



Transfer payload to Windows From Kali.

```
(root@kali)-[/var/lib/veil/output/compiled]
# ls
danger.exe
# scp danger.exe 'Ajay Pratap Singh'@192.168.1.45
# scp danger.exe 'Ajay Pratap Singh'@192.168.1.45:C:/Users/Public/
```

Fig 1.2 Show payload being sent

On Windows , execute the payload manually.



Public Account Pictures	13-08-2025 16:41	File folder	
Public Desktop	16-09-2025 22:57	File folder	
Public Documents	28-06-2025 00:23	File folder	
Public Downloads	07-12-2019 14:44	File folder	
Public Music	07-12-2019 14:44	File folder	
Public Pictures	07-12-2019 14:44	File folder	
Public Videos	07-12-2019 14:44	File folder	
danger.exe	16-09-2025 21:08	Application	73 KB
other_host_marker.txt	12-09-2025 14:45	Text Document	1 KB
systeminfo_before.txt	12-09-2025 12:39	Text Document	10 KB
test.txt	13-08-2025 17:02	Text Document	1 KB
test_time.txt	12-09-2025 12:39	Text Document	1 KB
wazuh_test_download.txt	12-09-2025 15:07	Text Document	0 KB
wazuh_test_marker.txt	12-09-2025 15:03	Text Document	1 KB
wazuh_time_after.txt	12-09-2025 15:07	Text Document	1 KB
wazuh_time_before.txt	12-09-2025 15:03	Text Document	1 KB

Fig 1.3 Shows payload being received by windows

Network Evasion (proxy chains + tor)

Install Tor by these commands.

```
(ajay@kali)-[~]
$ sudo apt install tor -y
tor is already the newest version (0.4.8.16-1).
tor set to manually installed.
The following packages were automatically installed and are no longer required:
  libdbus-glib-1-2 python3-gpg
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0

(ajay@kali)-[~]
$ sudo systemctl start tor

(ajay@kali)-[~]
$ sudo systemctl enable tor
Synchronizing state of tor.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable tor

(ajay@kali)-[~]
$ sudo systemctl status tor
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
   Loaded: loaded (/usr/lib/systemd/system/tor.service; enabled; preset: disabled)
   Active: active (exited) since Wed 2025-09-17 21:28:46 IST; 12s ago
  Invocation: 43de526ef15d4e1d92e57a4a1724b048
    Main PID: 38713 (code=exited, status=0/SUCCESS)
     Mem peak: 1.8M
        CPU: 8ms

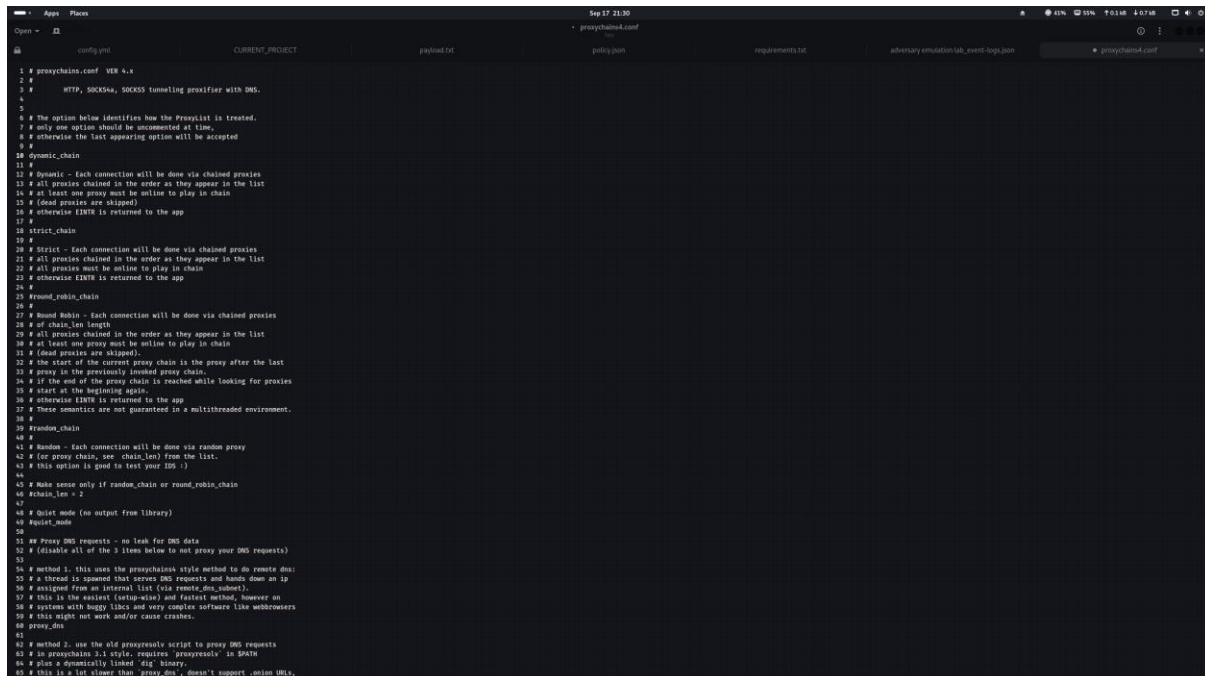
Sep 17 21:28:46 kali systemd[1]: Starting tor.service - Anonymizing overlay network for TCP (multi-instance-master)...
Sep 17 21:28:46 kali systemd[1]: Finished tor.service - Anonymizing overlay network for TCP (multi-instance-master).

(ajay@kali)-[~]
$
```

Fig 1.4 Tor running



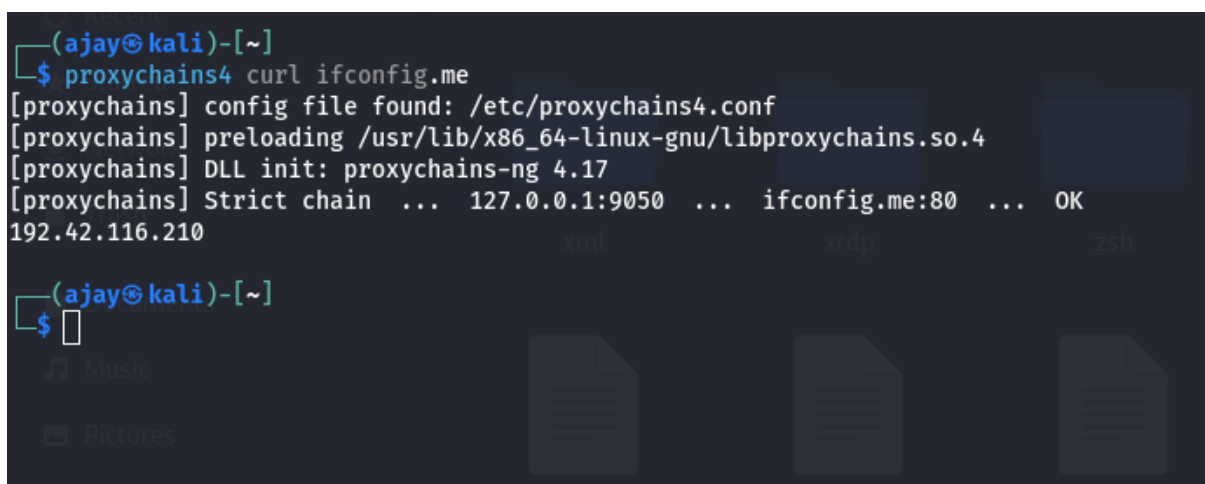
Now Configure ProxyChains.



```
1 # proxychains.conf  VER 4.x
2
3 # HTTP, SOCKS4a, SOCKS5 tunneling proxyifier with DNS.
4
5 # The option below identifies how the Proxylist is treated.
6 # only one option should be uncommented at time.
7 # otherwise the last appearing option will be accepted
8
9 # Dynamic chain
10 #
11 # Dynamic - Each connection will be done via chained proxies
12 # All proxies chained in the order as they appear in the list
13 # at least one proxy must be online to play in chain
14 # (dead proxies are skipped)
15 # otherwise FINTR is returned to the app
16 #
17 # Strict chain
18 #
19 # Strict - Each connection will be done via chained proxies
20 # All proxies chained in the order as they appear in the list
21 # at least one proxy must be online to play in chain
22 # (dead proxies are skipped)
23 # otherwise FINTR is returned to the app
24 #
25 # Round robin chain
26 #
27 # Round Robin - Each connection will be done via chained proxies
28 # of chain length
29 # All proxies chained in the order as they appear in the list
30 # at least one proxy must be online to play in chain
31 # (dead proxies are skipped)
32 # the start of the current proxy chain is the proxy after the last
33 # proxy in the previously loaded proxy chain.
34 # if the end of the proxy chain is reached while looking for proxies
35 # start at the beginning again.
36 # otherwise FINTR is returned to the app
37 # These semantics are not guaranteed in a multithreaded environment.
38 #
39 # Random chain
40 #
41 # Random - Each connection will be done via random proxy
42 # (or proxy chain, see chain_len) from the list.
43 # This option is good to test your IDS :)
44 #
45 # Make sense only if random_chain or round_robin_chain
46 # chain_len = 2
47 #
48 # Quiet mode (no output from library)
49 #
50 #
51 # Proxy DNS requests - no leak for DNS data
52 # (disable all of the 3 items below to not proxy your DNS requests)
53 #
54 # method 1: this uses the proxychains4 style method to do remote dns:
55 # a thread is spawned that serves DNS requests and hands down an ip
56 # assigned from an internal list (via remote_dns_method1).
57 # this is the easiest (setup-wise) and fastest method, however on
58 # systems with buggy libc and very complex software like webrowsers
59 # this might not work and/or cause crashes.
60 # proxy_dns
61 #
62 # method 2: use the old proxychains script to proxy DNS requests
63 # in proxychains 4.x style requires 'proxychains' in PATH
64 # plus a dynamically linked 'dig' binary.
65 # this is a lot slower than proxy_dns, doesn't support some URIs,
66 # and it's not recommended to use with complex software like webrowsers.
```

Fig 1.5 ProxyChain configuration

Launch Metasploit



```
(ajay@kali)-[~]
$ proxychains4 curl ifconfig.me
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:9050 ... ifconfig.me:80 ... OK
192.42.116.210
(ajay@kali)-[~]
$
```

Fig 1.6 Check for proxychains



Log Table

Type	AV Detection
Meterpreter	Bypassed