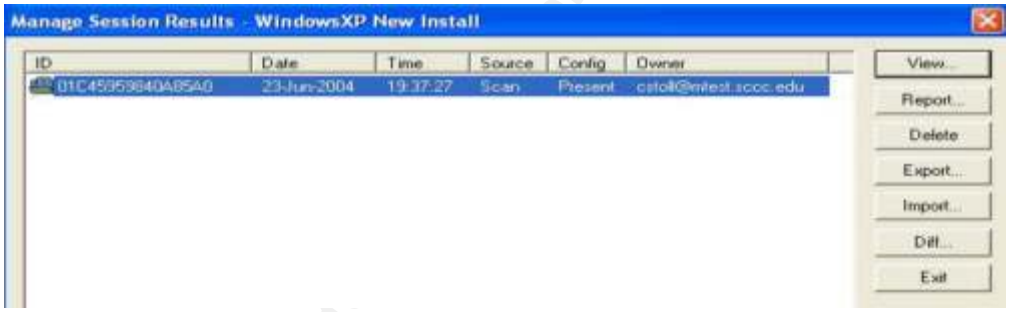


Figure 15 Manage Session Results screen.

From this screen, the administrator may generate a report by clicking on the **Report** button. This will open the **Report Options** window shown in Figure #16 below.

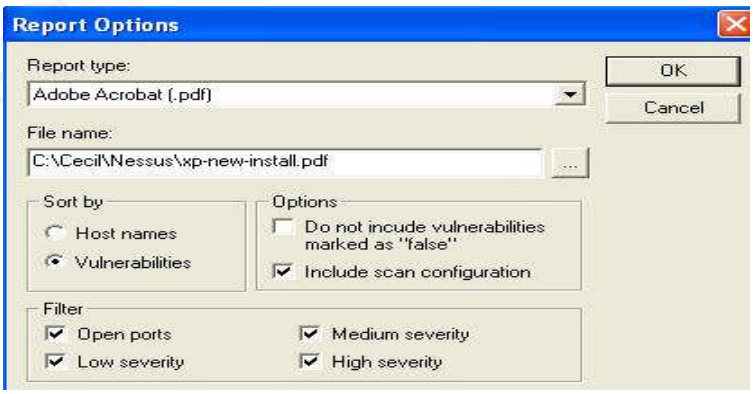
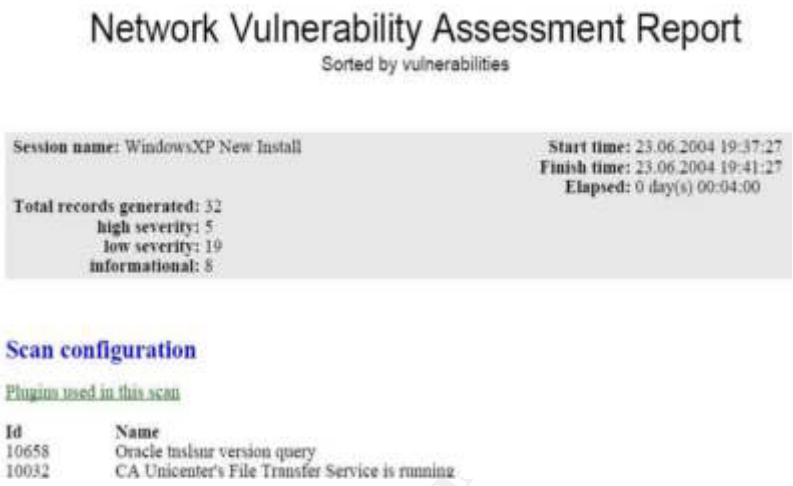


Figure 16 Report Options screen.

Reports may be generated into one of three formats. A report may be stored in an Adobe Acrobat .pdf, a web .html, or as a plain text .txt file. For this example, the Acrobat format will be used. This report will also list all levels of vulnerabilities and open ports. Since this scan was only of one host, the sort by **Vulnerabilities** option is checked. The following screenshots are samples of the report generated by Nessus.



Network Vulnerability Assessment Report

Sorted by vulnerabilities

Session name: WindowsXP New Install

Start time: 23.06.2004 19:37:27
Finish time: 23.06.2004 19:41:27
Elapsed: 0 day(s) 00:04:00

Total records generated: 32
high severity: 5
low severity: 19
informational: 8

Scan configuration

Plugins used in this scan

Id	Name
10658	Oracle tnslsnr version query
10032	CA Unicenter's File Transfer Service is running

Figure 17 Nessus report banner.

Figure #17 shows the beginning of the Nessus report with a summary of the number of vulnerabilities in each category and it also lists the plugins used in the scan. Since all the plugins were enabled on this scan, the listing of the plugins used in the scan is obviously quite long. If that information is not needed, then un-checking the **Include scan configuration** checkbox in report options shown in Figure #16 is probably a good idea. Figure #18 below shows a sample listing of the “ping of death” vulnerability found on the XP machine.

Port unknown (1026/udp) is open

Vulnerable hosts

192.168.5.1

general/tcp

Description

The machine crashed when pinged with an incorrectly fragmented packet.
This is known as the 'jolt' or 'ping of death' denial of service attack.

An attacker may use this flaw to shut down this server,
thus preventing you from working properly.

Solution : contact your operating system vendor for a patch.

Risk factor : High

Vulnerable hosts

192.168.5.1

Figure 18 Sample report - "ping of death" vulnerability.

Since this report is sorted by vulnerabilities instead of by hosts, the most severe flaws are listed first. In the example above the first severe flaw is listed which is known as the “ping of death”. The “ping of death” is a denial of service generated by sending a ping packet that exceeds the 65535 byte IP datagram limit, yet it is possible to send this packet because of the way it is fragmented for transmission. The fragments are sent to the target machine and the packet is re-assembled there. Since the re-assembled packet is larger than the legal limit, it can cause a buffer overflow. The buffer overflow can cause unpredictable results including a system crash. More information on this flaw may be found at the Insecure.org website at http://www.insecure.org/sploits/ping-o-death.html. Nessus used the “ping of death” to attack the target computer and according to the sample report above was able to crash the system. The unpredictable results, in this case, happened to be a system crash and a reboot of the machine.

Another feature of the Nessus reporting is the ability to generate a Diff report. This report can display the differences between two scans. Likewise, similarities between two scans may also be found by generating a different type of Diff report. For an example, the above Windows XP scan will be re-run with the same parameters before. Patches and service packs have been applied to the machine now. A differential report may be created by selecting one of the results listed in the **Manage Session Results** window and then clicking on the **Diff** button shown in Figure #15. Figure #19 below shows the resulting window.

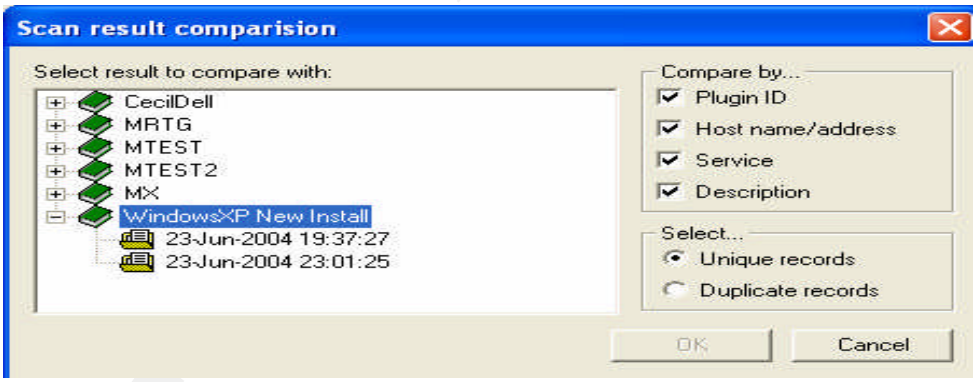


Figure 19 Scan result comparison window.

In this case, a comparison is being made between the scan created at on June 23 at 7:37 pm, and the scan created on June 23 at 11:01 pm. An administrator can create a report showing the records which are in the first scan but are not in the second scan. Or if the **Duplicate records** radio button is selected in Figure #19, then the records shared in both scans will be shown. This is somewhat awkward, but it does show the differences or similarities, if any, between two scans.

The reporting of Nessus seems to be an area of concern for some administrators with larger networks to control. In an article for Computerworld titled “Cheap

Scanning Comes at a Price", Mathias Thurman expresses concerns about the Nessus reporting feature. He points out "No matter how robust, easily manageable, intuitive and inexpensive the tool is, if we can't produce meaningful reports, it's hard to get management support.[3] For small networks, the Nessus reporting may be fine. For networks with thousands of systems spanning a global network, a great amount of time is needed to gather all of the necessary Nessus data and compile it into a central database.

Besides the XML output, Nessus does offer an **Export** command where results may be exported to other formats for the manipulation of the raw data. Nessus can export results into a comma delimited flat file, a MySQL database, and a SQL command file just to name a few. In doing the exports into these formats, reports may be generated using the reporting tools available for the database application being used. Again a limiting factor in generating these reports will be the availability of personnel to spend the time gathering data and generating these reports.

Updating Nessus

As with any technology, and especially in the security field, updates are extremely important to detect and eradicate the stream of new threats emerging almost on a daily basis. Nessus has a script included to keep the plugins current. The **nessus-update-plugins** script is run from the command line of the Nessus server machine. This can be run from a cron job at different intervals using a line in crontab such as:

```
10    23    *    *    1-5    /usr/local/sbin/nessus-update-plugins
```

This will execute the update script Monday through Friday at 11:10 pm each night.

The manual page for the program does warn of a security risk in using this script. The NASL scripts being downloaded are not signed using PGP or another signing entity. A cracker could poison the DNS server being used to send a forged response to the **nessus-update-plugins** script, resulting in the update script downloading an invalid NASL script file. Since no integrity checks are run against the download, malicious scripts could be installed onto the Nessus server machine. A cautious implementation of new Nessus scripts should be pursued. Running the new scripts on a test network and server isolated from the rest of the network first will help to eliminate any surprises, thus keep any forged scripts from doing a great deal of damage.

[3] Mathias Thurman. Computerworld 5 Apr 2004 http://www.computerworld.com/managementtopics/management/story/0,10801,91829,00.html (24 Jun 2004)

The update plugin script uses the tar, gzip and lynx programs to download and install the scripts. With the implementation of FreeBSD 4.9 being used for the Nessus server, a problem did occur with the downloading and unzipping of the NASL scripts. The version of Lynx being used was somewhat dated.(2001) and required an upgrade before the plugin update would download and install correctly. Problems encountered with this script may be corrected easily by ensuring the lynx, tar, and gzip programs used by the system are up to date and current.

Shortcomings and Commercial scanners

As powerful as Nessus is, it does have shortcoming that need to be addressed. The reporting tool has already been addressed above. In large organizations, centralized management of all the Nessus servers would be a nice addition. In the current version, administrators must login to each Nessus server separately to setup and perform scans. When the number of servers is larger than four or five systems, the task of logging into each and every one for configuration and management purposes becomes extremely tedious and time consuming.

In large enterprises, it may be necessary to add Nessus administrators who do not have full control of the entire set of Nessus plugins. If an individual is a user on a Nessus system, then this individual can run any and all tests available. Nessus does not support users with limited scanning rights. A feature enabling a user to be able to run only a limited set of checks on one portion and only one portion of the network could distribute the work load of performing security scans. At the same time this user would not have the ability to run all scans thus limiting the danger of bringing a system or entire network down.

Commercial products are available that do offer what is lacking in Nessus. The additional features come at a price obviously. Commercial products do address the reporting and management issues. An article on Network Computing's web site offers a comparison of 11 different commercial vulnerability scanners. A few of the scanners tested were Foundstone's FoundScan, Qualys' QualysGuard and eEye's Retina. Tenable Nessus was of interest in the article since it is a commercial front-end for the Nessus scanner. The complete article for more information may be found at http://www.networkcomputing.com/1412/1412f2.html.

Conclusion

In the course of this paper, simple scans of systems have been performed. Following the scans, a few security patches were installed and the scans were re-run showing a security flaw had been eliminated. In doing so, the true power and usefulness of Nessus has been demonstrated. The securing of data and information is the ultimate goal of any vulnerability scanner. As can be

concluded, even in the brevity of this paper, Nessus can and will help network and system administrators reach that goal.

References

Anderson, Harry. “Introduction to Nessus.” SecurityFocus. 28 Oct. 2003. URL: http://www.securityfocus.com/infocus/1741 (24 Jun 2004)

Anderson, Harry. “Nessus, Part 2:Scanning.” SecurityFocus. 16 Dec. 2003. URL: http://www.securityfocus.com/infocus/1753 (24 Jun 2004)

Anderson, Harry. “Nessus, Part 3: Analyzing Reports.” SecurityFocus. 28 Oct. 2003. URL: http://www.securityfocus.com/infocus/1759 (24 Jun 2004)

Bradley, Tony. “Nessus Vulnerability Scanner”. About.com URL: http://netsecurity.about.com/cs/vulnerabilities/a/aa040604.htm (24 Jun 2004)

Thurman, Mathias. “Cheap Scanning Comes at a Price.” Computerworld. 5 Apr 2004. URL:http://www.computerworld.com/securitytopics/security/story/0,10801,91829,00.html (24 Jun 2004)

Novak, Kevin. “VA Scanners Pinpoint Your Weak Spots.” Network Computing. 26 Jun 2003 URL: http://www.networkcomputing.com/1412/1412f2.html (24 Jun 2004)

“Features.” Nessus. URL: http://www.nessus.org/features (25 Jun 2004)

“NessusWX – Nessus Client for Win32.” URL: http://nessuswx.nessus.org/ (25 Jun 2004)

“Download the stable version of the Nessus Security Scanner for Unix-compatible systems” Nessus. URL: http://www.nessus.org/nessus_2_0.html (25 Jun 2004)

“Plugins.” Nessus. URL: http://cgi.nessus.org/plugins (25 Jun 2004)

“Saving the knowledge base” Nessus. URL: http://www.nessus.org/features (25 Jun 2004)

“How to install Nessus on a Unix-like system.” URL: http://www.nessus.org/install.html (25 Jun 2004)

Deraison, Renaud. “nessus-update-plugins(8)” User Manual Page. May 2000

“Nessus Knowledge Base.” Edgeos URL: http://www.edgeos.com/nessuskb/ (25 Jun 2004)

Cole, Eric. SANS Security Essentials Cookbook, Version 2.2 SANS Institute, 2003, 11-1 – 11-26.

Cole, Eric et. al. Internet Security Technologies. SANS Institute 2004. 204

Kenney, Malachi. “Ping of Death.” Insecure.org. 21 Oct 1996. URL: http://www.insecure.org/sploits/ping-o-death.html (25 Jun 2004).

Last Updated: June 25th, 2017

Upcoming SANS Training

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

SANS Cyber Defence Canberra 2017	Canberra, AU	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MDUS	Jun 26, 2017 - Jul 01, 2017	Live Event
SEC564:Red Team Ops	San Diego, CAUS	Jun 29, 2017 - Jun 30, 2017	Live Event
SANS London July 2017	London, GB	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, JP	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TXUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, SG	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CAUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, DE	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DCUS	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TNUS	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TXUS	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Hyderabad 2017	Hyderabad, IN	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MAUS	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, CZ	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UTUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NYUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, ILUS	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, AU	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VAUS	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CAUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FLUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NVUS	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, IE	Sep 11, 2017 - Sep 16, 2017	Live Event
SANS Paris 2017	OnlineFR	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced