Global Information Assurance Certification Paper

Setting Up a Honeypot Using a Bait and Switch Router

Practical Assignment
GIAC Security Essentials Certification (GSEC)

Submitted By: Lorie W. Carter

July 29, 2004

Track 1 – SANS Security Essentials and the CISSP 10 Domains
Orlando, FL, April 2004

Table of Contents:

Abstract

While conducting research for this practical I found that there were many different arenas that warrant a closer look. I chose honeypots for this practical because they allow an administrator to track and learn from black-hats first hand without the attacker ever being aware that somebody is watching.

A honeypot can be as simple or elaborate as necessary. The Bait and Switch router method was used for this practical because it is cost effective, and it provided an opportunity to learn the Linux operating system.

This practical covers how to set up a Bait and Switch router on a Red Hat Linux server that will direct all unauthorized traffic to a honeypot while legitimate traffic will continue to the private network. I will then analyze some of the data collected during the exercise to demonstrate that the Bait and Switch method worked. A brief explanation of the exploits will be included in addition to the analysis of the data.

A Honeypot Defined

A honeypot is a device placed openly on a network for tracking unauthorized activity and to collect data related to that activity. The data collected is studied to learn about the latest tools and trends used by black-hats. The knowledge gained from this analysis is used as an aid for administrators in protecting system networks[1].

The two most common types of honeypots include High-Interaction and Low-Interaction. A High-Interaction honeypot makes the entire operating system along with the software installed accessible to a black-hat, where as a Low-Interaction honeypot only emulates systems and services running.

Both have advantages and disadvantages. High-Interaction honeypots do not limit the actions of the attacker, which allows for more data to be captured. Using a High-Interaction honeypot, however, can be expensive to deploy and comes at a high risk of being used to launch attacks. Low-Interaction honeypots are easy to deploy and can be considered inexpensive because they can emulate several machines running at the same time. The disadvantage is that the attacker is limited on what actions can be performed, thus, limiting the amount of data that can be captured for analysis.[2]

[1] The Honeynet Project, Know Your Enemy, Learning About Security Threats, Second Edition
[2] Spitzner, Lance, "Honeypots, Definitions and Value of Honeypots", URL: http://www.tracking-hackers.com/papers/honeypots.html

Overview

Red Hat Linux 9.0 was installed and was hardened using Bastille and Red Hat's Up2date on the Bait and Switch Router. A simple firewall was put into place using Firewall Builder as a front end for IP Tables on the same Linux machine.

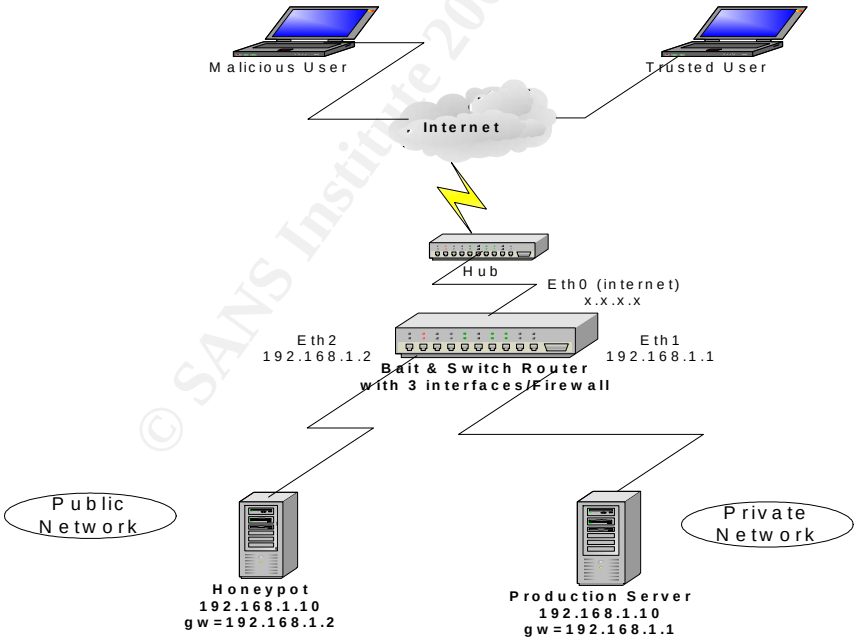
For the honeypot server, Windows 2000 Server was installed and nothing was locked down. This would allow black-hats free reign of everything on the system.

On the private network, data was also backed up on a regular basis to reinforce the integrity of the machines on the private network. Microsoft Baseline Security Analyzer (MBSA) was used to audit systems for vulnerabilities, then, all security holes were patched.

To track activity, the Snort plug-in for Bait and Switch was used to analyze the types of traffic received and was cross-referenced with the IIS and firewall logs.

The diagram below is an example of the architecture of the project.

DIAGRAM OF BAIT & SWITCH HONEYPOT



Getting Started

This project required a full install of the Linux 9.0 operating system. This helped to minimize some of the dependency problems that were encountered during an attempt to do a lean install of the operating system. Ximian desktop was also installed from www.ximian.com by clicking on the Ximian desktop link. Ximian is a GUI desktop interface for Linux.[3]

Up2date

The Linux Server was updated with the latest security patches using Red Hat's Up2date. Up2date is a client for a Red Hat maintained network that provides system updates by connecting to Red Hat servers to download and install the latest security patches. These packages are updated on a regular basis to address the latest security concerns. To obtain the latest version of Up2date, go to http://rhn.redhat.com/help/latest-up2date.pxt.[4]

Bastille

Bastille is recommended to harden any Linux server. Bastille is a free hardening tool that is helpful to new users in the Linux environment because of its ease of use. Bastille is a software tool that checks the system for security holes and then makes suggestions by providing a description of what the vulnerability is and why it should be closed. The Bastille download can be found out on http://www.bastille-linux.org/.[5]

Firewall Builder and IP Tables

The firewall was put in place primarily for masquerading purposes. Masquerading (IP Table's version of Network Address Translation) was needed because this project is set up in a home lab on a cable modem, and only one IP address is assigned. All of the machines needed to be hidden behind the public IP address of the cable modem. Also, NAT (Network Address Translation) was needed to allow public access through the Bait & Switch router to the privately addressed honeypot and production servers.

The firewall consists of IP Tables and Firewall Builder as the front end. IP Tables is a Linux based network packet filter that is installed by default when the operating system is installed. It filters packets as they enter the machine and then allows them to pass through or to be stopped according to what the rule sets call for. Firewall Builder uses an object-oriented approach that has a GUI

[3] Ximian, v. 2.0, URL: http://www.novell.com/products/desktop/
[4] Up2Date, URL: http://rhn.redhat.com/help/latest-up2date.pxt
[5] Bastille, v. 2.1, URL: http://www.bastille-linux.org (June 18, 2004)

and set of compilers used for configuring and managing a variety of platforms including IP Tables.

Downloading and Installing Firewall Builder

Since IP Tables is already installed by default when the operating system is installed, it is only required to install Firewall Builder. The packages required for the install can be obtained from the Firewall Builder web site by clicking on the download section (http://www.fwbuilder.org/).

First, install the following files in the order described. Start with the corresponding development packages for libsigc++10-1.0 and gtkmm-1.2.10.

```
rpm -ivh gtkmm-1.2.10-fr3.i386.rpm
rpm –ivh gtkmm-devel-1.2.10-fr.1386.rpm
rpm –ivh libsigc++10-1.0.4-fr3.1386.rpm
rpm –ivh libsigc++10-devel-1.0.4-fr3.1386.rpm
```

Next, run libfwbuilder-1.0.2.2.rh9.i386.rpm, fwbuilder-1.1.2-1.rh9.i386.rpm, fwbuilder-ipt.1.1.2-1.rh9.i386.rpm, and finally firewall_initscript.tar.gz with the following commands:

```
rpm  -ivh libfwbuilder-1.0.2-2.rh9.i386.rpm
rpm –ivh fwbuilder-1.1.2-1.rh9.i386.rpm
rpm – ivh fwbuilder-ipt.1.1.2-1.rh9.i386.rpm
tar –ivh firewall_initscript.tar.gz
```

Configuring Firewall Builder

To configure Firewall Builder, set the global configurations, the working path, create the firewall object and configure the interfaces for it, and finally add the host object for the honeypot. I also chose to show the object tree as a single view (in my opinion, this is an easier choice for viewing and compiling the rules),

Set up a directory for the service to get its rules from.

```
mkdir /etc/firewall.
```

Next, copy the firewall service into that service directory. The directory structure called for:

```
cp /opt/firewall-initscript/firewall /etc/init.d/firewall.
```

Then change permissions on the firewall directory by running the following command:

chmod 744 /etc/init.d/firewall.

Finally, include the 'symbolic link' for the service. Ideally, this should go after the network service in both run level 3 and run level 5 with the following commands:

ln –s /etc/init.d/firewall /etc/rc.d/rc3.d/S11firewall
ln –s /etc/init.d/firewall /etc/rc.d/rc5.d/S11firewall

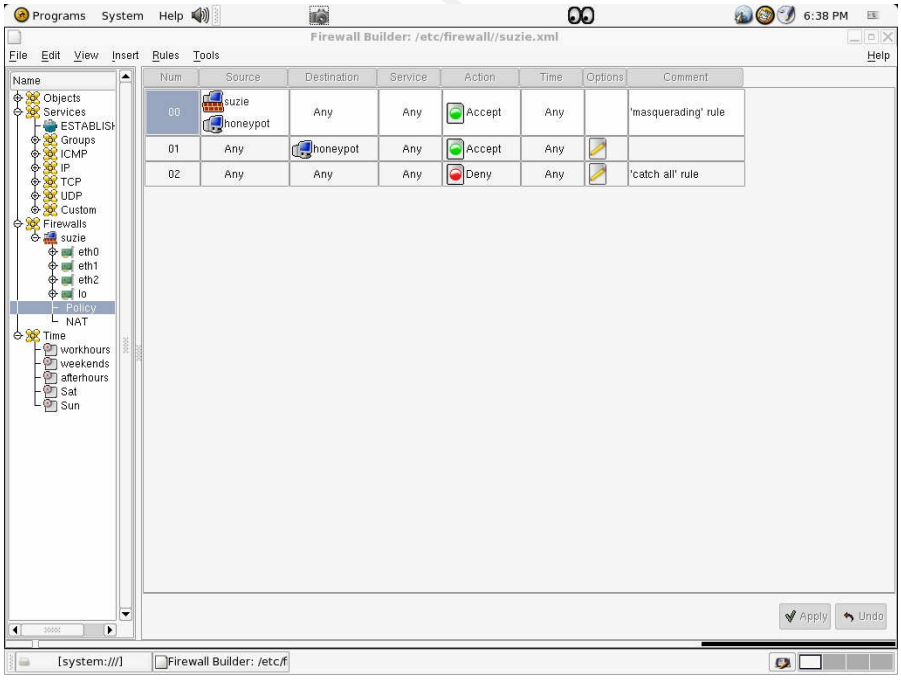
Setting up Rules and NAT

Three simple firewall rules applied:

From the source of the firewall and honeypot to any destination with any service, accept – (this is defining the masquerading rule).

From any source to the honeypot on any service, accept – (which means let any traffic enter the honeypot).

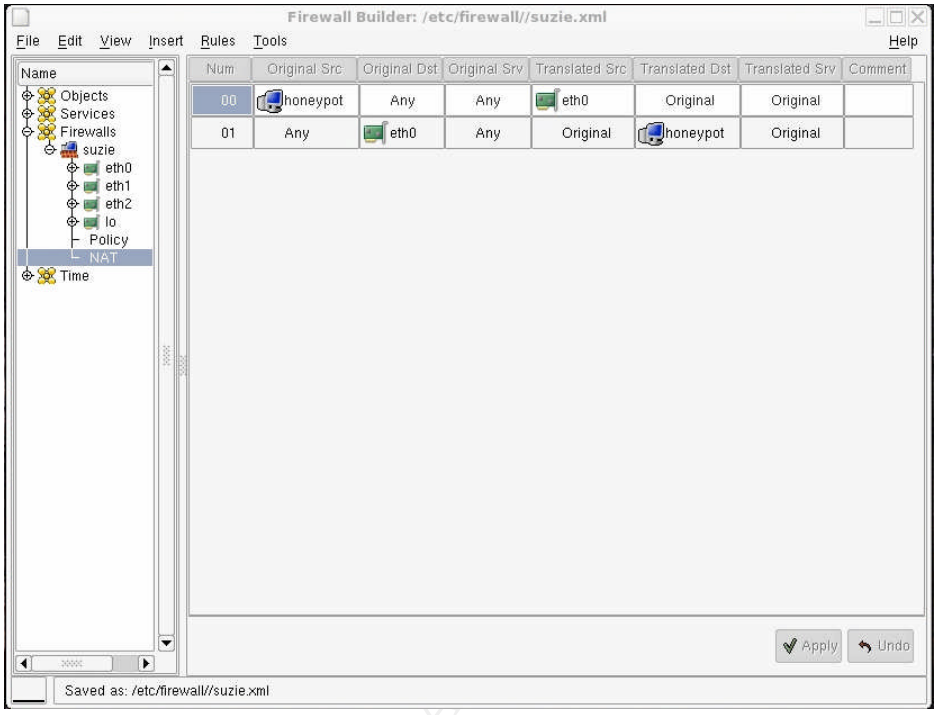
From any source to any destination on any service, deny (which is the catch all rule).



Under Network Address Translation (NAT), the following rules were applied:

From the original source (which is the honeypot) to any destination with any service, use Ethernet 0.

From any source to Ethernet 0 on any service gets translated to the honeypot.



Finally, compile the rules by clicking on the rules drop down menu and selecting the compile option. Then return to the terminal window to restart the firewall service with the following command to apply to newly created policy to IP tables.[6]

service firewall restart

Installing Bait and Switch

Bait and Switch is a Linux based freeware tool. It is used in combination with Snort version 2.0.2 and iproute2[7]. The effort is to redirect all hostile traffic silently to the honeypot that will partially mimic the production environment, but will still secure the production side. Bait and Switch works by having two machines that have the same IP address which makes them appear as one server, but the Bait and Switch router tells traffic which server to travel to via the Snort rule set. For example, if you want a known exploit to be "switched" to the honeypot you would write a Snort rule for that exploit, otherwise if no Snort rule matches then the traffic is legitimate and will be allowed to travel to the production server. Also,

[6] Unknown author, "Firewall Builder's User's Guide", November 04, 2003, URL: http://www.fwbuilder.org/UsersGuide.pdf

[7] The latest version of Bait and Switch includes an updated version of snort, but the instructions have not been updated to include this information.

you can set a particular “untrusted host” to always be switched to the honeypot by adding an entry to a “blacklist file”.

On the Linux Bait and Switch machine, three network interface cards will be required. Two interfaces will be used for the honeypot and production gateways. The other interface will communicate with the honeypot and production servers, back and forth to and from the Internet.

After configuring Ethernet 1 and Ethernet 2 with private IP’s and configuring Ethernet 0 for Internet usage (I used DHCP), download Bait and Switch, version 2.0, from http://baitnswitch.sourceforge.net and Snort, version 2.0.2, from www.snort.org. I chose to extract the software to the /opt directory. Do not install Snort at this point. It will need to be installed and configured after the Bait and Switch piece is complete. The command to extract and create the bnsroot directory for Bait and Switch is as follows:

```
tar –zxvf baitnswitch-**.tar.gz
```

To configure Bait and Switch, reference the following document found on the Bait and Switch web site, “How to Configure and Use a Bait and Switch Router”, https://sourceforge.net/docman/?group_id=64718[8]. These instructions were found to be straightforward and easy to use.

Switch to the bns/config directory and run the script to configure Bait and Switch.

```
cd /bns/config
./bns_conf.bash script
```

Choose option one only once so that it echoes the names of the new routing tables to /etc/iproute2/rt_tables. Next, select option two; you will be prompted to answer several questions including configuring the external, production, and honeypot interfaces. It also answers several questions pertaining to incoming packets, how Snort shall interact, and provides the option to make a file to hold a list of known blacklist addresses, which are to be automatically rerouted to the honeypot machine (if any are known at this time). Next, select option three in order to patch Snort. Now go back to the terminal window, switch to the bnsroot directory and run ./bnsroutes.bash (but first, you may need to change permissions). Then go to bnsroot/bns/switching and compile switchcore by typing in the following commands.[9]

```
chmod 777 ./bnsroutes.bash
./bnsroutes.bash
gcc –lpthread switchore.c –o switchcore.
```

[8] Witsitt, Jack, “Configuring a Bait and Switch Honeypot Router”, 56:4E, Version 1-b, URL: https://sourceforge.net/docman/display_doc.php?docid=15441&group_id=64718,
[9] Witsitt, Jack, “The Bait and Switch Honeypot”, July 14, 2003, URL: http://baitnswitch.sourceforge.net/

Installing Snort to Run on Bait and Switch

Lets talk about how Snort interacts with the Bait and Switch mechanism. Basically, incoming traffic gets sent through the Snort rules set and if a rule is matched, then it is sent to the Bait and Switch router via the Bait and Switch output plug-in for Snort, and is in turn, 'switched' to the honeypot.

Earlier, Snort, version 2.0.2, was downloaded and now it needs to be installed. Make a directory for Snort in the /etc directory.

```
mkdir  /etc/snort
```

Make the directory where the logs will be stored:

```
mkdir  /var/log/snort.
```

Go to the directory where the Snort packages were download and extract the program:

```
tar –xvzf snort-2.0.2.tar.gz
```

Then go to the Snort directory and type the following command to configure Snort.

```
cd /etc/snort
./configure
```

Issue a 'make' command and then a 'make install' command. Next, include the Snort rules. For ease of use, the standard rule set that was downloaded with Snort for the Bait and Switch output plug-in was added, but the preprocessors will not be used. From the Snort directory, switch to the rules directory and run the following commands:

```
cp * /etc/snort.
cd ../etc/snort
cp snort.conf /etc/snort
```

Note: The snort.conf file is where the rules would be modified if needed.[10]

Bringing The Bait and Switch Router Up

To run Bait and Switch, first, load switchcore with the following command:

[10] Caswell, Brian, *et al*, "Snort User's Manual current, The Snort Project", 2003, http://www.snort.org/docs/snort_manual/

/opt/bns/switching/switchcore.

Next, start Snort with this command:

/usr/local/snort –c /etc/snort/snort.conf

The Honeypot

Earlier different types of honeypots were discussed. A High-Interactive honeypot was chosen for this project. Nothing special was done here; the basic Windows 2000 Server install was implemented on the honeypot machine and nothing was locked down. Also, an administrative password was not used for this piece. A basic IIS web site was placed on both the honeypot and production servers to draw additional attention. Also some interesting bogus data was copied to the server to make it look authentic.

Note: Do not put a sniffer or tracking tools directly on the honeypot. Hackers are known to retaliate when they discover that they are being watched.

Securing the Production Network

On the production network, the data was backed up on a regular basis. Microsoft Baseline Analyzer (MBSA) was also used to provide an audit of what vulnerabilities existed. MBSA points out vulnerabilities of both the Microsoft operating system and other Microsoft related software that may be already installed on the machine. MBSA can be installed by the following link. Click on download and then save.

http://www.microsoft.com/downloads/details.aspx?FamilyID=8b7a580d-0c91-45b7-91ba-fc47f7c3d6ad&DisplayLang=en

Double click on the downloaded file, MBASetup_50_590_043.exe. This will extract the program to install automatically under: c:\program files\Microsoft Baseline Security Analyzer. Navigate to this folder and launch the executable called mbsa.exe and follow the prompts of the program. Choose the 'pick a computer to scan' option, and go with the default options. This will check to see if vulnerabilities exist for the Windows operation system, weak passwords, IIS, and SQL. It will also notify on which security patches need to be implemented. After the scan completes, a list is provided with the results and how to correct each problem. Unfortunately, MBSA will not automatically repair the vulnerabilities, so you will have to correct them manually. It does, however, provide links that have more information on what was scanned, details of the

results, and provides information on how to correct the vulnerability, and where to find the latest patches. [11]

Monitoring Traffic

The best practice for tracking activity would be to cross-reference the logs. First, the firewall logs were checked to see what activity was captured coming into the honeypot, or trying to get from the honeypot out to the internet. These logs can be found under /var/log/messages. A script called logwatcher2.pl was downloaded from the download link off of the Firewall Builder site. Once it is downloaded, change the permissions.

chmod 777 ./logwatcher2.pl.

There are a couple of options I recommend to be changed using the vi editor. These would be not to resolve DNS or resolve the port number. It will be easier to correlate to the Snort logs later because they, themselves, do not resolve DNS or the port numbers. Next, run this command from the opt directory to import the log to a text document.[12]

./logwatcher2.pl > ./fwlog

The Snort plug-in for the Bait and Switch Router was also used to monitor the logs. These files can be found under the var/log/snort directory. There is, however, a problem with Snort when it is used as a plug-in for Bait and Switch where it is not able to dump these files into an alert file. Later releases of Bait and Switch hope to address this issue.[13] So for now, you must parse through the alert directories manually.

Since we know Bait and Switch redirects traffic through the use of Snort rules, the Snort logs were used to pull the Snort alerts and were cross-referenced them against the firewall logs and from the IIS logs found on the honeypot to prove that it successfully worked. In the next section, I would like to demonstrate this and provide some basic research for each of the examples.

[11] Microsoft Baseline Security Analyzer, v 1.2, January 19, 2004, http://www.microsoft.com/downloads/details.aspx?FamilyID=8b7a580d-0c91-45b7-91ba-fc47f7c3d6ad&DisplayLang=en
[12] logwatcher2.pl, http://www.fwbuilder.org/archives/cat_downloads.html
[13] Forums, http://baitnswitch.sourceforge.net/

Example 1

Snort Rule

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS view
source via translate header"; flow:to_server,established; content: "Translate|3a| F"; nocase;
reference:arachnids,305; reference:bugtraq,1578; classtype:web-application-activity; sid:1042 ;
rev:6;)
```

Details

This alert is generated by intelligence gathering techniques where a packet can be crafted to allow the script source code to be returned to the hacker[14].

This IP address was looked up on www.samspade.org and found that it originated from an ISP in Rome, Italy.

False Positives

It is possible that this alert could generate a false positive through access to an application like WebDAV, however, a web publisher is not being used so this is not a false positive in this case[15].

Correlating Data

Snort Alert:

```
[**] WEB-IIS view source via translate header [**]
07/15-01:07:54.152883 82.49.199.217:3758 -> 24.225.84.229:80
TCP TTL:108 TOS:0x0 ID:4892 IpLen:20 DgmLen:189 DF
***AP*** Seq: 0xB5C51FDC  Ack: 0x4894A32E  Win: 0x4470  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] WEB-IIS view source via translate header [**]
07/15-01:09:43.838807 82.49.199.217:4803 -> 24.225.84.229:80
TCP TTL:108 TOS:0x0 ID:17262 IpLen:20 DgmLen:189 DF
***AP*** Seq: 0xC19242C5  Ack: 0x82206F79  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Firewall Log:

```
82.49.199.217 -> 192.168.1.10
Source Port: 3758
  Dest Port: 80
```

[14] Caswell, Brian, *et al*, "WEB-IIS view source via translate header", July 21, 2004, URL: www.snort.org/snort-db/sid.html?id=1042

[15] Caswell, WEB-IIS view source

```
Jul 15 01:07:53 RULE 1 ACCEPT eth0 TCP 82.49.199.217:3758 192.168.1.10:80

--  --
82.49.199.217 -> 192.168.1.10
Source Port: 4803
  Dest Port: 80
Jul 15 01:09:43 RULE 1 ACCEPT eth0 TCP 82.49.199.217:4803 192.168.1.10:80
```

IIS Log

```
2004-07-15 05:17:06 82.49.199.217 - 192.168.1.10 80 OPTIONS / - 200 Microsoft-WebDAV-MiniRedir/5.1.2600
```

Example 2

Snort Rules

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS cmd.exe access"; flow:to_server,established; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002;  rev:5;)
```

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%c0%af../"; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:981; rev:6;)
```

Details

These rules were generated from the same IP address and time line.

WEB-IIS cmd.exe access:
In this scenario, an attacker may be trying to gain information about IIS prior to an attack or to gain administrative access in an attempt to deface the web site, get information about the users of this system, or to gain sensitive customer information[16].

WEB-IIS unicode directory traversal attempt:
This alert is generated when an attacker tries to access the web directory. If they are successful, this could also allow them to gain access to other directories where commands can be executed possibly to launch attacks[17].

This IP address was looked up on www.samspade.org and this address originated from a user of Comcast Cable Company in Cherry Hill, NJ.

[16] Caswell, Brian, *et al*, "WEB-IIS cmd.exe access", URL: http://www.snort.org/snort-db/sid.html?id=1002
[17] Caswell, Brian, *et al*, "WEB-IIS unicode directory traversal attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=981

False Positives

WEB-IIS cmd.exe access:
No known false positives

WEB-IIS Unicode directory traversal attempt:
Snort's web site describes that false positives can be generated by visiting a particular web site[18]. However, this is not the case, because the administrator did not access the particular web site, and there are no users involved with this project.

Correlating Data

Snort Alert:

```
[**] WEB-IIS cmd.exe access [**]
07/13-08:19:15.492905 24.15.214.185:4416 -> 24.225.84.229:80
TCP TTL:111 TOS:0x0 ID:53260 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0xEC44FF7D  Ack: 0xBE296562  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] WEB-IIS unicode directory traversal attempt [**]
07/13-08:19:16.390421 24.15.214.185:4429 -> 24.225.84.229:80
TCP TTL:111 TOS:0x0 ID:53299 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0xEC50D8AF  Ack: 0xF646786C  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] WEB-IIS unicode directory traversal attempt [**]
07/13-08:19:17.115483 24.15.214.185:4443 -> 24.225.84.229:80
TCP TTL:111 TOS:0x0 ID:53350 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xEC5DFC55  Ack: 0xF64B71C9  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] WEB-IIS unicode directory traversal attempt [**]
07/13-08:19:17.799460 24.15.214.185:4458 -> 24.225.84.229:80
TCP TTL:111 TOS:0x0 ID:53428 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xEC6BBBBE  Ack: 0xF6506D4A  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] WEB-IIS cmd.exe access [**]
07/13-08:19:18.449964 24.15.214.185:4475 -> 24.225.84.229:80
TCP TTL:111 TOS:0x0 ID:53487 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0xEC7A1412  Ack: 0xF6557D22  Win: 0x4470  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

[18] Caswell, WEB-IIS cmd.exe

Firewall Logs:

```
24.15.214.185 -> 192.168.1.10
Source Port: 4410
  Dest Port: 80
Jul 13 08:19:14 RULE 1 ACCEPT eth0 TCP 24.15.214.185:4410 192.168.1.10:80

-  --

24.15.214.185 -> 192.168.1.10
Source Port: 4412
  Dest Port: 80
Jul 13 08:19:15 RULE 1 ACCEPT eth0 TCP 24.15.214.185:4412 192.168.1.10:80

--  --
24.15.214.185 -> 192.168.1.10
Source Port: 4416
  Dest Port: 80
Jul 13 08:19:15 RULE 1 ACCEPT eth0 TCP 24.15.214.185:4416 192.168.1.10:80

--  --
24.15.214.185 -> 192.168.1.10
Source Port: 4418
  Dest Port: 80
Jul 13 08:19:15 RULE 1 ACCEPT eth0 TCP 24.15.214.185:4418 192.168.1.10:80

--  --
24.15.214.185 -> 192.168.1.10
Source Port: 4425
  Dest Port: 80
Jul 13 08:19:16 RULE 1 ACCEPT eth0 TCP 24.15.214.185:4425 192.168.1.10:80
```

IIS Log:

```
2004-07-13 15:29:25 24.15.214.185 - 192.168.1.10 80 GET /d/winnt/system32/cmd.exe /c+dir
404 -
2004-07-13 15:29:25 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..%5c../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:25 24.15.214.185 - 192.168.1.10 80 GET
/_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:25 24.15.214.185 - 192.168.1.10 80 GET
/_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:25 24.15.214.185 - 192.168.1.10 80 GET
/msadc/..%5c../..%5c../..%5c/..Á  ../..Á  ../..Á  ../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:26 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..Á  ../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:26 24.15.214.185 - 192.168.1.10 80 GET /scripts/winnt/system32/cmd.exe
/c+dir 404 -
```

```
2004-07-13 15:29:26 24.15.214.185 - 192.168.1.10 80 GET /scripts/../../winnt/system32/cmd.exe
/c+dir 404 -
2004-07-13 15:29:26 24.15.214.185 - 192.168.1.10 80 GET /scripts/..\../winnt/system32/cmd.exe
/c+dir 404 -
2004-07-13 15:29:26 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..%5c../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:27 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..%5c../winnt/system32/cmd.exe /c+dir 404 -
2004-07-13 15:29:27 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..%5c../winnt/system32/cmd.exe /c+dir 404 –
2004-07-13 15:29:27 24.15.214.185 - 192.168.1.10 80 GET
/scripts/..%2f../winnt/system32/cmd.exe /c+dir 404 -
```

Example 3

Snort Rule

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS Proxy attempt";
flags:S,12; reference:url,help.undernet.org/proxyscan/; classtype:attempted-recon; sid:615;
rev:4;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy attempt";
flags:S,12; classtype:attempted-recon; sid:618; rev:4;)
```

Details

These rules were generated from the same IP address and time frame.

SCAN SOCKS Proxy attempt:
This is an information-gathering attempt to see if the SOCKS proxy server is installed on this server. If port 1080 is found open, an attempt can be made to start communication locally and possibly to gain access to the network. Proxies are often used to start attacks to other hosts[19].

SCAN Squid Proxy attempt:
This is also an information-gathering attempt to see if Scan Squid Proxy is running on this server. Usually scans of this nature are attempted to see what is present on the host prior to an attack and to check which ports are being monitored by the firewall[20].

A lookup of this address on www.samspade.org indicated that the address originated from Comcast Cable Communications in Mt. Laurel, NJ.

False Positives

[19] Gomez, Gene R., "SCAN SOCKS Proxy attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=615
[20] Caswell, Brian, *et al*, SCAN Squid Proxy attempt, July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=618

SCAN SOCKS Proxy attempt:
There are some known false positives. Non-proxy applications can also use port 1080, which will cause the alert to trigger[21].

SCAN Squid Proxy attempt:
False positives can be generated when a scanner is being used during a vulnerability assessment in a security audit[22]. Not likely the case here!

Correlating Data

Snort Alert:

```
[**] SCAN SOCKS Proxy attempt [**]
07/14-02:54:26.869978 :22002 -> 24.225.84.229:1080
TCP TTL:113 TOS:0x0 ID:15214 IpLen:20 DgmLen:40
******S* Seq: 0x323F  Ack: 0x5792  Win: 0x7860  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] SCAN Squid Proxy attempt [**]
07/14-02:54:27.087029 68.62.127.9:22002 -> 24.225.84.229:3128
TCP TTL:118 TOS:0x0 ID:1649 IpLen:20 DgmLen:40
******S* Seq: 0x1BA9  Ack: 0x4F5E  Win: 0x2A9D  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Firewall Log:

```
68.62.127.9 -> 192.168.1.10
Source Port: 22002
  Dest Port: 1080
Jul 14 02:54:26 RULE 1 ACCEPT eth0 TCP 68.62.127.9:22002 192.168.1.10:1080

--  --
68.62.127.9 -> 192.168.1.10
Source Port: 22002
  Dest Port: 3128
Jul 14 02:54:27 RULE 1 ACCEPT eth0 TCP 68.62.127.9:22002 192.168.1.10:3128
```

[21] Gomez, SCAN SOCKS
[22] Caswell, Scan Squid

Example 4

Snort Rule

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL Worm propagation
attempt"; content:"|04|"; depth:1; content:"|81 F1 03 01 04 9B 81 F1 01|"; content:"sock";
content:"send"; reference:bugtraq,5310; classtype:misc-attack; reference:bugtraq,5311;
reference:url,vil.nai.com/vil/content/v_99992.htm; sid:2003; rev:2;)
```

Details

The Slammer worm is making an attempt to infect a SQL Server by sending a buffer overflow to the MS SQL Server 2000 Resolution Service[23].

www.samspade.org reported that this IP address originated from GT Telecom Group in Vancouver, BC.

False Positives

There are no known false positives.

Correlating Data

Snort Alert:

```
[**] MS-SQL Worm propagation attempt [**]
07/15-21:59:12.711969 216.183.13.189:4276 -> 24.225.84.229:1434
UDP TTL:110 TOS:0x0 ID:56960 IpLen:20 DgmLen:404
Len: 376
04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  ................
01 DC C9 B0 42 EB 0E 01 01 01 01 01 01 01 70 AE  ....B.........p.
42 01 70 AE 42 90 90 90 90 90 90 90 90 68 DC C9  B.p.B........h..
B0 42 B8 01 01 01 01 31 C9 B1 18 50 E2 FD 35 01  .B.....1...P..5.
01 01 05 50 89 E5 51 68 2E 64 6C 6C 68 65 6C 33  ...P..Qh.dllhel3
32 68 6B 65 72 6E 51 68 6F 75 6E 74 68 69 63 6B  2hkernQhounthick
43 68 47 65 74 54 66 B9 6C 6C 51 68 33 32 2E 64  ChGetTf.llQh32.d
68 77 73 32 5F 66 B9 65 74 51 68 73 6F 63 6B 66  hws2_f.etQhsockf
B9 74 6F 51 68 73 65 6E 64 BE 18 10 AE 42 8D 45  .toQhsend....B.E
D4 50 FF 16 50 8D 45 E0 50 8D 45 F0 50 FF 16 50  .P..P.E.P.E.P..P
BE 10 10 AE 42 8B 1E 8B 03 3D 55 8B EC 51 74 05  ....B....=U..Qt.
BE 1C 10 AE 42 FF 16 FF D0 31 C9 51 51 50 81 F1  ....B....1.QQP..
03 01 04 9B 81 F1 01 01 01 01 51 8D 45 CC 50 8B  ..........Q.E.P.
```

[23] Caswell, Brian, *et al*, "MS-SQL Worm propagation attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=2003

```
45 C0 50 FF 16 6A 11 6A 02 6A 02 FF D0 50 8D 45  E.P..j.j.j...P.E
C4 50 8B 45 C0 50 FF 16 89 C6 09 DB 81 F3 3C 61  .P.E.P.....<a
D9 FF 8B 45 B4 8D 0C 40 8D 14 88 C1 E2 04 01 C2  ...E...@.....
C1 E2 08 29 C2 8D 04 90 01 D8 89 45 B4 6A 10 8D  ...).....E.j..
45 B0 50 31 C9 51 66 81 F1 78 01 51 8D 45 03 50  E.P1.Qf..x.Q.E.P
8B 45 AC 50 FF D6 EB CA                          .E.P....
```

Firewall Log:

```
216.183.13.189 -> 192.168.1.10
Source Port: 4276
  Dest Port: 1434
Jul 15 21:59:12 RULE 1 ACCEPT eth0 UDP 216.183.13.189:4276 192.168.1.10:1434
```

Example 5

Snort Rule

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC
WebDAV search access"; flow:to_server,established; content: "SEARCH "; depth: 8;
nocase;reference:arachnids,474; classtype:web-application-activity; sid:1070; rev:6;)
```

Details

WebDAV is a web-publishing tool that is included with IIS. A rule is generated when a remote user does a search for WebDAV in an attempt to obtain a list of directories on the web server. It can also be a probe to gain information for a more serious attack[24].

IP address originated from Rogers Cable, Inc. in Toronto, ON.

False Positives

There are no known positives for this alert.

Correlating Data

Snort Alert:

```
[**] WEB-MISC WebDAV search access [**]
07/14-12:12:01.781855 24.43.55.176:4102 -> 24.225.84.229:80
TCP TTL:104 TOS:0x0 ID:45532 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xE300D391  Ack: 0x93D07917  Win: 0x4470  TcpLen: 20
53 45 41 52 43 48 20 2F 90 02 B1 02 B1 02 B1 02  SEARCH /.....
B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02  ................
B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02  ................
```

[24] Unknown author, "WEB-MISC WebDAV search access", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=1070

```
B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02  ................
B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02 B1 02  ................
B1 02 B1 02                                      ....
```

Firewall Log:

```
24.43.55.176 -> 192.168.1.10
Source Port: 4079
  Dest Port: 2745
Jul 14 12:12:00 RULE 1 ACCEPT eth0 TCP 24.43.55.176:4079 192.168.1.10:2745
```

Lessons Learned

Although Windows is known as a user-friendly operating system, it is not as flexible as Linux because its structure is not an open source format, therefore, there are not as many tools available to provide analysis of attacks.

There is, however, an abundance of freeware tools available for the Linux operating system, and scripts have been written to do just about anything. However, support, or lack there of, was very frustrating. In the quest to find out how to perform certain actions, there was little or no information available. In numerous forums, questions were posted from other new users, such as myself, that went unanswered. But if you can get over the newbie hurdle, there is a lot of gratification in what you set out to achieve.

Conclusion

A honeypot has proven to be an invaluable tool because an administrator can track and learn from black-hats first hand without the attacker ever being aware that they are being watched. As we discussed, there are many different types of honeypots, including High-Interaction and Low-Interaction honeypots. The Bait and Switch router method is just one of many possibilities of the High-Interactive type.

This practical covered how to set up a Bait and Switch router on a Red Hat Linux server that directed unauthorized traffic to the Windows 2000 honeypot server while legitimate traffic continued to flow to the private network. Some of the captured data was analyzed during the exercise to demonstrate that the Bait and Switch method worked. A brief explanation the exploits were included in addition to the analysis of the data.

References

Books

Barrett, Daniel J., Linux Pocket Guide, Sebastopol, CA, O'Reilly Media, Inc., February 2004

Honeynet Project, The, Know Your Enemy, Learning About Security Threats, Second Edition, Boston, MA, Addison Wesely, May 1, 2004

Taylor, Dave, James C. Armstrong, Jr, Teach Yourself Unix in 24 Hours, Indianapolis, IN, Sams Publishing, 1997

Web Sites

Caswell, Brian, *et al*, "MS-SQL Worm propagation attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=2003, (Last accessed July 21, 2004)

Caswell, Brian, *et al*, "SCAN Squid Proxy attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=618, (Last accessed July 21, 2004)

Caswell, Brian, *et al*, "Snort User's Manual current, The Snort Project", 2003, URL: http://www.snort.org/docs/snort_manual/, (Last accessed May 2, 2004)

Caswell, Brian, *et al*, "WEB-IIS cmd.exe access", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?id=1002, (Last accessed July 21, 2004)

Caswell, Brian, *et al*, "WEB-IIS unicode directory traversal attempt", July 21, 2004. URL: http://www.snort.org/snort-db/sid.html?sid=981, (Last accessed July 21, 2004)

Caswell, Brian, *et al*, "WEB-IIS view source via translate header", July 21, 2004, URL: www.snort.org/snort-db/sid.html?id=1042, (Last accessed July 21, 2004)

Gomez, Gene R., "SCAN SOCKS Proxy attempt", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=615, (Last accessed July 21, 2004)

Spitzner, Lance, "Honeypots, Definitions and Value of Honeypots", May 29, 2003, URL: http://www.tracking-hackers.com/papers/honeypots.html, (Last accessed April 25, 2004)

Unknown author, "Firewall Builder's User's Guide", November 04, 2003, URL: http://www.fwbuilder.org/UsersGuide.pdf (Last accessed, July 17, 2004)

Unknown author, "WEB-MISC WebDAV search access", July 21, 2004, URL: http://www.snort.org/snort-db/sid.html?sid=1070, (Last accessed July 21, 2004)

Witsitt, Jack, The Bait and Switch Honeypot, July 14, 2003, URL: http://baitnswitch.sourceforge.net/, (Last accessed May 25, 2004)

Witsitt, Jack, "Configuring a Bait and Switch Honeypot Router", 56:4E, Version 1-b, URL: https://sourceforge.net/docman/display_doc.php?docid=15441&group_id=64718, (May 28, 2004)

Tool Downloads

Bait and Switch, v. 2.0, September, 27, 2003, https://sourceforge.net/project/showfiles.php?group_id=64718

Bastille, v. 2.1, http://www.bastille-linux.org

Firewall Builder, v. 2.0, http://www.fwbuilder.org/

logwatcher2.pl, http://www.fwbuilder.org/archives/cat_downloads.html

Microsoft Baseline Security Analyzer, v 1.2, January 19, 2004, http://www.microsoft.com/downloads/details.aspx?FamilyID=8b7a580d-0c91-45b7-91ba-fc47f7c3d6ad&DisplayLang=en

Snort, v. 2.0.2, September17, 2003, http://www.snort.org/dl/old/

Up2Date, http://rhn.redhat.com/help/latest-up2date.pxt

Ximian, v. 2.0, http://www.novell.com/products/desktop/download.html

Last Updated: May 27th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event