Global Information Assurance Certification Paper

Using Basic Security Module (BSM), Tripwire, System Logs, and Symantec's ITA for Audit Data Collection

There are 2 main sources of information regarding intrusion detection to a networked computer system: networking traffic and auditing files. A more effective intrusion detection system (IDS) is provided when a multi-layered security approach is incorporated into a network of computers. Whatever attacks not detected by one layer of the IDS tools, another security application may detect an attack enabling an administrator to take future action preventing further intrusions. An IDS can provide security through application of three basic techniques: detection of anomalous traffic patterns (an intruder's usage of an automated port scanning tool on the network), misuse of computer resources (like unauthorized user access to files and/or services), and passive (auditing data collection) and active methods (using Symantec's Intruder Alert program to automatically take an action upon the detection of a specified event). In this paper we will provide examples of all three IDS techniques applied to a network of SUN boxes using the Solaris 8 Operating System (OS). It is not intended for this essay to present a complete security system. The primary focus of this paper is to provide host based set of tools auditing trace records of attempted attacks on a secured network of Solaris boxes. Until recently UNIX systems were generally not known to have good auditing tools available. The National Security Agency (NSA) National Computer Security Center (NCSC) published the Rainbow Series of papers (http://csrc.nist.gov/secpubs/rainbow/) that promotes computer security. Computer security levels are defined in the "Orange Book" (http://csrc.nist.gov/secpubs/rainbow/nsaorder.txt). These levels are graded on a scale A (most secure) through D (least secure). Using default settings, most UNIX systems are rated at the C1 level (security discretionary) level. SUN has provided a new kernel auditing tool called the Basic Security Module (BSM). This tool can be activated to upgrade the security level of a Solaris 8 system to C2 (added auditing capability and access control). This tool provides kernel auditing and device allocation features. The disadvantage of using the BSM tool is that, in general, system performance can be reduced by about 10 per cent. Hard disc partition usage must also be regularly scrutinized to avoid filling up disc allocations that are populated with auditing data such as the /var partition. The advantage of an auditing tool processing at the kernel level is that the possibility of tampering with the auditing data by unauthorized intruders is greatly reduced. The disadvantage is that the output of the BSM tool is binary which requires more effort to use.

After enabling BSM, the volume manager is no longer available which disables any automatic mounting of portable media like diskettes, CD-ROMs, and magnetic tapes. The C2 environment demands control and monitoring of portable media access. In most cases simultaneous access to portable storage devices by multiple users should not be allowed. Users typically must walk to an area where portable device drives are available. It takes time to walk from a remote workstation to this area. If multiple user access were available it is feasible for an attacker to gain access to a storage device that contains sensitive data. After an authorized user is finished with the data, it is typical to deallocate the device for pickup. While the original user of that data is walking to pick up the sensitive data, there is sufficient time available for an unauthorized user to gain control of that data. If managed properly by the administrator, the device allocation system can prevent unauthorized access to portable data just after the authorized user has finished using the portable device. The administrator must apply the allocate, list_devices, and dminfo commands to monitor and control access to portable media.

The Tripwire program, provided from Purdue University, could be used to regularly check the permissions of all device files under the /dev and /devices directories. The Tripwire tool is available at ftp://cost.cs.purdue.edu/pub/tools/unix/Tripwire/ or http://www.tripwiresecurity.com/. This tool can

provide more auditing information regarding any file that has possible been tampered with by an unauthorized user. Tripwire includes a default configuration file that lists what files this tool is to monitor. Tripwire maintains a database of information about of all files listed in configuration monitoring file. The first time Tripwire is run this database is populated with the initial condition of all files listed in the configuration monitoring file. It is therefore very important for an administrator to install and run Tripwire immediately after initial Solaris 8 installation so that an accurate database of file information can be maintained without the possibility of undetected file tampering before Tripwire has been utilized. Subsequent execution of the Tripwire program compares the current state of all monitored files with the previous file state stored in the database. Any key file differences can be routed to alert interested user administrators and/or store to audit files. An intruder may change the permissions to key files to gain system access to data and/or services. The intruder may then change the file permissions back to the initial state in an attempt to cover his or her tracks. Depending on the functional use of each SUN box, Tripwire can be programmed to run at some regularly scheduled interval of time, like once every 24 hours, using the cron service as described below when running the BSM tool through the cron utility. Of course the Tripwire tool can be used to monitor other selected files besides those in the device related directories such as: /etc, /etc/init.d, /etc/rc0.d, /etc/rc1.d, /etc/rc2.d, /etc/rc3.d, /usr/sbin, and, /kernel.

To enable BSM execute the following commands and follow the subsequent actions:

```
cd /etc/security
./bsmconv
```

To actually start BSM execution reboot the system.

The BSM utility can be stopped using the 'bsmunconv' command. The BSM binary audit output file names include the time stamp of when the BSM auditing function was enabled after reboot. A typical file name follows 20010911211735.not_terminated.hostname

The first part of the name shows the timestamp of when the audit daemon was first enabled (yyyymmddhhmmss). The 'not_terminated' shows that the audit daemon, auditd, is still running since it was initiated or started. If the system is rebooted or the daemon restarted, the file name will reflect the new the timestamp. When the auditd daemon crashes, the output binary file name may also have the 'not_terminated' text preceding the hostname.

If the daemon is stopped using the 'audit –t' or the 'bsmunconv' commands, the output binary file name could be as follows: 20010912082935.20010914203505.hostname. This name shows a second time stamp specifying time when the daemon was gracefully stopped.

BSM is configured by editing files in the /etc/security directory. The BSM utility uses the auditd daemon and is controlled by what is specified in the /etc/security/audit/audit_control file. This file specifies the directory that the binary audit data should be sent and what general events that the BSM tool should audit. In this file is a line that contains the characters, 'dir:' that specifies the BSM binary output directory. The 'minfree:' text of the audit_control file specifies the minimum amount of free disk space in the binary output directory, in percentage, that must be available. If this minimum is less than the percentage figure, the script /ect/security/audit_warn is run. This script could be configured to send an email warning message(s) to the root or any administrator user account. A more detailed type of event binary audit trail can be collected for the a specified list of user accounts by editing the /etc/security/audit_user file. By default the audit_user file applies to the root user account. Another line in this file the characters, 'flags:' is where the user specifies the event types that are to be traced in the binary output file. The flag specification and meaning follow:

ad – administrative actions like mount or exportfs
all – collect all events
ap – application auditing
cl – close(2) system call
da – change in object access
dc – creation or deletion of any object
ex – exec(2) system call
fa – access object attributes
fm – change object attributes
fr – reading or opening for reading
fw – writing or opening for writing
fd – deletion of object
io – ioct(2) system call
ip – system V IPC operations
lo – login or logout events
no – null for no event trace
pc – process operations
nt – network events
na – non-attributable events
ot –any other events
p0 – minor privileged actions
p1 – unusual privileged actions

For purposes of this paper let us specify that the audit_control file has been edited as follows:

```
#ident  @(#)audit_control      01/09/01 SANS Paper
        #
        dir:/var/security
        flags: ad,lo,pc,-all,^-ot
        minfree:10
          naflags:lo
```

The 'naflags:' text specifies the classes of events to be recorded that cannot be associated to a specific user. The above audit_control file script specifies that administrative, logging, and processing operations will be recorded in the /var/security directory. The plus sign (default) in front of the class applies to recording successful events and the minus sign indicates event types to be recorded that have failed. The carrot (^) sign is a 'not' operator. The '-all' text specifies that any failed events will be recorded with the exception of a failed other (-ot) event. This script will send a warning message to the administrator when the audit directory is 90% full.

The BSM generates binary output files that contain auditing trace history data. BSM includes 2 tools to convert this binary data into a readable format: praudit and auditreduce. To display the text of the entire audit file simply input the following command at the Solaris prompt: auditreduce | praudit. To print the text of the entire audit file simply input the following command at the Solaris prompt: auditreduce |

praudit | lp. The auditreduce '-c' option specifies what class or classes of events are to be converted into readable format. The auditreduce '-O' option specifies the directory location and file name of the ascii audit output file. The options '-a' and '-b' provide the user to specify the display of any events recorded after and/or before any times. The following command captures any events after 8 am on September 12th and before 7 am on September 13th:
auditreduce –a 010912080000 –b 010913070000 | praudit.
The following command calls for all login events applicable to user dobbs attempted from August 1st through, and including, August 31st.
auditreduce -a 20010801 -b +31d -u dobbs -c lo | praudit

The praudit output provides a record of information for each event recorded. The praudit command allows the user to specify the binary file(s) generated by the BSM tool to translate data into a readable format. Standard input is the default input for the praudit command. Each record of the praudit output file is divided into at least three parts: the header, subject, and the return. The header will show the record length in bytes, the audit record version number, event description information, and a time and date. The subject section will provide information for unauthorized privilege escalation. The subject part is composed of the following information: subject, user audit ID, effective user ID, effective group ID, real user ID, real group ID, process ID, session ID, and terminal ID consisting of a device and machine name. The return part of the record provides the return token, the error status of the system call, and the system call return value. There could be additional tokens in a record like an argument, internet address, path information, socket, or path tokens, depending on how the BSM product is configured to collect auditing data.

Using scripts in sed or awk on the praudit output files can provide important analysis data to trace the history of an intrusive event. The script write should take advantage of the header, subject, and return tokens when selecting the lines from the praudit output file to inspect in the awk or sed output files. An attacker may commonly use specific commands like 'finger' or 'watch' to verify if an administrator is currently on the system. An attacker may scan the process table to see what utilities are running (like BSM). Simple intrusion detection scripts are available for use from various websites like http://www.securityfocus.com/focus/sun/articles/bsmaudit1.html, http://www.cs.rit.edu/~hpb/Man/_Man_Solaris_2.6_html/html1m/praudit.1m.html, or http://ftp.za.kernel.org/pub/linux/libs/security/Orange-Linux/refs/Orange.html,. The scenarios of possible intrusions available to an attacker can be endless and are beyond the scope of this paper.

There are all types of ways to implement the BSM event data into the system logging (syslog) files. A System log file is a recording of computer events. One approach will be shown here in the hope that any numerous other techniques can easily be considered. The Solaris 8 OS provides a system log daemon named syslogd that is used to receive and route system log events from syslog() calls and logger commands. This daemon can be used as another key security service that the Solaris 8 OS provides. The outputs of this daemon are configured and controlled by editing the '/etc/syslog.conf' file which is read at boot time. This file can be configured for the syslogd daemon to send messages to specific user names and/or specified system log files. Note that the default settings for the syslog.conf file is not optimal for all Solaris 8 installations. The administrator should consider editing this file to control and configure the syslogd daemon to best meet the security and functional needs of each SUN box. Typical default system log files that the syslogd sends security and error event messages to are:
/var/adm/sulog – log os the super user (su) command
/var/adm/log/aspplog – PPP software logs error and connections in this file
/var/log/syslog – error messages from sendmail are sent to this file

/var/adm/messages – errors generated from a variety of different daemons are sent to this file
/var/cron/log – the cron daemon sends a message to this file when it runs a scheduled job.

To read the security and/or error related messages sent to these syslog files, messages are categorized by facility and severity levels. Facility categories include the following examples: lpr (printer system), auth (generated by authorization applications like login, su, and getty), kern (messages from the system kernel), user (messages from the default facility that do not fit into any of the other listed facilities), news (messages from the UseNet News system), cron (messages from the cron and/or at services), daemon (messages from system daemons such as inetd or lpsched or local0-7 (messages from up to 8 locally defined categories). Severity message levels are listed as follows: emergency (emerg), alert (alert), critical (crit), error (err), waringing (warning), notice (notice), informational (info), and debug (debug).

A system administrator may choose the cron to use the cron utility to start, stop, and then restart the BSM tool once every 24 hours. The audit binary file is then used to generate a single daily audit ascii file using auditreduce and praudit. The audit binary and ascii output files are placed into one of twelve audit folders, one for each month of the year, for future analysis if and when the need arises. Therefore each monthly audit folder will have up to 31 different binary and 31 ascii folders.

The cron process must be properly configured to prevent easy foreign intrusion attempts. The cron utility is configured using the crontab, cron.deny, and cron.allow files in the /etc/cron.d directory. The administrator could specify in the cron.deny file that 'all' and 'nobody' users do not have access to the crontab file. The cron.allow file should include only the privileged administrator(s) like root. It is better yet to create a user account with all the root privileges so that an intruder will have a more difficult time guessing what account has all the root privileges. The administrator can create a file containing cron process commands to implement this proposed audit plan. The administrator should change into a directory with only superuser/root read privileges. Assume the name of this ascii file is named /super/bsm_cron. The enter the command 'crontab –l > /super/bsm_cron. Using the vi editor, add the following script into the bsm_cron file.

```
52 23 * * * cd /etc/security
53 23 * * * ./bsmunconv
54 23 * * * auditreduce –c lo –O /etc/security/current_month/start_time.end_time.host.txt
56 23 * * * mv start_time.end_time.host.txt /etc/security/current_month/start_time.end_time.host.txt
57 23 * * * mzip /etc/security/current_month/start_time.end_time.host.txt
59 23 * * * mv start_time.end_time.host.txt /etc/security/current_month/start_time.end_time.host.txt.Z
1 0 * * * ./bsmconv
```

Be advised that usage of character substitution has not been included in the above script. For example, the second fields can be replaced in the audit output file names with the asterisk or wild character since the exact time the BSM utility has been started and stopped may not be exactly known. The file names can be constructed through concatenation using character substitution to construct the file name like 'date +%Y%m`00`*`.`date +%Y%m%H*`.hostname'
(where %Y is the 4 digit number for the current year, %m is the 2 digit number for current month, and %H is the 2 digit number for the hour). A 3-ascii character month folder name (like Jul July or Sep for September) can be specified with the following command: `date +%h`
This script will first stop the BSM utility, start it again, and then generate an audit ascii file of any logging attempts made during the last 24 hours. The start time will be the time the BSM utility was started at 0001 early in the morning. The end time will be the 2353 time when the BSM utility was stopped late in the evening. To actually include this file into the cron table, simply enter the following command:

crontab < /super/bsm_cron.

This approach can provide organization to any audit file of interest for future reference. Partial backup operations can be easily scripted to target particular months of the year for archiving. If an intrusion of any particular type has been discovered, the administrator can easily go back to a binary file for further analysis using the auditreduce and praudit tools. The analysis may require the administrator to take further actions to prevent future intrusions and/or to include more particular auditing trace data collection pertinent to the type of intrusion that has been detected. Consequently the crontab file may be changed to include more detailed binary collection or ascii conversion of particular event types like processing or change access to object events.

One could incorporate the Symantec Intruder Alert (ITA) product to work in concert with the Solaris BSM tool. Auditing text data could be dumped into a syslog file that is monitored by the ITA product. Depending on how often the agent is configured to include the selective praudit and auditreduce output data into the host system log files, suspicious intrusive events in near real time could be identified by the ITA software. For example, ITA ccould be configured to generate warnings to be posted on the ITA Windows console.

The ITA product is designed to include intrusion detection functionality on all hosts in a computer network. Symantec's ITA software is installed on different computers that perform one of 3 distinct functions. The Intruder Alert software is composed of at least one ITA agent, at least one manager, and at least one console that provides the Graphical User Interface (GUI).

The ITA manager houses a database of events that the ITA software is configured to capture through the use of policies. The ITA product can be distributed to include one or more managers. A master list of policies are stored on the manager(s). Policies are constructed with one or up to three categories of elements: select criteria, ignore criteria, and action criteria. The select criteria specifies the event that is to be captured. The ignore criteria provides any exceptions that the selection rules would otherwise collect. The action criteria specify any active computer response that is commanded when a recognized event occurs. For example the ITA policies can then be configured to search for particular character string patterns generated in the auditreduce and praudit output text files and then sent to the system log files. The manager also save a list of policies and domains applied for each agent. The manager also provides a secure interface to the event viewer, administrator, and all agents.

Assume the administrative console is installed on a computer running the Windows operating system. This console provides the administrative functions where domains and policies can be set. An ITA domain is simply any user friendly method of grouping together a set of processing nodes on a network. For example one domain may be named UNIX which includes all UNIX agents. Another domain may be named Windows98 to include all agents running the Windows 98 operating system. Depending on the complexity and size of the network being protected, a sensible approach can easily be constructed to properly organize a network of computers. The administrator can also connect and/or disconnect to ITA managers, manage ITA user accounts and privileges, and configure ITA mangers and agents.

The event viewer is commonly installed on the same machine as the administrative console. The event viewer also sends commands to agents. The event viewer can display selected event information stored on the manager. The event viewer provides the analysis and reporting functions of suspicious events that match a policy.

An ITA agent is simply any computer that has the ITA agent software installed and is registered to interface with one or more ITA managers. The ITA agent host monitors itself and reports back to the manager any suspicious intrusion activity that matches the policies assigned to that host. Any event that matches the policy criteria is reported back to the manager(s) and is saved on the ITA manager database.

The policies on an agent may also drive the host to take some sort of action after it recognizes an activity that may be suspicious. For example, after 3 failed logon attempts within 3 minutes, the computer may disable the logon panel for 5 minutes to prevent the usage of automated login attacking tools like crack.

There are many possible scenarios that could qualify as an attempted intrusion on a host. One TCP traffic pattern that would qualify as a possible intrusion attempt is repeated attempts to access numerous services and/or ports from the same host. A series of events like this could be enough to alert the administrator to a possible hostile reconnaissance of the host computer. An attacker will typically look for any open ports that a service on a computer is using to gain access to a network. An open port may provide key information that an attacker can use to gain unauthorized access to network resources. The first 1024 ports (0-1023) on a computer are reserved or allocated to standard protocols as specified in the current Request for Comments (RFC's) listed on the following URL address: http://www.rfc-editor.org/rfc.html. The RFC's are public documents that available for anyone to reference, including potential attackers. Those port numbers above 1023 are available for any use or protocol. Once an attacker has performed a port scan, subsequent attacks can be focused on a host to reconnoiter the open ports discovered on the first attack. Depending on the message response pattern an intruder receives, more particular information can be gained regarding the software version number of services and/or daemons running on the SUN box. The more specific information an attacker receives, the easier it becomes to gain unauthorized access to network resources.

It is important that an administrator carefully consider what the primary functional needs are for each processing host on a network. After installation of a Solaris 8 system, the default setting enables most of the common standard services on the SUN box. For example, services/daemons like http/httpd (uses port 80), ftp/ftpd (uses port 21), or telnet (uses port 23) should be disabled unless absolutely necessary to meet a primary requirement allocated to a host computer. Disabling unneeded services can be implemented by commenting out the daemons and/or services listed in the /etc/inetd.conf and /etc/services files. The /etc/inetd.conf file contains the list of all daemons running on the host Solaris 8 box. The /etc/services file contains a list of services and the associated network ports used for each service enabled on the host SUN box. It is easy to see the valuable information that is stored in these files that would be of interest to an attacker.

If a service is necessary to be enabled, the administrator is advised to at least configure and limit the access of the service to authorized users only. One example of properly configuring a service was described previously regarding the cron service. Most of the Solaris 8 services can be configured by editing the contents of a file(s) applicable for each required service. To control the access to these key configuration files, the administrator should limit read access to the root and/or authorized administrator user accounts. The service configuration file could contain important information that an intruder could use in later attacks.

ITA provides many 'canned' policies that a network administrator may wish to monitor for possible attempts. However ITA also provides the network administrator the capability to customize a policy by monitoring for a particular pattern of character strings that may appear in any of the system log files being monitored on an agent host. These character string patterns could also be configured to include a sequence of characters that is picked up by the BSM monitoring tool. For, example, the script below could be included into the crontab file to be executed once an hour every day, just before the hour is passed

```
59 * * * * auditreduce –a`date +%y%m%d%H`0000 –b `date +%y%m%d%H%M`00 –O
syslog_filename | praudit
```

Many attempted attacks can be detected by first clarifying the primary functional requirements of a

network. Knowing what is required to meet system needs will enable an administrator to determine the kind of normal traffic patterns that can be expected to appear on the network to be secured. Any traffic patterns that show up outside the boundaries of the normal realm can be considered for inclusion in the ITA policies, Solaris 8 system log events, and/or BSM configuration options for recording important auditing data. Ideally hosts on a network should be categorized into groups depending on their functional use. The idea is to configure each group using the same host hardening scripts to greatly simplify the implementation of security policies driven by the higher level system requirements. Writing a solid set of requirements for a network can easily be an arduous and time consuming task. System requirements work should be done as early as possible to avoid expensive redesign of security policies after the system is already built. However once written, many unforeseen problems that consume both time and effort can be avoided to keep both schedules and budgets within acceptable limits.

References:

http://www.securityfocus.com/frames/?focus=ids&content=/focus/ids/articles/idsbsm.html
Intrusion Detection using Solaris' Basic Security Module
by David Endler last updated Friday, July 14, 2000

Solaris Security by Peter H. Gregory, 2000 by Sun Microsystems Press; A Prentice-Hall Title, Upper Saddle River, New Jersey 07458, ISBN 0-13-0960553-5, pages 50, 87, 91-94, 487-490

The Complete Reference Solaris 8 by Paul A. Watters and Sriranga Veeraraghavan, 2000 by Osborne/McGraw-Hill, 2600 Tenth Street, Berkeley, CA 94710, ISBN 0-07-212143-2, pages 487-489

Solaris for Managers and Administrators, By Curt Freeland, Dwight McKay, and Kent Parkinson, Third Edition, 2000 An OnWord Press Pulication, Albany, New York 12212-15015, ISBN 0-7668-2137-4, pages 595-597

http://www.sisu.se/~seger/doc/sunos/sunman/audit_control.5.html (SunOS manual pages can be browsed from the top-level index) Copyright 1994-2001 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA SunSHIELD Basic Security Module Guide

http://answerbook.cs.nmsu.edu/ab2/coll.47.11/SHIELD/@Ab2PageView/4930?
2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto CA 94303 USA
http://answerbook.cs.nmsu.edu/ab2/coll.47.11/SHIELD/@Ab2PageView/6278?
2000 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA

http://developer.novell.com/research/sections/reviews/inreview/2000/may/i000502.htm
Foiling Hackers with Axent's Intruder Alert

http://www.delrina.com/avcenter/security/Content/2000_06_28.html
June 28, 2000 Intruder Alert 3.5 ITA Shared Actions Policy Update

Last Updated: July 24th, 2017

Upcoming Training

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