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 Role of Bastille Linux in Information Security

Over the last 15 years, Linux has evolved from one man's hobby into a very robust and capable operating system (OS). Today, Linux is being used around the world in a wide variety of applications in businesses, academia, and industry. In addition to being a powerful general purpose Unix clone, Linux provides a cost efficient means to implement a wide range of computer security related functions including security auditing, computer forensics, intrusion detection, firewalling, routing, and vulnerability scanning. The con...

Copyright SANS Institute
Author Retains Full Rights

AD

The Role of Bastille Linux in Information Security

Michael Russell Grimaila

February 18, 2002

Introduction

Over the last 15 years, Linux has evolved from one man's hobby into a very robust and capable operating system (OS). Today, Linux is being used around the world in a wide variety of applications in businesses, academia, and industry. In addition to being a powerful general purpose Unix clone, Linux provides a cost efficient means to implement a wide range of computer security related functions including security auditing, computer forensics, intrusion detection, firewalling, routing, and vulnerability scanning. The concept of "Defense in Depth" relies upon each of these essential functions to provide a layered security solution. Since the operating system is the underlying framework on which all of these applications operate, it is imperative that it be as secure as possible. In this paper, I will briefly examine the evolution of Linux, discuss its popularity, and examine in detail Bastille Linux, which is used to increase the security of RedHat and Mandrake Linux distributions.

The Evolution of the Linux Operating System

In order to understand the popularity of Linux, we must first examine the successes and failures of its predecessor: The Unix operating system. Although Linux does not include any of the original Unix code, it is a functionally equivalent Unix clone. It all started back in 1969, when Ken Thompson, Dennis Ritchie and others started working on a PDP-7 at Bell Labs in Murray Hill, New Jersey on what was to become the Unix OS. Their goal was to develop a dependable multi-user, timesharing operating system where the environment was one of cooperation and not necessarily one of privacy. After a number of trials and tribulations, they succeeded. On November 3, 1971, Bell Labs released the first version of the Unix OS. Initially, Unix was made available to universities and research institutes for the cost of the magnetic tape used to distribute it. However, by the mid-1970's Bell Lab's released a commercial version of Unix. Within a few years vendors were marketing their own, diverging, versions of the Unix system optimized for their own applications and computer architectures, each with different strengths and features.

The popularity of Unix was due to its integrated multitasking capability, multi-user capability, portability, rich library of application software, and shell interface. The combination of these features provided a powerful software development environment that had never previously existed. Unix made it possible to solve complex programming problems by interconnecting a series of smaller general-purpose programs rather than creating a single, large application program. As more people became familiar with the unique features of Unix, the demand for Unix servers increased.

By the early 1980's, one could purchase any one of more than a dozen different, largely incompatible, versions of Unix. Commercial Unix vendors at this time were more interested in their own market share instead of sharing their innovations in the OS. Much to the delight of the vendors, customers became dependent upon them to implement improvements in the OS. Businesses quickly learned that although different Unix systems were readily available for

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

various platforms, they seldom were able to operate together without significant investment of money, time, and effort. It seemed as if the vendors were purposely impeding the development of the Unix operating system.

This trend began to change in early 1984 when Richard Stallman quit his job at MIT and began working on an open source version of the Unix. He despised the environment that prevented OS vendors from sharing in each other's innovations. His idea was to encourage a community of cooperating programmers to work towards the common good by developing a free Unix clone and related software. In 1985, Stallman founded the Free Software Foundation, a tax-exempt charity for free software development. The Free Software Foundation eventually created the GNU General Public License, the widely adopted licensing agreement that allows Linux and other software to remain completely free.

In 1987, Professor Andrew Tanenbaum developed Minix, an open-source operating system clone of Unix to teach his students the inner workings of a real operating system. He designed his OS to run on the Intel family of microprocessors, which at that time were readily available worldwide. Tanenbaum formed a newsgroup in support of the Minix project that had a strong following of more than 40,000 people. Although many of the newsgroup participants would email Tanenbaum with new code for inclusion into Minix, he would refuse their contribution because he wanted to keep the code small enough for his students to learn within one semester.

One of the Minix newsgroup followers was Linus Torvalds, a computer science student at the University of Helsinki in Finland. Torvalds took keen interest in Minix and after studying it he began thinking about how he might develop his own Unix clone. By mid-1991, Torvalds announced his plans to create his own free operating system on the Minix newsgroup. He considered his efforts to only be a hobby and did not believe that it would compete with the GNU OS that Stallman was developing. Later that year, Torvalds released the first Linux version on the Internet under a GNU public license. The rest is history. Since that time, countless people have made contributions to the Linux project by writing kernel enhancements, device drivers, and integrating GNU programs into Linux. Linux is now a mature, fully capable operating system that is used worldwide in a variety of different applications.

The Popularity of the Linux Operating System

The popularity of Linux cannot be attributed to a single factor but instead is the result of a large number of factors. As a Unix clone, Linux cannot be beat. Since it is distributed under an open source license, if a vendor adopts an innovation that becomes popular in the market other vendors can immediately adopt that innovation. As new technological advances are made in computer hardware, people unselfishly contribute their time, effort, and skills to write drivers and kernel modifications to take advantage of the hardware. If a particular vulnerability is identified in the core OS or related applications, programmers worldwide start simultaneously working to provide a patch to correct it. Some people do this for the common good, while others wish to gain some limelight for being the first one to fix the vulnerability. In either case, all Linux users benefit from these efforts. The informal environment of "cooperative-competition" results in discovered vulnerabilities being corrected much significantly sooner in Linux than in commercial versions of Unix.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Linux has found its home in the information infrastructures of organizations worldwide. It runs on the most popular computer architectures, is very inexpensive, and provides a robust means to implement generic computing servers. It can also be configured to provide critical network infrastructure capabilities such as routers, firewalls, and DNS servers. Linux also offers a great deal to security professionals. The vast number of security applications available makes it one of the best choices for use by the budget minded information security professional. The only thing that was lacking was a unified approach to securing the operating system.

What is Bastille Linux?

Bastille Linux is a set of PERL scripts that you run as "root" immediately after you do a "fresh" install of Linux on your target system. It can be run in a text mode or a graphical mode. You can also select interactive mode or in non-interactive mode. Using the interactive mode is highly recommended, as it will educate you by explaining the impact of each of the configuration settings while querying you for your response. It is essential that you run the Bastille scripts immediately after installing the OS and before placing your system on a network. This will help insure that it does not get hacked before the security configuration is completed. The Bastille scripts allow you to easily implement the industry "best practices" configuration to maximize the security of your system to insure your information security. The scripts install the latest available patches, disable all unnecessary services, secure the default configurations, and set up a firewall based upon your needs. The scripts incorporate the latest recommended security configurations as identified by CERT, SANS, and other trusted security authorities. It should be noted that information security encompasses more that just insuring the confidentiality of a system. Equally important are the integrity and the availability of the system. The Bastille scripts address this by minimizing the possibility of compromising a systems "CIA" as a result of poor configuration.

In the past, Linux distributions did not always provide you the capability to select the security level for your system. Most installation programs would enable a large number of unnecessary system services by default in an effort to make the resulting system fully functional to the average user. Since most system services need to run at an elevated privilege, a flaw that resulted in compromising that service will often lead to compromising the entire system. Since new vulnerabilities are being discovered every day, it makes sense to disable all unnecessary services. Today, most Linux distributions allow you the flexibility to select the initial "security" level and will configure the system appropriately. However, it is still wise to run the Bastille scripts when you configure a new Linux box to make sure that nothing was overlooked.

Unfortunately, one cannot guarantee that a system will not be compromised. The complexity of the software makes it difficult, if not impossible, to complete comprehensive security testing. The best thing you can do is to insure that your system is securely configured and that all of the known security patches have been applied to the system. The Bastille scripts provide an excellent way to achieve these goals while simultaneously providing you a means to become educated on the issues related to secure configuration. However, security is not a static subject and one must remain current on all security issues. You should monitor security advisory mailing lists daily such as those provided by Security Focus (www.securityfocus.com) and the SANS Institute

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

(http://www.incidents.org/about.php) to insure that you don't expose your system to an attack from a known vulnerability.

Who Developed Bastille Linux?

The Bastille Linux Project began as a joint venture between Jon Lasser, Ben Woodard, and others who met during the SANS '98 Conference. They shared the concern that there were a large number of inexperienced individuals who were installing Linux on their systems and not configuring their systems securely. As a result, many Linux boxes were being hacked. Their plan was to develop a set of modular scripts that would implement the security industry "best practices" for RedHat Linux systems. They drew upon their individual experiences as well as "SANS Securing Linux Step-by-Step", Kurt Seifried's "Linux Administrators Security Guide", and Jay Beale's Solaris hardening scripts. The latest version of Bastille is 1.2.0, which currently supports RedHat 6 and 7, and Mandrake 6, 7, and 8 Linux distributions. You can download the latest version of Bastille from www.bastille-linux.org as a RedHat RPM package or a complete source tarball. In any case, make sure that you verify the signatures after you download it to insure that you have an uncorrupted copy.

How Does Bastille Linux Work?

Bastille Linux helps you configure various networking components, system permissions, and system services so that you minimize your likelihood of having your system compromised. Bastille first determines if you are installing a server or a workstation. Based upon this information, the scripts determine which services are likely to be enabled and aid you in their configuration. Further, you may select one of three security levels (Lax, Moderate, or Paranoid) that determine how much security you wish to enforce. As it steps you through the configuration of the system, it educates you on the benefits and drawbacks associated with each setting.

Bastille Server Configuration

When you select server configuration, a number of items are set by default regardless of which of the three security levels you select. These setting correspond to the lax server security configuration. Bastille allows you to override each of the recommended settings, but you are warned about the dangers of doing so.

SUID capability for the programs dump, restore, cardctl, the DOS emulator DOSEMU, and the news server are all disabled to minimize the chance that a vulnerability can be used to gain elevated privilege. SUID is a mechanism that allows normal users the ability to run programs that require system level privileges. The danger is that if one can subvert an SUID program, one can compromise the system. Password aging is enabled so that accounts are disabled if the password has not changed within the last 180 days. This is good security policy and is intended to prevent inactive accounts from being cracked. The single user mode is password protected using the root password. This prevents someone who has physical access to the system from rebooting the system, entering single user mode, and compromising the system. Additional messaging is enabled to log kernel messages, error messages, and warning messages to a log file. One should review these logs periodically for signs of intrusion. A good tool for managing this

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

task is SWATCH (http://www.oit.ucsb.edu/~eta/swatch/). It is wise to send log messages to a separate log host system and reference this system using its IP address. Logging to a remote system reduces the likelihood that a local compromise will be undetected as a result of a hacker "cleaning" the logs on the compromised system. Referencing the system by IP address will prevent disruption of your logging is your DNS server is compromised. APMD, NFS, Samba, pcmcia, DHCP server, news server, routing daemons, NIS, SNMPD are all disabled. Each of these special programs provides unique services and vulnerabilities. You should only enable the programs and daemons you need and only if you fully understand the security risks in doing so. Sendmail's verify recipient existence (VRFY) and expand recipient alias/list contents (EXPN) commands are disabled so that a hacker cannot easily identify user accounts on the system. For example, one can use the EXPN command to identify the "postmaster" and "abuse" emails redirects, which often point to the user who is the system administrator. The DNS server is disabled and should only be enabled if you intend upon running a DNS server. If you require a DNS server, you can restrict the named daemon to have access only to a limit subset of the file system. Placing the named in a CHROOT jail severely restricts the damage the named daemon can inflict if it is later compromised. The Apache web server is disabled as it provides a number of potential vulnerabilities if it is misconfigured. If you decide to enable the Apache web server, Bastille encourages you to disable server-side includes. Server-side includes provide a means for the web server to execute code which can be used to provide powerful capabilities to a web page. However, if the server-side code is insecure it can be exploited to compromise the security of the system. The telnet and ftp protocols are inherently insecure because the user name, password, and data are all transacted in plain text and can be compromised by "sniffing" the network. For this reason, telnet and ftp are disabled. If you are configuring the Mandrake version of Linux, the msec program is enabled to enforce nightly security checks that can provide a warning if it detects the system has been compromised.

If you select moderate server security, the following configuration items are set by default in addition to the items set by selecting lax server security. The firewall is configured to block all unnecessary services. What is considered unnecessary is determined by a question and answer session where you can selectively allow/deny connections based upon their protocol, IP address, port number, or network interface. The BSD remote access tools rcp, rlogin, and rsh are disabled because the same functionality can be found in ssh and scp, which are much more secure. The BSD remote tools use IP-based authentication, which makes them vulnerable because anyone can initiate or hijack an existing connection to a system that appears to come from a "trusted" system. It is also encouraged to create a root-owned, non-writable .rhosts file in each users home directory. The .rhosts file contains the user name and machine name pairs of accounts that are consider trusted which allows the user to connect to another system without having to retype a password. This represents an incredible risk because a hacker can compromise an account (even root) by inserting data into the appropriate users .rhosts file. Session hijacking of a trusted connection and compromising a .rhosts file is exactly what was successfully used in the Mitnick attack. If you decide to enable the Apache web server, you are encouraged to disable all CGI bin scripting. If your users are not aware of the inherent risks of running code on the system via the web server CGI bin scripts, you should probably disable them. I personally have seen more than one case where a user was intending to provide enhanced capabilities on their web page by calling an application that runs on the server. In this case, the individual accessing the web page was allowed to pass commands directly to the operating system!. A hacker can exploit this

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

weakness in a number of different ways to compromise the server. The FTP daemon is disabled for a number of reasons. FTP anonymous mode has proven to be especially problematic because it allows anyone to connect and upload and download files to the system. A simple misconfiguration of the FTP server may provide the hacker a copy of the system password file. Hackers often need to upload files to facilitate the compromise of a system and the anonymous mode of FTP accelerates this process. Further, if users access the FTP server from a remote location, it becomes possible for someone to "sniff" their using their passwords from any point along the network connection. More restrictive file system permissions are set to reduce the number of administration utilities that the user can access by limiting read and execute privileges to the root user. The PATH environmental variable is restricted to remove the current directory identifier in the PATH statement. Users will typically include "."in their PATH variable to allow programs in the current directory to be executed by simply typing their name. Unfortunately, this also allows a Trojan horse program to be placed in your directory by a hacker in an attempt to execute commands of the hacker choice.

If you select paranoid server security, the following configuration items are set by default in addition to all of the items set in the moderate security configuration settings. The firewall is configured to block all services except those that you specifically allow. The SUID capability for the programs mount, umount, ping, at, usernetctl, and traceroute are all disabled. Although this might be inconvenient to the user, it greatly reduces the security risk if these daemons are compromised. The ability for users to submit cron jobs is disabled. Cron allows the user to submit jobs that will be executed at specific dates and times but is prone to abuse. For example, if a hacker compromises a system, they can install a cron job that will surreptitiously reinstate their status at a later time if the legitimate administrator discovers and removes the compromise. Limits are set on the amount of system resources that are available to each user. This reduces the likelihood that Denial of Service (DoS) attacks can disrupt the normal operation of the system by exhausting the system resources. Limits are set on each user to disable core files, limit the number of simultaneous processes to 150, and limit file sizes to 40MB each. Optionally, the limits can be assigned to specific programs. If the Apache web server is enabled, the ability for the web server to follows symbolic links is disabled. This will prevent web site user from viewing files that are located outside of the web page directories. Further, since the web server process runs as user "nobody", web users might be able to modify any world writable files in the file system. All print services are disabled by default, as there have been a number of security vulnerabilities discovered in the lpd service. Separate temporary directories will be created for each user on the system and replace the shared /tmp directory. Many programs make use of the /tmp directory for temporary file storage which is potentially dangerous on multi-user systems. By making each users temporary directory unique, the risks of one user corrupting another users files is reduced.

Bastille Workstation Configuration

When you select the workstation configuration, a number of settings are set by default that correspond to the lax workstation security configuration. Note that since most of these settings are similar to the server configuration, so I will not discuss these in detail. A system configured as a workstation typically will not need many of the daemons that a server has running.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

However, you are allowed to override the recommended settings if necessary. The SUID capability for the programs DOSEMU and the news server tools are disabled. Password aging is enabled. The single user mode is password protected using the root password. You can elect to apply limits by program or by user to minimize the chance that a denial of service attack will succeed. Additional logging is enabled to log kernel messages, error messages, and warning messages to a log file. Since it is unlikely that you will need to provide DHCP capability as a workstation, it is disabled. Likewise, it is unlikely that you need SNMP, DNS, or the Apache web server so they are all disabled. Sendmail's verify recipient existence (VRFY) and expand recipient alias/list contents (EXPN) commands are disabled. The current directory is disallowed from being used in the PATH variable. If you have decided to enable the Apache web server, server-side includes are disabled. The telnet and ftp programs are disabled.

If you select moderate workstation security, the following configuration items are set by default in addition to the items set above. The firewall is configured to block all unnecessary services. The dump, restore, cardctl commands SUID capability are disabled. The BSD remote access tools rcp, rlogin, and rsh are disabled. The APDM daemon, NFS, and Samba are disabled. Sendmail is configured to deactivate network listening mode, as the system does not need to respond as an email server. Separate temporary directories will be created for each user on the system and replace the shared /tmp directory.

If you select paranoid workstation security, the following configuration items are set by default in addition to all of the items set in the moderate security configuration settings. The firewall is configured to block all services except those, which you specifically allow. The SUID programs mount, umount, ping, at, usernetctl, and traceroute are all disabled. The ability for users to submit cron jobs is disabled. The PCMCIA startup script is disabled. All of the daemons except crond, syslog, keytable, network, gpm, and xfs disabled.

Bastille Configuration Summary

When you select server configuration, by default the following security measures are implemented:

- Disable SUID status from dump/restore, cardctl, dosemu, news server programs
- Enforce password aging
- Password protect single user mode
- Add additional logging
- Disable apmd, NFS, Samba, pcmcia, DHCP server, news server, routing daemons, NIS, SNMPD
- Disable VRFY/EXPN commands into sendmail
- Deactivate named (dns)
- Deactivate apache (web)
- Deactivate apache Server Side Includes (SSI)
- Deactivate telnet
- Deactivate ftp

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Depending upon which one of the three security levels you select for your server, the security configuration items shown in Table 1 will be implemented:

Security Level	**Lax**	**Moderate**	**Paranoid**
Firewall Strength	None	Medium	Strong
Disable SUID status from rsh, rlogin	No	Yes	Yes
Disable SUID status for mount, umount, ping, at, usernetctl, traceroute	No	No	Yes
Disable rhost-based authentication	No	Yes	Yes
Disable cron use to everyone but root	No	No	Yes
Enforce limits on resources to prevent DoS attack	No	No	Yes
Disable gpm	No	Yes	Yes
Deactivate apache CGI script execution	No	Yes	Yes
Deactivate apache's following of symlinks	No	No	Yes
Disable printing	No	No	Yes
Disable FTP user mode	No	Yes	Yes
Disable FTP anonymous mode	No	Yes	Yes
Activate TMPDIR protection	No	No	Yes
Default umask	022	022	077
Default security level	2	3	4
Default file permission level	2	3	4
Restrict "." from the PATH variable	No	Yes	Yes

Table 1 - Server configuration items as a function of security level

When you select workstation configuration, by default the following security measures are implemented:

- Disables SUID status to the news server tools and DOSEMU
- Setup password aging
- Password protects single-user mode
- Apply limits to any one program/user's resource usage, to block Denial of Service attacks
- Configure additional logging
- Deactivates the DHCP Server daemon
- Disable the SNMP daemons
- Disable the VRFY/EXPN commands in Sendmail
- Deactivate DNS server
- Deactivate Apache server
- Deactivate Apache Server Side Includes (SSI)
- Restrict "." from the PATH variable
- Deactivate telnet
- Deactivate ftp

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Depending upon which one of the three security levels you select for your workstation, the security configuration items shown in Table 2 will be implemented:

Security Level	**Lax**	**Moderate**	**Paranoid**
Firewall Strength	None	Medium	Strong
Disables SUID status to mount, umount, ping, at usernetctl, and traceroute	No	No	Yes
Disables SUID status to dump, restore, cardctl, rsh, rlogin and rcp	No	Yes	Yes
Disable rsh/rlogin access to this machine	No	Yes	Yes
Restrict use to cron to root account	No	No	Yes
Disable pcmcia startup script	No	No	Yes
Deactivates the APMd daemon	No	Yes	Yes
Disables NFS and Samba	No	Yes	Yes
Disables GPM	No	Yes	Yes
Deactivates Sendmail's network listening mode, so this WORKSTATION doesn't serve as a mail server	No	Yes	Yes
Deactivate Apache Server follow-symbolic links behavior	No	No	Yes
Deactivate Apache Server CGI's	No	No	Yes
Deactivate all remaining daemons, with the exception of crond, syslog, keytable, network, gpm, xfs and pcmcia	No	No	Yes
Default umask	022	022	077
Default security level	2	3	4
Default file permission level	2	3	4
Disable FTP's anonymous mode capability	No	Yes	Yes
Disable FTP's user mode capability	No	No	Yes
Apply TMPDIR protection	No	Yes	Yes

Table 2 - Workstation configuration items as a function of security level

Conclusions

The Bastille Linux scripts provide an effective means to securely configure your Linux server or workstation. Although Bastille currently only supports RedHat and Mandrake Linux distributions, there are plans to support Debian, SuSE, and TurboLinux in the near future. Bastille does a superb job of configuring your system to a configuration that is in line with the "best practices" of the information security field. In the interactive mode, Bastille provides the system administrator an invaluable tutorial on securing a system by educating you on the benefits and drawback of each of the configuration settings. In the non-interactive mode, Bastille automates the process of security hardening allowing you to easily harden large numbers of systems. The knowledge gained by running the Bastille scripts on Linux boxes provides valuable insight into protecting your Unix boxes. In conclusion, I highly recommend that you download and install Bastille Linux on all of your RedHat and Mandrake Linux boxes.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

References

Lucent Technologies. “The Creation of the UNIX Operating System.” 2000.
URL: http://www.bell-labs.com/history/Unix/ (December 2001).

Severance, Charles. “A Brief History of Unix.”
URL: http://www.hsrl.rutgers.edu/ug/unix_history.html (December 2001).

Stallman, Richard. “The GNU Project.” GNU’s Not UNIX! December 28, 2001.
URL: http://www.gnu.org/gnu/thegnuproject.html (January 2002).

Tanenbaum, Andrew. “Andrew S. Tanenbaum's Personal FAQ Sheet.”
URL: http://www.cs.vu.nl/~ast/home/faq.html (January 2002).

Linux International. “The History of Linux.” 2001.
URL: http://www.li.org/linuxhistory.php (January 2002).

Hasan, Ragib. “History of Linux.” November 4, 2000.
URL: http://ragib.hypermart.net/linux/ (January 2002).

Lasser, Jon. “Jon Lasser’s Personal Homepage.” July 23, 2001.
URL: http://www.tux.org/~lasser/ (January 2002).

Source Forge. “Bastille Linux.” June 11, 2001.
URL: http://www.bastille-linux.org/ (January 2002).

SANS Institute. “SANS/FBI Top Twenty Most Critical Internet Security Vulnerabilities.”
November 15, 2001. URL: http://www.sans.org/top20.htm (January 2002).

SANS Institute. “Intrusion Mailing List.” 2001.
URL: http://www.incidents.org/about.php (January 2002).

Security Focus Corporate Site. “SecurityFocus.”
URL: www.securityfocus.com (January 2002).

Hansen, Stephen and Atkins, E. Todd. “Centralized System Monitoring With Swatch.” July 30, 1992. URL: http://www.oit.ucsb.edu/~eta/swatch/lisa93.html (January 2002).

Atkins, E. Todd. “The Simple Watcher SWATCH.” November 8, 2001.
URL: http://www.oit.ucsb.edu/~eta/swatch/ (January 2002).

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 1st, 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