Global Information Assurance Certification Paper

Router Audit Tool: Securing Cisco Routers Made Easy!

Brian Stewart
March 29th, 2002
Version 1.3

Introduction

Securing and auditing Cisco routers has always been a time consuming task. There are many steps required to modify the default configuration of a Cisco router. In a large network environment it could take hours to confirm that routers are securely configured. Imagine a tool that could reduce this process to only a few minutes. The Center for Internet Security (CIS) has provided a tool to do just that. The Router Audit Tool or rat was designed to help audit the configurations of Cisco routers quickly and efficiently.

The Router Audit Tool performs a baseline test on the configuration of a Cisco Router. The baseline Level 1 is modeled on the *Router Security Configuration Guide* published by the National Security Agency (NSA) of the United States. The tool provides a list of the potential security vulnerabilities discovered in an easy to read format. It even provides a list of commands to be applied to the router in order to correct the potential security problems discovered.

This document will discuss the need for a tool like rat and it's function. The installation and quick start guide sections provide all the information necessary to get started using rat. For those seeking more detail, a step-by-step narrative to using and customizing the Router Audit Tool is included. It includes samples of how to quickly reduce the security vulnerabilities of a new router and customize the rat installation. Given the excellent work of those before me, this document will not cover specific security requirements for securing Cisco Routers. Instead it will focus on the use of the Router Audit Tool to assist in maintaining a minimum standard configuration. For more information on security Cisco Routers, the references section of this paper contains links to several excellent resources.

Background

Routers are a crucial part of any network infrastructure. Without routers the Internet as we know it would not exist. According to the NSA "Routers provide services that are essential to the correct, secure operation of the networks they serve. Compromise of a router can lead to various security problems on the network served by that router, or even other networks with which that router communicates."[i] This makes routers an ideal target for attackers.

"When Cisco says, "Virtually all Internet traffic travels across the systems of one company, Cisco Systems," it's not marketing-speak."[ii] This single statement highlights the need to increase the security of Cisco devices. Unfortunately, Cisco routers are delivered with a default configuration that includes several potential security vulnerabilities. Many network administrators have little or no knowledge of how to properly secure routers. Securing routers is made more difficult by the fact that default configuration values are not shown in the configuration. Extensive research is required to determine all the default values that must be changed to improve security.

Fortunately, the NSA has created guidelines that provide a wealth of information on potential security vulnerabilities and the configuration changes required to correct them. However, this document is 248 pages long and requires a significant amount of time to read and implement. The developers of the Router Audit Tool have created an easy to use tool that can confirm that a router

is in compliance with the recommendations made by the NSA. The rat program defines compliance with the NSA guidelines as Level 1 (basic) security. There are additional rules provided to increase the routers security to Level 2 (intermediate/advanced) security.

There have been several papers written about increasing the level of security on Cisco routers that can be found in the SANS Reading Room, links to several of these are provided in the references section. Cisco Systems has a web page devoted to increasing router security, located at: http://www.cisco.com/warp/public/707/21.html. Rob Thomas also provides additional information on securing router configurations on his web page at http://www.cymru.com/~robt/Docs/Articles/secure-ios-template.html.

Based on these and other documents, many corporations have drafted internal guidelines for deploying standard router configurations. Confirming compliance with these standards often involves the use of a manual checklist. The Router Audit Tool provides the ability to automate the audit of corporate standard guidelines. The configuration files for rat can be customized to meet the needs of any organization. This allows corporations to confirm compliance with their guidelines and eliminate the human error involved with a checklist audit.

The Center for Internet Security is a non-profit organization devoted to improving the security of the Internet. Founding members include a virtual "who's who" of networking organizations; SANS, ISC2, ISACA, IIA, AICPA. Membership is available to both organizations and individuals. "The Center for Internet Security's mission is to help organizations around the world effectively manage the risks related to information security. CIS provides methods and tools to improve, measure, monitor, and compare the security status of your Internet-connected systems and appliances, plus those of your business partners."[iii] The Center for Internet Security has created benchmarks for Windows 2000, Solaris and most recently Cisco routers. These benchmarks are based on industry recognized best practices.

How Does the Router Audit Tool Work?

Router Audit Tool is a Perl program. It consolidates 4 other Perl programs; snarf, ncat, ncat_report and ncat_config. Snarf is used to download the configuration files from the router. Ncat reads the rule base and configuration files and provides output in a text file. Ncat_report creates the html pages from the text files. Ncat_config is used to perform localization of the rule base. All of the components of rat are licensed under the GNU General Public license. The GNU licensing is one of the prominent benefits of the Router Audit Tool allowing the open modification of the program. The guidelines for the rat are licensed to the NSA under the terms included in the install package. The rules and baseline document are licensed by the Center for Internet Security.

Rat performs an audit by comparing text strings in the configuration file from the router with regular expressions in the rules. Each rule has either a required or forbidden regular expression element. Based on this element rat determines if a rule is passed or failed. Due to the use or regular expressions, the rat rule base is extremely flexible. Currently CIS differentiates between Level 1 and Level 2 audits. The Level 1 audit is based on the NSA guidelines. The Level 2 audit includes additional tests from several sources including Cisco. The majority of the rules are for the protection of the router. There are, however, several rules that provide limited protection to the networks they serve. Additional rules can be added to the rule base with relative ease. This allows rat to work with any configuration.

Installation

Version 1.1 of the Router Audit Tool will function on Unix, Linux and on the Microsoft Platform with

ActiveState or Cygwin. The tool can be downloaded from The Center for Internet Security http://www.cisecurity.org. Installing rat is fairly simple and the documentation provided in the INSTALL.txt file is easy to follow.

Before installing rat it is important to decide if snarf is going to be used to download the configuration files or if a text file will be provided that already contains the configurations. In order to download the configurations it is necessary to provide snarf the username and passwords for the router. As suggested in the INSTALL.txt file, let paranoia be the guide, is it really safe to type router passwords into a freeware tool? If you aren't willing or able to confirm the source code for snarf is safe, perhaps it's best just to download the configuration files some other way!

The Router Audit Tool requires that Perl be installed on the system. For these examples a Linux system with RedHat 7.2 and a standard installation including Perl was used. If there are no plans to use snarf, version 1.1 offers the ability to install rat without snarf through the use of the alternate Perl makefile.

```
perl Makefile-nosnarf.PL [PREFIX=/home/you]
make
make test
make install
```
[iv]

In this example, snarf was going to be used at least occasionally to download configurations. Four Perl modules must be added to the system in order for it to function. The modules can be installed using the Perl CPAN module and FTP. Modules can also be installed manually. In this case the modules were downloaded from the URLs (included in the INSTALL.txt file) and manually installed. This was due to the site firewall blocking all FTP traffic. After installing the Perl modules rat can then be installed.

```
perl Makefile.PL [PREFIX=/home/you]
make
make test
make install
```
[v]

Quick Start Guide to the Router Audit Tool

Now that rat is installed, a quick overview of the steps necessary to begin using rat immediately. The first step in using rat is to "localize" the rule base to your site. Rat requires some site-specific information in order to function properly. The ncat_config program is used to perform this step. Ncat_config will ask a number of questions to assist in generating a more accurate rule base. It is important that you have the site information available before running the ncat_config program. It can be helpful to have a printed copy of the router configuration available.

After localization determine if router configurations will be downloaded using snarf or if text files containing the configurations will be used. In order to use text files it will be necessary to either download them using tftp or manually cut and paste the output from a telnet or console session. Next run rat and view the output using a text viewer or a web browser. The process sounds simple and for the most part it is. Help is available at every step using the "-?" flag after any of the commands. The man pages provided with rat also contain a wealth of information. Below is a quick sample of what a first time use of rat will look like.

```
# ncat_config
ncat_config: Reading /usr/etc/ncat.conf.MASTER

Please answer the questions below about your network and router configuration.  Type ? to get a short
explanation of any parameter.  If you are unsure about what value to give for a parameter, hit RETURN to take
the default value.
```

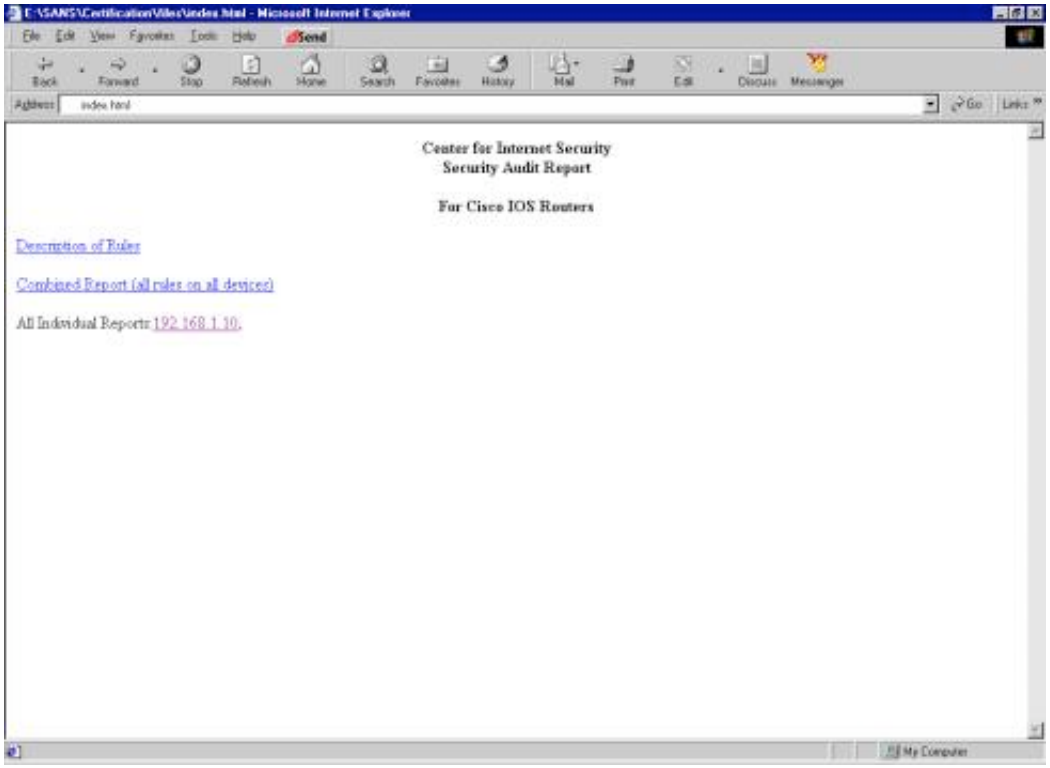
```
Select types of optional rules to be applied:
ncat_config: Apply rules for class use_multiple_ntp_servers [no] ?
ncat_config: Apply rules for class exterior_router [no] ? yes
ncat_config: Apply rules for class tacacs_aaa [no] ?
ncat_config: Apply rules for class localtime [no] ?
ncat_config: Apply rules for class gmt [no] ?
ncat_config: Apply rules for class snmp [no] ?
ncat_config: Apply rules for class exterior_router_with_2nd_if [no] ?

Change default configuration values:
ncat_config: Enter value for local_acl_num_egress [181] ?
ncat_config: Enter value for local_acl_num_ingress [180] ?
ncat_config: Enter value for local_acl_num_vty [182] ?
ncat_config: Enter value for local_address_internal_netblock_with_mask [192.168.1.0 0.0.0.255] ? 37.1.1.1
0.0.0.3
ncat_config: Enter value for local_address_loopback [192.168.1.3] ?
ncat_config: Enter value for local_address_ntp_host [1.2.3.4] ?
ncat_config: Skipping local_address_ntp_host_2 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Skipping local_address_ntp_host_3 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Enter value for local_address_syslog_host [192.168.1.3] ?
ncat_config: Enter value for local_address_telnet_acl_block_with_mask [192.168.1.0 0.0.0.7] ? 192.168.1.0
0.0.0.15
ncat_config: Enter value for local_address_telnet_acl_host [192.168.1.254] ?
ncat_config: Enter value for local_exec_timeout [5 0] ?
ncat_config: Enter value for local_external_interface [Ethernet0] ? Serial0
ncat_config: Skipping local_external_interface_2 because none of the prerequisite classes
(exterior_router_with_2nd_if) were selected.
ncat_config: Enter value for local_gmt_offset [0] ?
ncat_config: Enter value for local_loopback_num [0] ?
ncat_config: Enter value for local_timezone [GMT] ?

ncat_config: Writing /usr/etc/ncat.conf...Done.
ncat_config: Now examine /usr/etc/ncat.conf.
ncat_config: Edit /usr/etc/ncat.conf.MASTER and rerun ncat_config if not satisfactory.
# rat -a 192.168.1.10
snarfing 192.168.1.10...Hit Enter if no username is needed.
Username: root
User Password:
Enable Password:
done.
auditing 192.168.1.10...done.
ncat_report: Guide file rscg.pdf not found in current directory.  Searching...
Linking to guide found at /usr/doc/rscg.pdf
ncat_report: writing 192.168.1.10.ncat_fix.txt.
ncat_report: writing 192.168.1.10.ncat_report.txt.
ncat_report: writing 192.168.1.10.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

Router Audit Tool Output

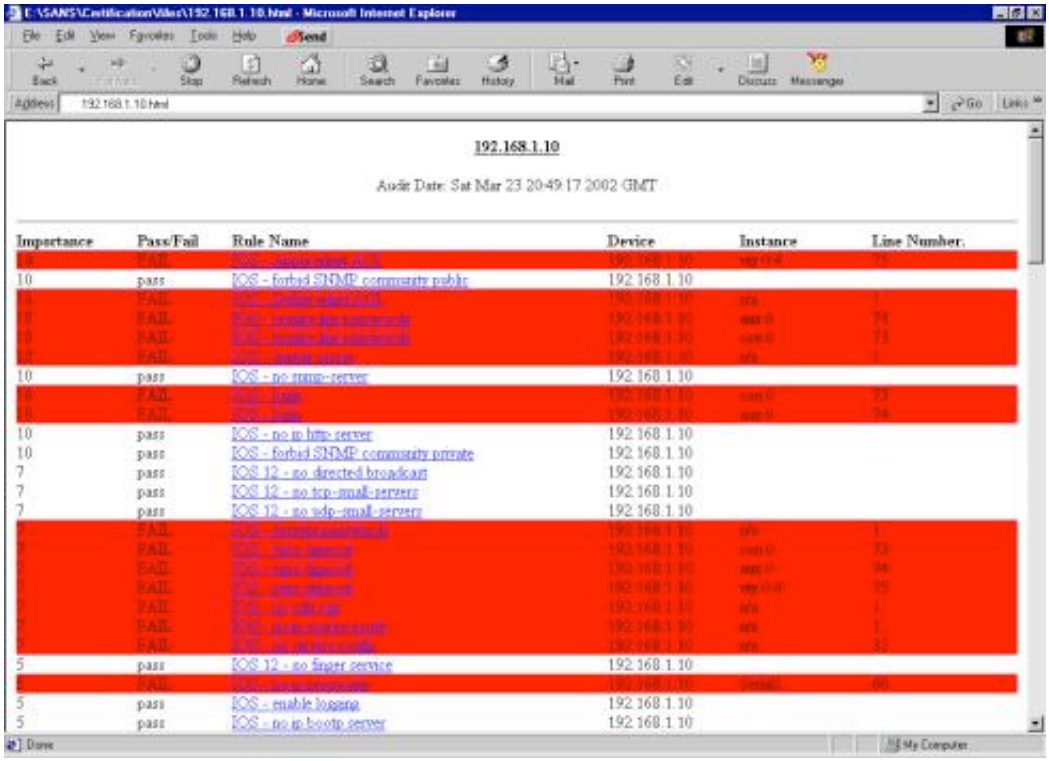
The Router Audit Tool creates a number of text and html files. First take a look at the files test.log and 192.168.1.10. Both of these files contain complete copies of the router show version and show configuration commands. These files should be deleted or sanitized immediately as they contain passwords to the device! The cisco-ios-benchmark.html and rscg.pdf files are links to the CIS and NSA documents on which rat is based.

The index.html file contains links to the description of rules page and links to the results of the test for each individual device. This Rules page is automatically generated from the ncat.conf files and contains the rules applied to the tested devices. For each rule listed there is a brief description of the security vulnerability. Each rule has a link to the *Router Security Configuration Guide* or other appropriate document for additional information. (See sample index.html below)

The individual device results page includes a pass/fail chart listing possible configuration vulnerabilities. The pass/fail results are easy to discern given the use of color. Each test result contains a link to the rule's description on the rules page. The device results include a weighted score. The weighted score is based on each rule having a weight, or importance, value between 1 and 10. Critical rules are given higher weights. Login passwords hold a weight of 10. Missing passwords will create a major security issue. Conversely logging critical messages to the console has a value of 3. Not logging critical messages to the console does not specifically create a vulnerability. The final output is a list of configuration commands required to correct the detected security issues. (See sample 192.168.1.10.html below)



Index.html



192.168.1.10.html

Now to examine the text files created by rat. The file 192.168.1.10.ncat_report.txt contains the same pass/fail and weighted score information found in the 192.168.1.10.html file in a text format. (For brevity much of the output is omitted)

```
# more 192.168.1.10.ncat_report.txt
Importance Pass/Fail  Rule                                Device          Line#       Instance
10         FAIL       ios - apply telnet acl              192.168.1.10    75          vty 0 4
10         FAIL       ios - define telnet acl             192.168.1.10    1           n/a
10         pass       ios - forbid snmp community private  192.168.1.10
7          pass       ios 12 - no directed broadcast      192.168.1.10
7          FAIL       ios - no cdp run                          192.168.1.10    1           n/a
7          FAIL       ios - no ip source-route            192.168.1.10    1           n/a
5          FAIL       ios - no ip proxy-arp               192.168.1.10    60          Serial1
5          pass       ios - no ip bootp server            192.168.1.10
5          FAIL       ios - no ip proxy-arp               192.168.1.10    52      FastEthernet0
3          FAIL       ios - logging trap debugging        192.168.1.10    1           n/a
3          FAIL       ios - logging console critical      192.168.1.10    1           n/a
Summary for 192.168.1.10
#Checks          #Passed          #Failed          %Passed
37                10                27                27
PerfectWeightedScore     ActualWeighedScore      %WeightedScore
252                              76                      30
Overall Score (0-10)
3
Note: PerfectWeightedScore is the sum of the importance value of all rules.
ActualWeightedScore is the sum of the importance value of all rules passed,
minus the sum of the importance each instance of a rule failed
```

The sample configurations for correcting the vulnerabilities can be found in the 192.168.1.10.ncat_fix.txt file. The commands may require some additional manipulation in order to avoid any compatibility issues. In many cases site-specific data must be inserted into the configurations before use. These can easily be found by looking for the phrase “EDIT-BY-HAND”. In the cases of adding access-lists it may also be necessary to configure the interface to which they should apply by adding “access-group # inlout” to the proper interface configuration. (Output

is omitted for brevity.)

```
# more 192.168.1.10.ncat_fix.txt
line vty 0 4
access-class 182 in
exit

no access-list 182
access-list 182 permit tcp 192.168.1.0 0.0.0.7 any eq telnet
access-list 182 permit tcp host 192.168.1.254 any eq telnet
access-list 182 deny   ip any any log
!
!line aux 0
!password EDIT-BY-HAND
!exit
!
!line con 0
!password EDIT-BY-HAND
!exit
line con 0
login
exit
```

The Basics of Using rat

The Quick Start Guide provided the necessary information to get started using rat, now it's time to investigate in greater detail how it works and how to better customize it to a specific application. After installing rat, a test run was made to confirm that everything was working properly. This also offered an opportunity to get more familiar with the tool before performing any customization. For the first test, a configuration was imported directly from the router using snarf. To simplify things, rat was executed from the directory where the files were to be stored. The –a flag was used to force rat to use snarf. The Router Audit Tool prompted for the router username, user password and enable password. After providing the information, rat connected to the router and performed the baseline audit.

```
# rat -a 192.168.1.10
snarfing 192.168.1.10...Hit Enter if no username is needed.
Username:
User Password:
Enable Password:
done.
auditing 192.168.1.10...done.
ncat_report: Guide file rscg.pdf not found in current directory.  Searching...
Linking to guide found at /usr/doc/rscg.pdf
ncat_report: writing 192.168.1.10.ncat_fix.txt.
ncat_report: writing 192.168.1.10.ncat_report.txt.
ncat_report: writing 192.168.1.10.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

A listing of the files created by rat:

```
# ls
192.168.1.10
192.168.1.10.html
192.168.1.10.ncat_fix.txt
192.168.1.10.ncat_out.txt
192.168.1.10.ncat_report.txt
all.html
all.ncat_fix.txt
```

```
all.ncat_report.txt
cisco-ios-benchmark.html
index.html
rscg.pdf
rules.html
test.log
```

Router Audit Tool Localization and Customization Features

The default commands for rat provide great results. However, the basic rules for rat are primarily designed to prevent unauthorized access, enable logging and prevent potentially dangerous services. For rat to function properly it must check for access-lists, addresses of internal servers and other site-specific information. In Version 1.0 this information had to be manually entered into the rule base. Version 1.1 has made this process much easier by adding a localization feature. This feature allows rat to quickly modify the rule base adding customized rules.

The program ncat_config is used to gather the required information for localizing rat. It uses this to create a new ncat.conf rule base file. Ncat_config can also be used to specify which rule classes should be applied. "Until you run this program certain rules will fail, such as ingress/egress filtering, VTY ACL definitions, syslog and time hosts, etc."[vi] Ncat_config applies the customization to the ncat.conf.MASTER file to create the ncat.conf file used by rat. The LOCALIZE.txt file contains additional information on customizing ncat_config. It is possible to further customize rat by modifying the localization script. A sample of the default ncat_config program is shown.

```
# ncat_config
ncat_config: Reading /usr/etc/ncat.conf.MASTER

Please answer the questions below about your network and
router configuration.  Type ? to get a short explanation of
any parameter.  If you are unsure about what value to give
for a parameter, hit RETURN to take the default value.

Select types of optional rules to be applied:

ncat_config: Apply rules for class use_multiple_ntp_servers [no] ?
ncat_config: Apply rules for class exterior_router [no] ? yes
ncat_config: Apply rules for class tacacs_aaa [no] ?
ncat_config: Apply rules for class localtime [no] ?
ncat_config: Apply rules for class gmt [no] ?
ncat_config: Apply rules for class snmp [no] ?
ncat_config: Apply rules for class exterior_router_with_2nd_if [no] ?

Change default configuration values:

ncat_config: Enter value for local_acl_num_egress [181] ?
ncat_config: Enter value for local_acl_num_ingress [180] ?
ncat_config: Enter value for local_acl_num_vty [182] ?
ncat_config: Enter value for local_address_internal_netblock_with_mask [192.168.1.0 0.0.0.255]?
ncat_config: Enter value for local_address_loopback [192.168.1.3] ?
ncat_config: Enter value for local_address_ntp_host [1.2.3.4] ?
ncat_config: Skipping local_address_ntp_host_2 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Skipping local_address_ntp_host_3 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Enter value for local_address_syslog_host [192.168.1.3] ?
ncat_config: Enter value for local_address_telnet_acl_block_with_mask [192.168.1.0 0.0.0.7] ?
ncat_config: Enter value for local_address_telnet_acl_host [192.168.1.254] ?
ncat_config: Enter value for local_exec_timeout [5 0] ?
```

```
ncat_config: Enter value for local_external_interface [Ethernet0] ?
ncat_config: Skipping local_external_interface_2 because none of the prerequisite classes
(exterior_router_with_2nd_if) were selected.
ncat_config: Enter value for local_gmt_offset [0] ?
ncat_config: Enter value for local_loopback_num [0] ?
ncat_config: Enter value for local_timezone [GMT] ?

ncat_config: Writing /usr/etc/ncat.conf...Done.

ncat_config: Now examine /usr/etc/ncat.conf.
ncat_config: Edit /usr/etc/ncat.conf.MASTER and rerun ncat_config if not satisfactory.
```

Often in security Audits the goal is to determine only if a router is vulnerable to specific types of attacks. Rat distinguishes between rule types by using classes. The default class is equivalent to the NSA guideline requirements. Additional classes have been added for additional security features. These classes can be used to specify the types of rules checked. This allows an Audit to be better refined and to save valuable time. In the case of the recent SNMP vulnerabilities rat allowed security teams to rapidly check hundreds of routers for potential vulnerabilities. Only the SNMP class was tested in order to reduce the amount of time required to perform an enterprise wide audit. In addition to the classes provided with rat custom classes can be created.

```
default          - default rules according to NSA Router Security Config Guidelines. Note that
                       rules that are in the "default" class are also in other classes.
access           - Access control related items such as vty/con/aux settings, passwords, servers,
                       SNMP, ACLs controlling management.
logging          - Logging related functions
routing          - Routing related functions
services         - services
tacacs_aaa       - Rules for using AAA (TACACS+ or radius). None of these are default.
exterior_router  - rules applicable to exterior routers
snmp             - rules that apply to SNMP[vii]
```

Classes can be selected using the ncat_config program or by using command line input when executing rat. The Router Audit Tool will normally only check rules from the default class.

```
-c, --limitclassto
        The `--limitclassto' allows the command line specification of a regular
        expression to limit the rules that are checked. The class of the rule
        must match the regexp specified or the rule is skipped. You might try
        something like

          --limitclassto=access
          --limitclassto=localrules
          --limitclassto=access,logging,aaa
          --limitclassto='access\|logging\|localrules'

        See the rules file for definition of rule classes. By default, only
        rules matching the class "default" are checked. "all" is synonym for
        ".*". You can give a "normal" comma separated list of classes that you
        want to check because "," is treated as a synonym for the regular
        expression or ("|").[viii]
```

As is often the case in an Audit, there is no interest in knowing the rules that routers have passed. Instead a list of the rules that fail is usually desired. The Router Audit Tool output can be modified to provide only a list of rules that are failed. This is done using the command line flag -f.

```
# rat -f 192.168.1.10
```

-

Step by Step to quickly increase the security of a new router.

The goal of this guide is to create a basic script to apply to a new Cisco router in order to quickly improve the level of security. This example uses a new Cisco 1720 router with 12.2.7b IP/FW/IDS PLUS IPSEC 3DES IOS. The configuration is that of a very basic single office environment. IP addresses have been assigned to one external interface, the Serial 0 port connected to an ISP, and one internal interface, Ethernet 0, for the office network. In addition Network Addresses Translation (NAT) has been enabled for users to access resources on the Internet. The routers initial configuration is show below.

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
hostname Router
enable password cisco
memory-size iomem 25
ip subnet-zero
!
ip audit notify log
ip audit po max-events 100
ip ssh time-out 120
ip ssh authentication-retries 3
!
interface FastEthernet0
 ip address 192.168.1.10 255.255.255.0
 ip nat inside
 speed auto
!
interface Serial0
 ip address 37.1.1.2 255.255.255.252
 ip nat outside
!
interface Serial1
 no ip address
 shutdown
!
interface Serial2
 no ip address
 shutdown
!
ip nat inside source list 10 interface Serial0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 37.1.1.1
no ip http server
ip pim bidir-enable
!
access-list 10 permit 192.168.1.0 0.0.0.255
!
line con 0
line aux 0
line vty 0 4
 password cisco
 login
end
```

A default installation of rat and was used to perform an initial audit of the router to see what possible security holes existed. Snarf was used to download the configuration.

```
# rat -a 192.168.1.10
snarfing 192.168.1.10...Hit Enter if no username is needed.
Username:
User Password:
Enable Password:
```

```
done.
auditing 192.168.1.10...done.
ncat_report: Guide file rscg.pdf not found in current directory.  Searching...
Linking to guide found at /usr/doc/rscg.pdf
ncat_report: writing 192.168.1.10.ncat_fix.txt.
ncat_report: writing 192.168.1.10.ncat_report.txt.
ncat_report: writing 192.168.1.10.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

The results from the test were less than impressive. The router failed many of the tests and scored a "2" overall.

```
Summary for 192.168.1.10
#Checks         #Passed         #Failed          %Passed
42               11               31               26
PerfectWeightedScore ActualWeighedScore %WeightedScore
281                  83                          29
Overall Score (0-10)
2
```

However, the fix file provided a wealth of information on changes to be made to the configuration. Typically, with a little cleanup, these changes can be applied to the router.

```
int Ethernet0
ip access-group 181 out
exit

no access-list 181
access-list 181 deny   ip any 10.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 127.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 172.16.0.0 0.15.255.255 log
access-list 181 deny   ip any 192.168.0.0 0.0.255.255 log
access-list 181 permit ip  192.168.1.0 0.0.0.255 any
access-list 181 deny   ip any any log
```

There is a problem though, on closer examination of the Egress Access Control List it looks like the script couldn't determine the difference between the internal and external interfaces. It's trying to apply an egress filter to the Ethernet interface which is connected to the inside network. Since the default installation of rat is used there appear to be some problems. Perhaps rat should have been localized first. At this point run ncat_config to localize rat and try again!

```
# ncat_config
ncat_config: Reading /usr/etc/ncat.conf.MASTER

Please answer the questions below about your network and
router configuration.  Type ? to get a short explanation of
any parameter.  If you are unsure about what value to give
for a parameter, hit RETURN to take the default value.

Select types of optional rules to be applied:

ncat_config: Apply rules for class use_multiple_ntp_servers [no] ?
ncat_config: Apply rules for class exterior_router [no] ? yes
ncat_config: Apply rules for class tacacs_aaa [no] ?
ncat_config: Apply rules for class localtime [no] ?
ncat_config: Apply rules for class gmt [no] ?
ncat_config: Apply rules for class snmp [no] ?
ncat_config: Apply rules for class exterior_router_with_2nd_if [no] ?

Change default configuration values:

ncat_config: Enter value for local_acl_num_egress [181] ?
```

```
ncat_config: Enter value for local_acl_num_ingress [180] ?
ncat_config: Enter value for local_acl_num_vty [182] ?
ncat_config: Enter value for local_address_internal_netblock_with_mask[192.168.1.0 0.0.0.255] ?
ncat_config: Enter value for local_address_loopback [192.168.1.3] ?
ncat_config: Enter value for local_address_ntp_host [1.2.3.4] ?
ncat_config: Skipping local_address_ntp_host_2 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Skipping local_address_ntp_host_3 because none of the prerequisite classes
(use_multiple_ntp_servers) were selected.
ncat_config: Enter value for local_address_syslog_host [192.168.1.3] ?
ncat_config: Enter value for local_address_telnet_acl_block_with_mask [192.168.1.0 0.0.0.7] ?
ncat_config: Enter value for local_address_telnet_acl_host [192.168.1.254] ?
ncat_config: Enter value for local_exec_timeout [5 0] ?
ncat_config: Enter value for local_external_interface [Ethernet0] ? Serial0
ncat_config: Skipping local_external_interface_2 because none of the prerequisite classes
(exterior_router_with_2nd_if) were selected.
ncat_config: Enter value for local_gmt_offset [0] ?
ncat_config: Enter value for local_loopback_num [0] ?
ncat_config: Enter value for local_timezone [GMT] ?

ncat_config: Writing /usr/etc/ncat.conf...Done.

ncat_config: Now examine /usr/etc/ncat.conf.
ncat_config: Edit /usr/etc/ncat.conf.MASTER and rerun ncat_config if not satisfactory.
```

Now that rat has been "Localized", rat can be run again. There is no need to use snarf to download the configuration again. The configuration of the router hasn't been changed yet and the text files rat created are still available. To run rat against the text dump from the previous attempt simply omit the "-a" flag and provide the filename of the configuration.

```
# rat 192.168.1.10
auditing 192.168.1.10...done.
ncat_report: writing 192.168.1.10.ncat_fix.txt.
ncat_report: writing 192.168.1.10.ncat_report.txt.
ncat_report: writing 192.168.1.10.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

Now take another look at the fix file and see if localization corrected some of the issues.

```
int serial0
ip access-group 180 in
exit

no cdp run
no access-list 181
access-list 181 deny   ip any 10.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 127.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 172.16.0.0 0.15.255.255 log
access-list 181 deny   ip any 192.168.0.0 0.0.255.255 log
access-list 181 permit ip  192.168.1.0 0.0.0.255 any
access-list 181 deny   ip any any log
```

The localization seems to have done the trick. After editing the "fix" file and consolidating some repetitive commands the configuration changes that follow were made to the router. Also corrected were a couple of minor issues with the access list created for ingress due to the internal use of private address space and NAT. It looks like the ncat_config script that Localizes rat was provided with incorrect information! The next time the script is run it should be provided with the NAT global address instead of the actual internal network address. That would have corrected the Access Control List problems. For now these changes are made manually and the corrected

configurations applied to the router.

```
service password-encryption
clock timezone GMT 0
enable secret cisco

line con 0
password cisco
exec-timeout 5 0
login
exit

line aux 0
password cisco
exec-timeout 5 0
login
transport input none
no exec
exit

line vty 0 4
access-class 182 in
exec-timeout 5 0
transport input telnet
exit

no access-list 182
access-list 182 permit tcp 192.168.1.0 0.0.0.15 any eq telnet
access-list 182 permit tcp host 192.168.1.254 any eq telnet
access-list 182 deny   ip any any log

int serial0
ip access-group 181 out
ip access-group 180 in
no ip proxy-arp
exit

no cdp run
no access-list 181
access-list 181 deny   ip any 10.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 127.0.0.0 0.255.255.255 log
access-list 181 deny   ip any 172.16.0.0 0.15.255.255 log
access-list 181 deny   ip any 192.168.0.0 0.0.255.255 log
access-list 181 permit ip  37.1.1.1 0.0.0.3 any
access-list 181 deny   ip any any log

no access-list 180
access-list 180 deny   ip 10.0.0.0 0.255.255.255 any log
access-list 180 deny   ip 127.0.0.0 0.255.255.255 any log
access-list 180 deny   ip 172.16.0.0 0.15.255.255 any log
access-list 180 deny   ip 192.168.0.0 0.0.255.255 any log
access-list 180 deny   ip any 10.0.0.0 0.255.255.255 log
access-list 180 deny   ip any 127.0.0.0 0.255.255.255 log
access-list 180 deny   ip any 172.16.0.0 0.15.255.255 log
access-list 180 deny   ip any 192.168.0.0 0.0.255.255 log
access-list 180 permit ip any any

int Serial1
no ip proxy-arp
exit
```

```
int Serial2
no ip proxy-arp
exit

ntp server 1.2.3.4
ntp source Loopback0
no ip source-route

int FastEthernet0
no ip proxy-arp
exit

logging 192.168.1.3
logging buffered 16000
logging console critical
logging trap debugging
```

After applying the configuration changes rat is run again. The router scores a much more impressive "8".

```
Summary for 192.168.1.10
#Checks          #Passed          #Failed          %Passed
35                31                4                88
PerfectWeightedScore    ActualWeighedScore     %WeightedScore
232                     203                    87
Overall Score (0-10)
8
```

While a score of "8" is not bad there is room for improvement. In order to reduce the size of the report use the "-f" flag to show only the failed rules.

```
# more 192.168.1.10.ncat_report.txt
Importance Pass/Fail  Rule                      Device         Line#       Instance
10         FAIL       ios - define telnet acl   192.168.1.10   1           n/a
7          FAIL       ios - egress filter definition   192.168.1.10   1           n/a
7          FAIL       ios - ingress filter definition  192.168.1.10   1           n/a
5          FAIL       ios - ntp source          192.168.1.10   1           n/a
….remaining output omitted….
```

Now it is time to correct the remaining problems. But wait, the telnet ACL and egress/ingress filtering were part of the changes that were applied previously. Why does the router still fail on these rules? Perhaps it is time to take a closer look at the rule base in the ncat.conf file. In specific the rule "Define telnet ACL" should be examined.

```
RuleName:IOS - Define telnet ACL
ruleclass:default,access
rulecontext:Global
ruledescription:Define telnet ACL.\
Telnet ACLs control what addresses may attempt to log in to your router.\
Change the numbers to match your ACL or use \d+.\
See rscg.pdf#Page55 for more information.
rulefix:\
no access-list 182\
access-list 182 permit tcp 192.168.1.0 0.0.0.7 any eq telnet\
access-list 182 permit tcp host 192.168.1.254 any eq telnet\
access-list 182 deny   ip any any log
ruleimportance:10
ruleinstance:.*
rulematch:access-list 182 permit tcp 192.168.1.0 0.0.0.7 any eq telnet\
access-list 182 permit tcp host 192.168.1.254 any eq telnet\
access-list 182 deny   ip any any log
ruletype:Required
ruleversion:version 1[12]\.*[ix]
```

Well it appears a mistake was made. Before applying the changes to the router the access-lists

were manually edited. The subnet-mask of the permitted hosts was modified to allow additional systems access to the router via telnet. Since rat must match a text string exactly the configuration fails the rule. This highlights the fact that care must be used when running the ncat_config localization. It also highlights the fact that changes should not be made before applying configurations. Localization should be done using ncat_config or by editing the ncat.conf.MASTERS file before running ncat_config.

```
access-list 182 permit tcp 192.168.1.0 0.0.0.15 any eq telnet
access-list 182 permit tcp host 192.168.1.254 any eq telnet
access-list 182 deny   ip any any log
```

At this point ncat_config is run again and the previous input errors are corrected. Net rat is run again and the results should be much better.

```
Summary for 192.168.1.10
#Checks         #Passed         #Failed          %Passed
35               35               0               100
PerfectWeightedScore ActualWeighedScore %WeightedScore
232                  232                100
Overall Score (0-10)
10
```

Wow a perfect 10! But, is the router really secure? According to Level 1 yes. For a basic site installation this is pretty impressive. The router is protected from the know security bugs and provides some limited protection against smurf attacks and, since it is an external router, has ingress and egress filters to prevent IP Spoofing.

But in a more advanced configuration there would be many more issues. This router is not using any routing protocols, such as EIGRP or BGP. Advanced filtering of traffic to protect hosts behind the router, including our web and mail servers, has not been defined. There is no secure method of remotely configuring the router, such as ssh. Most importantly this router is not in compliance with the corporate policy on router configuration. These are all issues that will become prevalent in larger environments.

With some customized configuration the Router Audit Tool can be used to audit each of these issues. There are several rules that are in the ncat.conf file that are not checked under the Level 1 baseline. In order to further improve security, additional classes of rules should be checked. We can also create custom rules to perform specific tasks.

Custom rules can be defined with relative ease. A complete description of how to create a rule can be found in the LOCALIZE.txt file. When creating new rules it is important to place them into the ncat.config.MASTER file located in usr/etc. If rules are placed in the ncat.conf file they will be overwritten the next time ncat_config is executed.

Examples are shown below for several new rules created to check for the presence of a notification banner and to disable ICMP redirect, unreachable and mask replies on external interfaces.

```
RuleName:IOS - No Notification Banner
RuleClass:default
RuleVersion:version 1[012]\.*
RuleContext:Global
RuleType:Required
RuleMatch:banner motd
RuleImportance:10
RuleDescription:Implement a notification banner\
A notification banner is required in order to prosecute\
unauthorized access to a computer system\
See rscg.pdf#Page49 for more information.
RuleFix:\
```

```
banner motd j\
Use is restricted to authorized users\
Usage is monitored.  Unauthorized use will be prosecuted\
Go Away you are not welcome here\
j

RuleName:IOS - no ICMP Redirect
RuleClass:exterior_router,routing
RuleVersion:version 1[012]\.*
RuleContext:IOSInterface
RuleInstance:LOCAL_EXTERNAL_INTERFACE
RULETYPE:Required
RuleMatch:no ip redirect
RuleImportance:7
RuleDescription:Disable ICMP Redirect Messages to external interfaces\
See rscg.pdf#Page66 for more information.
RuleFix:\
int LOCAL_EXTERNAL_INTERFACE\
no ip redirect\
exit

RuleName:IOS - no ICMP Unreachable
RuleClass:exterior_router,routing
RuleVersion:version 1[012]\.*
RuleContext:IOSInterface
RuleInstance:LOCAL_EXTERNAL_INTERFACE
RULETYPE:Required
RuleMatch:no ip unreachable
RuleImportance:7
RuleDescription:Disable ICMP Unreachable Messages to external interfaces\
See rscg.pdf#Page66 for more information.
RuleFix:\
int LOCAL_EXTERNAL_INTERFACE\
no ip unreachable\
exit

RuleName:IOS - no ICMP Mask Reply
RuleClass:exterior_router,routing
RuleVersion:version 1[012]\.*
RuleContext:IOSInterface
RuleInstance:LOCAL_EXTERNAL_INTERFACE
RULETYPE:Required
RuleMatch:no ip mask-reply
RuleImportance:7
RuleDescription:Disable ICMP Mask Reply Messages to external interfaces\
See rscg.pdf#Page66 for more information.
RuleFix:\
int LOCAL_EXTERNAL_INTERFACE\
no ip mask-reply\
exit
```

Many sites are also concerned about configuring routers using unsecured transport protocols such as telnet. If possible it is preferred to use ssh. Below the telnet rule is modify to require ssh to be configured in addition to telnet.

```
RuleName:IOS - vty transport telnet ssh
ruleclass:default,access
rulecontext:IOSLine
ruledescription:Permit only telnet and ssh transport.\
Only permit protocols you intend to use.  This prevents the other\
protocols from being misused.  Note that newer versions of IOS support\
SSH.  SSH should be used in place of telnet wherever possible.\
```

```
See rscg.pdf#Page55 for more information.
rulefix:\
line INSTANCE\
transport input ssh\
exit
ruleimportance:5
ruleinstance:vty
rulematch: transport input telnet ssh
ruletype:Required
ruleversion:version 1[012]\.*
```

This change will also require the telnet ACL rule to be modified adding a statement allowing ssh!

Conclusion. Using the Router Audit Tool can assist organizations in auditing their Cisco routers for security vulnerabilities. The tool can greatly decrease the time required to create secure configurations. The default rules use industry best practices based on the experience of hundreds of organizations and countless research. The tool can also be customized to an organizations specification to address issues specific to their infrastructure. Given that the Router Audit Tool is licensed under the GNU license it provides an excellent resource for organizations and the Internet community as a whole.

This document provided the background to substantiate the need for a tool like rat. The procedure for installing, customizing and using rat was discussed. The narrative guide described the steps to increase the security on a new router. New rules were created to increase the security of the configuration and insure the router was in compliance with the site standard.

What are the next steps to consider? What about creating a script to automatically run rat and post the output to a web server? This would allow an organization to check for unintentional configuration changes with security implications. With only a little work it should be possible to create a web based front end for rat to allow easier wide scale deployment for instant configuration checks. Imagine a corporate tool that network engineers could paste a configuration into and instantly confirm compliance with the corporate standard. Or allow an engineer to instantly check every router in an organization for a new vulnerability with only a few mouse clicks.

References:

---

[i] National Security Agency, “Router Security Configuration Guide”, November 21, 2001. URL: http://nsa2.www.conxion.com/cisco/

[ii] Potter, Valerie and Rosoff, Matt, “The Decade in Computing, Part 4: Success Stories”, C-net Tech Trends, September 27, 1999. URL:http://www.cnet.com/techtrends/0-6014-7-1463126.html

[iii] The Center for Internet Security Home Page. URL: http://www.cisecurity.org/

[iv] Jones, George, “INSTALL.txt” Documentation file for rat v1.1, 2002

[v] Jones, George, “INSTALL.txt” Documentation file for rat v1.1, 2002

[vi] Jones, George, “LOCALIZE.txt” Documentation file for rat v1.1, 2002

[vii] Jones, George, “ncat.conf.MASTER” Rule base file for rat v1.1, 2002

[viii] Jones, George, “rat man pages” Documentation for rat v1.1, 2002

[ix] Jones, George, “ncat.conf” Rule base file for rat v1.1, 2002

Additional Resources:

Improving Security on Cisco Routers, URL: http://www.cisco.com/warp/public/707/21.html

Akin, Thomas, “Hardening Cisco Routers”, O’reilly & Associates, 2002

Acohido, Byron, “Agency raises the bar on tech security Non-profit works to plug holes -- for free”, USA Today, February 27, 2002. URL: http://www.usatoday.com/usatonline/20020227/3896905s.htm

Hatta, Mohammed Shafri, “Securing IP Routing and Remote Access on Cisco Routers”, SANS Reading Room, September 20, 2001. URL: http://rr.sans.org/netdevices/telnet.php

Mordijck, Toon, “Disabling Unneeded Features and Services on Cisco Internet Gateway Routers”, SANS Reading Room, August 13, 2001. URL: http://rr.sans.org/netdevices/disabling.php

Jones, George, “Router Audit Tool and Benchmark”, SANS Webcast, February 20, 2002. URL:http://www.sans.org/webcasts/cisco_feb02.php

Ramsey, Steven, “Using Regular Expressions”, Electronic Text Center, University of Virginia, URL: http://etext.lib.virginia.edu/helpsheets/regex.html

Last Updated: July 24th, 2017

Upcoming Training

SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	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 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
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
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 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 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
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
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor