Proxying Tools

An important aspect of using web proxies is enabling the interception of web requests made by command-line tools and thick client applications. This gives us transparency into the web requests made by these applications and allows us to utilize all of the different proxy features we have used with web applications.

To route all web requests made by a specific tool through our web proxy tools, we have to set them up as the tool's proxy (i.e. `http://127.0.0.1:8080`), similarly to what we did with our browsers. Each tool may have a different method for setting its proxy, so we may have to investigate how to do so for each one.

This section will cover a few examples of how to use web proxies to intercept web requests made by such tools. You may use either Burp or ZAP, as the setup process is the same.

> Note: Proxying tools usually slows them down, therefore, only proxy tools when you need to investigate their requests, and not for normal usage.

Proxychains

One very useful tool in Linux is proxychains, which routes all traffic coming from any command-line tool to any proxy we specify. `Proxychains` adds a proxy to any command-line tool and is hence the simplest and easiest method to route web traffic of command-line tools through our web proxies.

To use `proxychains`, we first have to edit `/etc/proxychains.conf`, comment out the final line and add the following line at the end of it:

Proxying Tools

```
#socks4         127.0.0.1 9050
http 127.0.0.1 8080
```

We should also enable `Quiet Mode` to reduce noise by un-commenting `quiet_mode`. Once that's done, we can prepend `proxychains` to any command, and the traffic of that command should be routed through `proxychains` (i.e., our web proxy). For example, let's try using `cURL` on one of our previous exercises:

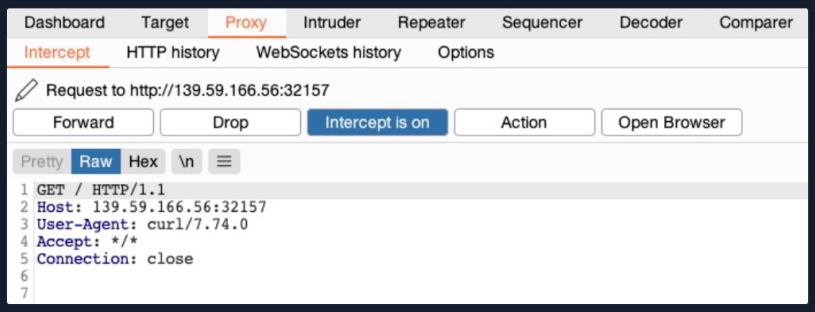
Proxying Tools

```
MisaelMacias@htb[/htb]$ proxychains curl http://SERVER_IP:PORT

ProxyChains-3.1 (http://proxychains.sf.net)
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Ping IP</title>
    <link rel="stylesheet" href="./style.css">
</head>
...SNIP...
</html>
```

We see that it worked just as it normally would, with the additional `ProxyChains-3.1` line at the beginning, to note that it is being routed through `ProxyChains`. If we go back to our web proxy (Burp in this case), we will see that the request has indeed gone through it:



Nmap

Next, let's try to proxy `nmap` through our web proxy. To find out how to use the proxy configurations for any tool, we can view its manual with `man nmap`, or its help page with `nmap -h`:

Proxying Tools

```
MisaelMacias@htb[/htb]$ nmap -h | grep -i prox

  --proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
```

As we can see, we can use the `--proxies` flag. We should also add the `-Pn` flag to skip host discovery (as recommended on the man page). Finally, we'll also use the `-sC` flag to examine what an nmap script scan does:

Proxying Tools

```
MisaelMacias@htb[/htb]$ nmap --proxies http://127.0.0.1:8080 SERVER_IP -pPORT -Pn -sC
```

```
Starting Nmap 7.91 ( https://nmap.org )
Nmap scan report for SERVER_IP
Host is up (0.11s latency).

PORT      STATE SERVICE
PORT/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
```

Once again, if we go to our web proxy tool, we will see all of the requests made by nmap in the proxy history:

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Logger | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
111	http://139.59.166.56	POST	/IPHTTPS						
110	http://139.59.166.56	PROPFI...	/						
109	http://139.59.166.56	PROPFI...	/						
108	http://139.59.166.56	OPTIONS	/						
107	http://139.59.166.56	GET	/robots.txt					text	txt
106	http://139.59.166.56	GET	/nmaplowercheck1628146178						
105	http://139.59.166.56	OPTIONS	/						

Note: Nmap's built-in proxy is still in its experimental phase, as mentioned by its manual (`man nmap`), so not all functions or traffic may be routed through the proxy. In these cases, we can simply resort to `proxychains`, as we did earlier.

Metasploit

Finally, let's try to proxy web traffic made by Metasploit modules to better investigate and debug them. We should begin by starting Metasploit with `msfconsole`. Then, to set a proxy for any exploit within Metasploit, we can use the `set PROXIES` flag. Let's try the `robots_txt` scanner as an example and run it against one of our previous exercises:

Proxying Tools

```
MisaelMacias@htb[/htb]$ msfconsole

msf6 > use auxiliary/scanner/http/robots_txt
msf6 auxiliary(scanner/http/robots_txt) > set PROXIES HTTP:127.0.0.1:8080

PROXIES => HTTP:127.0.0.1:8080


msf6 auxiliary(scanner/http/robots_txt) > set RHOST SERVER_IP

RHOST => SERVER_IP


msf6 auxiliary(scanner/http/robots_txt) > set RPORT PORT

RPORT => PORT


msf6 auxiliary(scanner/http/robots_txt) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Once again, we can go back to our web proxy tool of choice and examine the proxy history to view all sent requests:

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Logger | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	
112	http://139.59.166.56:32157	GET	/robots.txt			404	393	HTML	txt	Error

Request

Pretty | Raw | Hex | \n

```
1 GET /robots.txt HTTP/1.0
2 Host: 139.59.166.56:32157
3 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
4 Connection: close
5
6
```

Response

Pretty | Raw | Hex | Render | \n

```
1 HTTP/1.1 404 Not Found
2 X-Powered-By: Express
3 Content-Security-Policy: default-src 'none'
4 X-Content-Type-Options: nosniff
5 Content-Type: text/html; charset=utf-8
6 Content-Length: 149
```

We see that the request has indeed gone through our web proxy. The same method can be used with other scanners, exploits, and other features in Metasploit.

We can similarly use our web proxies with other tools and applications, including scripts and thick clients. All we have to do is set the proxy of each tool to use our web proxy. This allows us to examine exactly what these tools are sending and receiving and potentially repeat and modify their requests while performing web application penetration testing.

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

+ 1 Try running 'auxiliary/scanner/http/http_put' in Metasploit on any website, while routing the traffic through Burp. Once you view the requests sent, what is the last line in the request?

msf test file

Submit Hint

Previous Next

Mark Complete & Next