LINUX FIREWALL EXPLORATION LAB

Contents

TASK 1

- Ensuring we have firewall iptables installed and active which it is.

```
[10/30/22]seed@VM:~$ sudo service ufw status
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; ena
   Active: active (exited) since Sun 2022-10-30 05:52:0
  Process: 240 ExecStart=/lib/ufw/ufw-init start quiet
 Main PID: 240 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Warning: Journal has been rotated since unit was starte
...skipping...
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; ena
   Active: active (exited) since Sun 2022-10-30 05:52:0
  Process: 240 ExecStart=/lib/ufw/ufw-init start quiet
 Main PID: 240 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Warning: Journal has been rotated since unit was starte
```

- Now flushing the iptables policy and to ensure policy table is empty we will list the policy and checking subnet configurations in Machine A.

```
[10/30/22]seed@VM:~$ sudo iptables -F
[10/30/22]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[10/30/22]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:6b:36:a8
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::69d7:5d87:c7b7:5df9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9028 (9.0 KB)  TX bytes:14691 (14.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:99 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23586 (23.5 KB)  TX bytes:23586 (23.5 KB)
```

- Checking ip configurations of Machine B

```
[10/30/22]seed@VM:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:8a:d7:2c:d0  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.6  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::43c4:2cc1:3839:af09  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:4f:f5:7f  txqueuelen 1000  (Ethernet)
        RX packets 216  bytes 239053 (239.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 133  bytes 13694 (13.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Prevent A from doing telnet to Machine B

- Now using the command to drop the machine IP based on subnet configurations.

```
[10/30/22]seed@VM:~$ sudo iptables -A OUTPUT -p tcp --dport 23 -d 10.0.2.14 -j DROP
[10/30/22]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             10.0.2.14            tcp dpt:telnet
```

- Finally testing if we are successful in the task. Which it seems we are successful as the Machine A can't telnet to Machine B because the connection gets timed out after trying for quite some time.

```
[10/30/22]seed@VM:~$ telnet 10.0.2.14
Trying 10.0.2.14...
telnet: Unable to connect to remote host: Connection timed out
```

Prevent B from doing telnet to Machine A

Implementing the firewall by rejecting input instead of output to prevent B machine from telnetting to A.

```
[10/30/22]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 23 -s 10.0.2.14 -j REJECT
[10/30/22]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  10.0.2.14            anywhere             tcp dpt:telnet reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             10.0.2.14            tcp dpt:telnet
```

Prevent A from visiting an External Website

- Now rejecting output to a website in order for machine A to not reach the external web which is New York University's web page in this case.

```
[10/30/22]seed@VM:~$ sudo iptables -A OUTPUT -d www.nyu.edu -j REJECT
[10/30/22]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  10.0.2.14            anywhere             tcp dpt:telnet reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             10.0.2.14            tcp dpt:telnet
REJECT     all  --  anywhere             server-13-35-169-127.fjr50.r.cloudfront.net  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             server-13-35-169-122.fjr50.r.cloudfront.net  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             server-13-35-169-85.fjr50.r.cloudfront.net  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             server-13-35-169-35.fjr50.r.cloudfront.net  reject-with icmp-port-unreachable
```

TASK 2

- Going to Directory /Desktop/SEED. Then flushing and checking if the policy list is cleared.

```
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -F
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- Here we are using code for building a simple firewall while providing ip configurations to iptables.

simplefirewall.c

```c
#include "simplefirewall.h"
/*
• Prevent A from doing telnet to Machine B.
• Prevent B from doing telnet to Machine A.
• Prevent A from visiting an external web site NYIT.edu (64.35.176.173).
*/


void init_policies(void) {
        //for rule 1, Prevent A from doing telnet to Machine B.
        simplefirewall_polices[0].policyType = OUTPUT;
        simplefirewall_polices[0].protocolType = TCP;
        //ANY
        simplefirewall_polices[0].srcIp[0] = 0;
        simplefirewall_polices[0].srcIp[1] = 0;
        simplefirewall_polices[0].srcIp[2] = 0;
        simplefirewall_polices[0].srcIp[3] = 0;
        simplefirewall_polices[0].srcPort = -1; //ANY
        //destination address of Machine B
        simplefirewall_polices[0].destIp[0] = 172;
        simplefirewall_polices[0].destIp[1] = 16;
        simplefirewall_polices[0].destIp[2] = 0;
        simplefirewall_polices[0].destIp[3] = 4;
        simplefirewall_polices[0].destPort = 23;

        //for rules 2, Prevent B from doing telnet to Machine A.
        simplefirewall_polices[1].policyType = INPUT;
        simplefirewall_polices[1].protocolType = TCP;
        //ANY
        simplefirewall_polices[1].srcIp[0] = 172;
        simplefirewall_polices[1].srcIp[1] = 16;
        simplefirewall_polices[1].srcIp[2] = 0;
        simplefirewall_polices[1].srcIp[3] = 4;
        simplefirewall_polices[1].srcPort = -1; //ANY
        //destination address of Machine A
        simplefirewall_polices[1].destIp[0] = 172;
        simplefirewall_polices[1].destIp[1] = 16;
        simplefirewall_polices[1].destIp[2] = 0;
        simplefirewall_polices[1].destIp[3] = 5;
        simplefirewall_polices[1].destPort = 23;

        //for rule 3, Prevent A from visiting an external web site NYIT.edu (64.35.176.173) via HTTP.
        simplefirewall_polices[2].policyType = OUTPUT;
        simplefirewall_polices[2].protocolType = TCP;
        //ANY
        simplefirewall_polices[2].srcIp[0] = 0;
        simplefirewall_polices[2].srcIp[1] = 0;
        simplefirewall_polices[2].srcIp[2] = 0;
        simplefirewall_polices[2].srcIp[3] = 0;
        simplefirewall_polices[2].srcPort = -1; //ANY
        //destination address of NYIT.edu (64.35.176.173).
        simplefirewall_polices[2].destIp[0] = 64;
```

simplefirewall.h

```
#define pr_fmt(fmt) "%s:%s: " fmt, KBUILD_MODNAME, __func__
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/tcp.h>

#define MAX_FW_POLICY   5

typedef enum policyType {INPUT, OUTPUT} policyType;
typedef enum protocolType {TCP, UDP, OTHER} protocolType;

const unsigned char any[4] = {0,0,0,0};

typedef struct simplefirewall_policy{
  policyType policyType;
  protocolType protocolType;

  unsigned char srcIp[4];
  int srcPort;
  unsigned char destIp[4] ;
  int destPort;


} simplefirewall_policy;

//static policy support 5 rules
static simplefirewall_policy simplefirewall_polices[5];

//NF_IP_PRE_ROUTING for inbound
static struct nf_hook_ops simpleFirewall_netfilter_ops_in;
//NF_IP_POST_ROUTING for outbound
static struct nf_hook_ops simpleFirewall_netfilter_ops_out;
```

MakeFile

```
obj-m += simplefirewall.o
all:
        make --debug=j -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
        make --debug=j -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

- Compiling the firewall application code files with set rules/policies.

```
[10/30/22]seed@VM:~/.../SEED$ ls
Makefile  simplefirewall.c  simplefirewall.h
[10/30/22]seed@VM:~/.../SEED$ make all
make --debug=j -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Desktop/SEED modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
Putting child 0x9b74640 (crmodverdir) PID 3212 on the chain.
Live child 0x9b74640 (crmodverdir) PID 3212
Reaping winning child 0x9b74640 PID 3212
Removing child 0x9b74640 PID 3212 from chain.
Putting child 0x9b73a70 (Module.symvers) PID 3223 on the chain.
Live child 0x9b73a70 (Module.symvers) PID 3223
Reaping winning child 0x9b73a70 PID 3223
Removing child 0x9b73a70 PID 3223 from chain.
Putting child 0x9b744a8 (_module_/home/seed/Desktop/SEED) PID 3231 on the chain.
Live child 0x9b744a8 (_module_/home/seed/Desktop/SEED) PID 3231
Putting child 0x8f451b8 (/home/seed/Desktop/SEED/simplefirewall.o) PID 3232 on the chain.
Live child 0x8f451b8 (/home/seed/Desktop/SEED/simplefirewall.o) PID 3232
  CC [M]  /home/seed/Desktop/SEED/simplefirewall.o
```

- Checking the presence of simplefirewall.ko file and then removing it. Moreover, initializing creating simplefirewall.

```
[10/30/22]seed@VM:~/.../SEED$ ls
Makefile          simplefirewall.ko
modules.order     simplefirewall.mod.c
Module.symvers    simplefirewall.mod.o
simplefirewall.c  simplefirewall.o
simplefirewall.h
[10/30/22]seed@VM:~/.../SEED$ sudo insmod simplefirewall.ko
[10/30/22]seed@VM:~/.../SEED$ sudo lsmod | grep simple
simplefirewall         16384  0
[10/30/22]seed@VM:~/.../SEED$ sudo rmmod simplefirewall.ko
[10/30/22]seed@VM:~/.../SEED$ dmesg -wH
[Oct30 05:51] Linux version 4.8.0-36-generic (buildd@lgw01-13) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.4) ) #36~16.04.1-Ubun
tu SMP Sun Feb 5 09:39:41 UTC 2017 (Ubuntu 4.8.0-36.36~16.04.1-generic 4.8.11)
[  +0.000000] KERNEL supported cpus:
[  +0.000000]   Intel GenuineIntel
[  +0.000000]   AMD AuthenticAMD
[  +0.000000]   NSC Geode by NSC
[  +0.000000]   Cyrix CyrixInstead
[  +0.000000]   Centaur CentaurHauls
[  +0.000000]   Transmeta GenuineTMx86
[  +0.000000]   Transmeta TransmetaCPU
[  +0.000000]   UMC UMC UMC UMC
[  +0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[  +0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[  +0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[  +0.000000] x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
[  +0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[  +0.000000] x86/fpu: Using 'eager' FPU context switches.
[  +0.000000] e820: BIOS-provided physical RAM map:
[  +0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[  +0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[  +0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[  +0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000007ffefffff] usable
[  +0.000000] BIOS-e820: [mem 0x000000007fff0000-0x000000007fffffff] ACPI data
[  +0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
```

- As proof that firewall has been setup we can notice the configurations modified and added in the iptables

```
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ufw-before-logging-input  all  --  anywhere             anywhere
ufw-before-input  all  --  anywhere             anywhere
ufw-after-input  all  --  anywhere             anywhere
ufw-after-logging-input  all  --  anywhere             anywhere
ufw-reject-input  all  --  anywhere             anywhere
ufw-track-input  all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
ufw-before-logging-forward  all  --  anywhere             anywhere
ufw-before-forward  all  --  anywhere             anywhere
ufw-after-forward  all  --  anywhere             anywhere
ufw-after-logging-forward  all  --  anywhere             anywhere
ufw-reject-forward  all  --  anywhere             anywhere
ufw-track-forward  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ufw-before-logging-output  all  --  anywhere             anywhere
ufw-before-output  all  --  anywhere             anywhere
ufw-after-output  all  --  anywhere             anywhere
ufw-after-logging-output  all  --  anywhere             anywhere
ufw-reject-output  all  --  anywhere             anywhere
ufw-track-output  all  --  anywhere             anywhere

Chain ufw-after-forward (1 references)
target     prot opt source               destination

Chain ufw-after-input (1 references)
target     prot opt source               destination
ufw-skip-to-policy-input  udp  --  anywhere             anywhere             udp dpt:netbios-ns
ufw-skip-to-policy-input  udp  --  anywhere             anywhere             udp dpt:netbios-dgm
ufw-skip-to-policy-input  tcp  --  anywhere             anywhere             tcp dpt:netbios-ssn
```

TASK 3

- Ensuring the ufw service is enabled.

```
[10/30/22]seed@VM:~/.../SEED$ sudo service ufw status
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor
   Active: active (exited) since Sun 2022-10-30 05:52:06 EDT; 40min
  Process: 240 ExecStart=/lib/ufw/ufw-init start quiet (code=exited
 Main PID: 240 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Oct 30 06:29:23 VM systemd[1]: Started Uncomplicated firewall.
Warning: Journal has been rotated since unit was started. Log outpu
lines 1-9/9 (END)
```

- Flushing and checking if the iptables have been flushed of previous configurations.

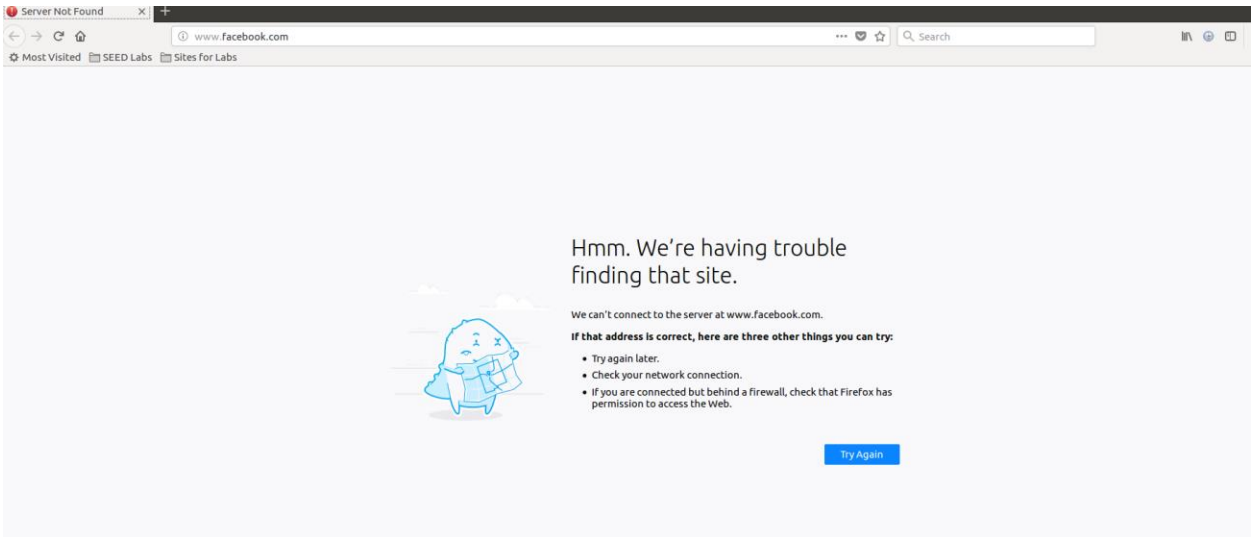
```
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -F
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain ufw-after-forward (0 references)
target     prot opt source               destination

Chain ufw-after-input (0 references)
target     prot opt source               destination

Chain ufw-after-logging-forward (0 references)
target     prot opt source               destination

Chain ufw-after-logging-input (0 references)
target     prot opt source               destination

Chain ufw-after-logging-output (0 references)
target     prot opt source               destination

Chain ufw-after-output (0 references)
```

- Now we have blocked all the outgoing connections to telnet port. It can be seen as the connection is not made instead timed out and iptables have the configurations as well.

```
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -A OUTPUT -p tcp --dport 23 -j REJECT
[10/30/22]seed@VM:~/.../SEED$ telnet 10.0.2.14
Trying 10.0.2.14...
telnet: Unable to connect to remote host: Connection timed out
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp dpt:telnet reject-with icmp-port-unreachable
```

- Now verifying of traffic blocked going to Facebook

```
[10/30/22]seed@VM:~/.../SEED$ wget -t 1 -T 10 http://www.facebook.com
--2022-10-30 06:48:41--  http://www.facebook.com/
Resolving www.facebook.com (www.facebook.com)... failed: Connection timed out.
wget: unable to resolve host address 'www.facebook.com'
[10/30/22]seed@VM:~/.../SEED$ wget -t 1 -T 10 https://www.facebook.com
--2022-10-30 06:49:24--  https://www.facebook.com/
Resolving www.facebook.com (www.facebook.com)... failed: Connection timed out.
wget: unable to resolve host address 'www.facebook.com'
```



Task 3.a

- Using the given command to connect to Machine B using command ,"ssh –L 8000:10.0.2.6:23 seed@10.0.2.6"

```
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI
.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '10.0.2.6' (ECDSA) to the list of known hosts.
seed@10.0.2.6's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

- After closing the connection I noticed we can't reconnect.

```
[10/30/22]seed@VM:~/.../SEED$ telnet localhost 8000
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection timed out
```

- Telnet on Port 8000

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Harshita-MachineB login: seed
Password:
Last login: Thu Oct 14 15:33:00 EDT 2021 from 10.0.2.5 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

- Now verifying connection by checking ip configurations

```
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:5d:b8:de
          inet addr:10.0.2.6  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::3017:cbd6:681b:f4b1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:214 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19803 (19.8 KB)  TX bytes:27306 (27.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:379 errors:0 dropped:0 overruns:0 frame:0
          TX packets:379 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:55952 (55.9 KB)  TX bytes:55952 (55.9 KB)
```

Task 3.b

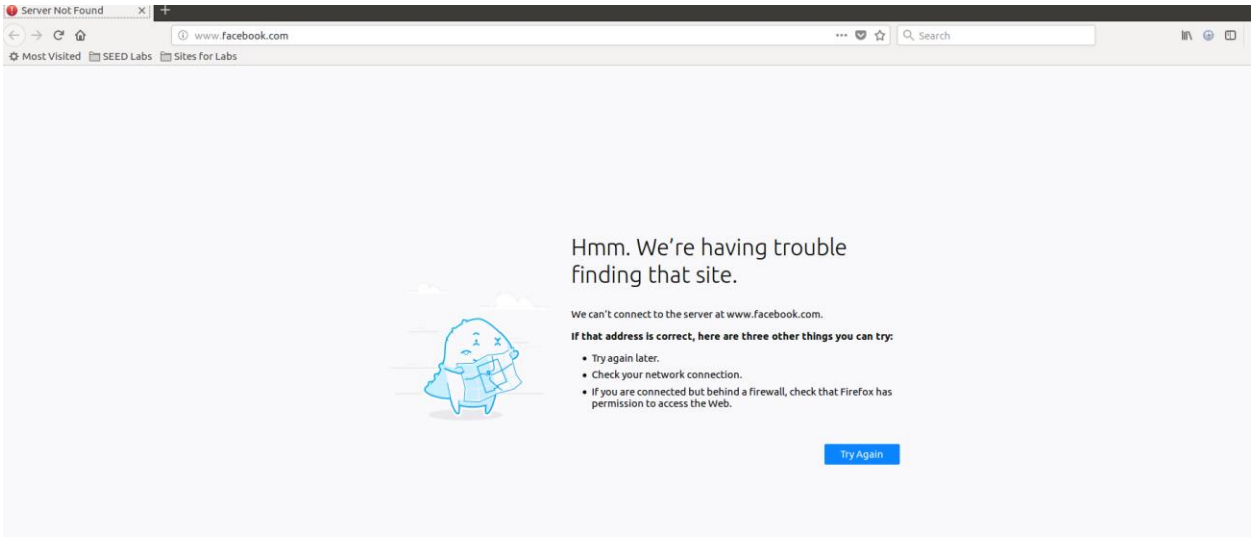
- Now trying to connect to Facebook using SSH tunneling using command,"sudo host facebook.com"

```
facebook.com has address 157.240.3.35
facebook.com has IPv6 address 2a03:2880:f101:83:face:b00c:0:25de
facebook.com mail is handled by 10 smtpin.vvv.facebook.com.
```

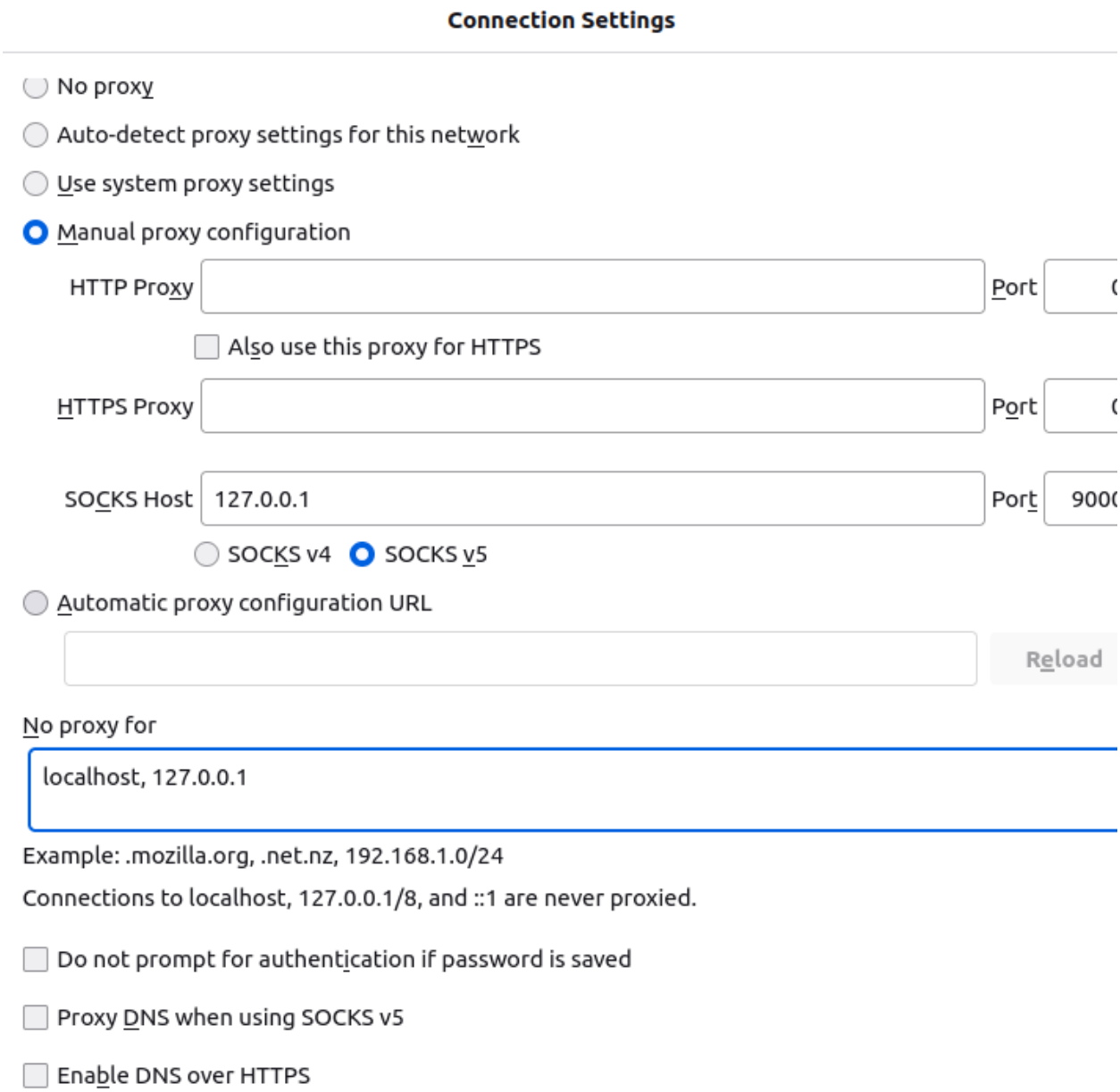
- Now connecting with Machine B for the above purpose with the command,"ssh –D 9000 –C seed@10.0.2.6

```
seed@10.0.2.6's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

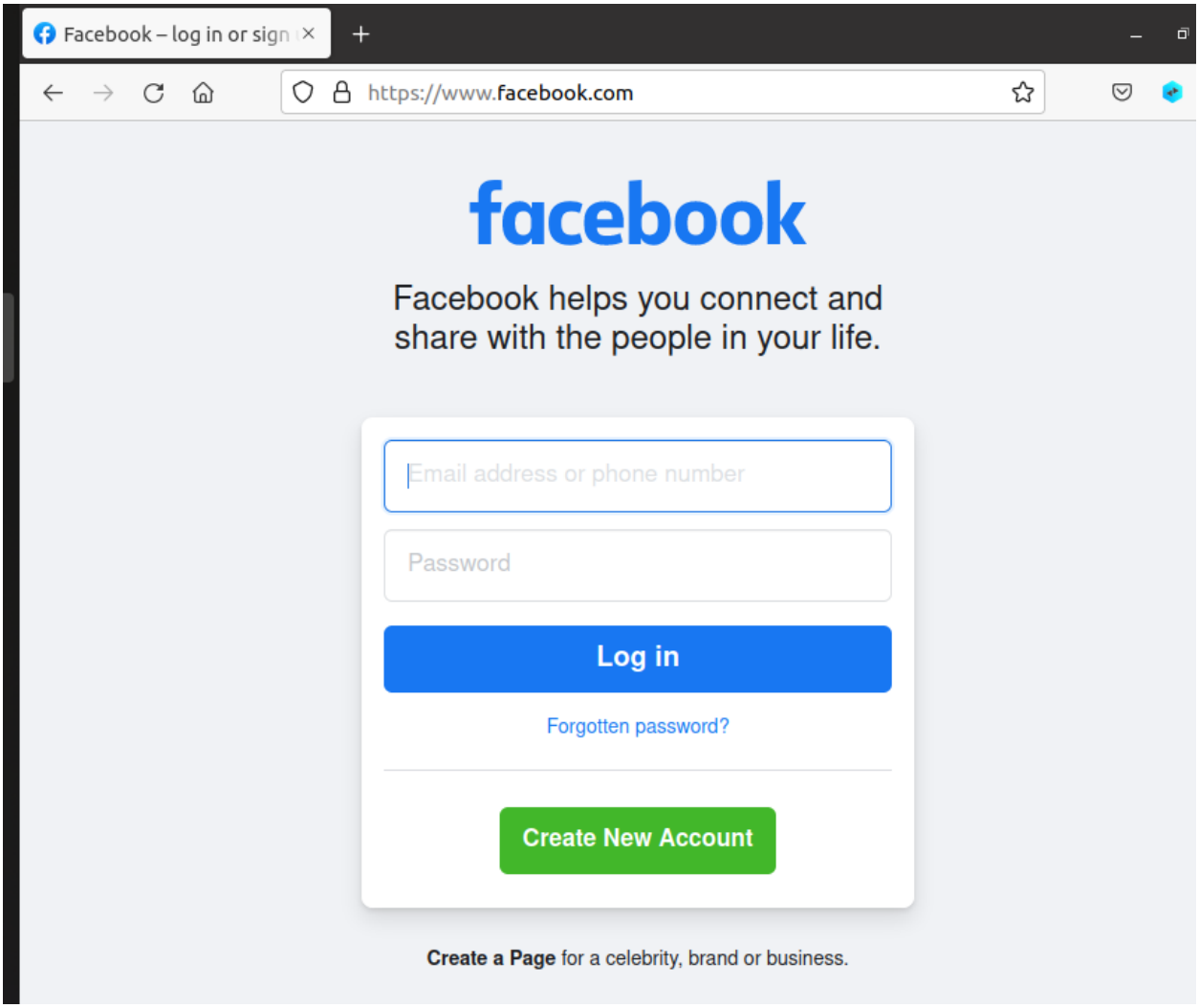
- Now I can't connect to Facebook



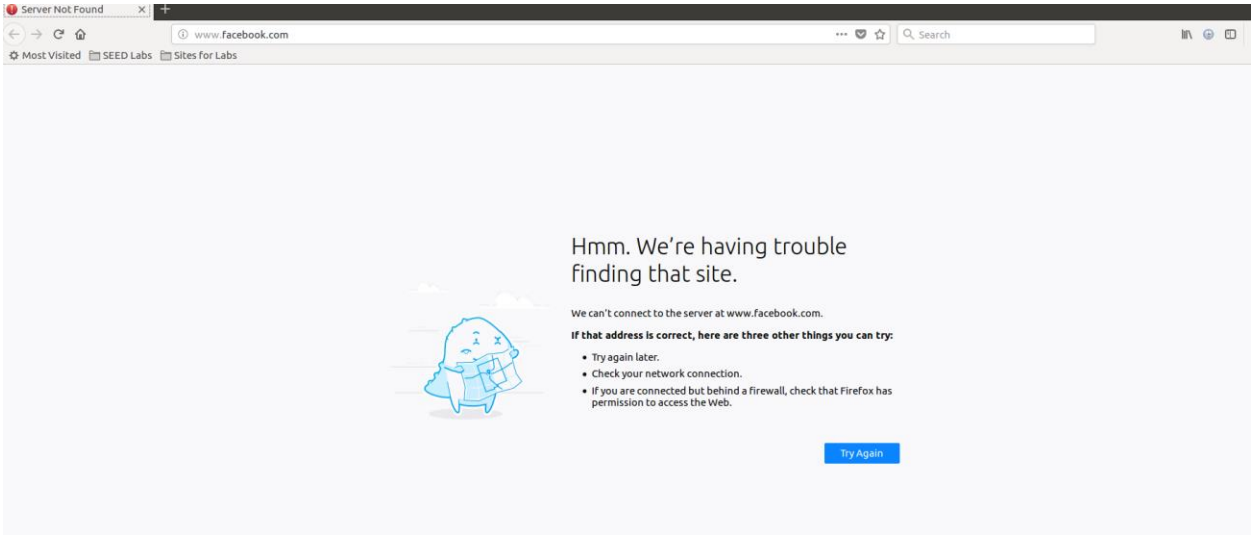
- Making some proxy changes in order to establish the connection



- Now Facebook is accessible.



- After closing connection and retrying to connect I couldn't succeed.



TASK 4

- Ensuring the ufw service is enabled.

```
[10/30/22]seed@VM:~/.../SEED$ service ufw status
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor
   Active: active (exited) since Sun 2022-10-30 05:52:06 EDT; 3h 17
  Process: 240 ExecStart=/lib/ufw/ufw-init start quiet (code=exited
 Main PID: 240 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Oct 30 06:29:23 VM systemd[1]: Started Uncomplicated firewall.
Warning: Journal has been rotated since unit was started. Log outpu
```

- Flushing and checking if the iptables have been flushed of previous configurations.

```
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -F
[10/30/22]seed@VM:~/.../SEED$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain ufw-after-forward (0 references)
target     prot opt source               destination

Chain ufw-after-input (0 references)
target     prot opt source               destination

Chain ufw-after-logging-forward (0 references)
target     prot opt source               destination

Chain ufw-after-logging-input (0 references)
target     prot opt source               destination
```

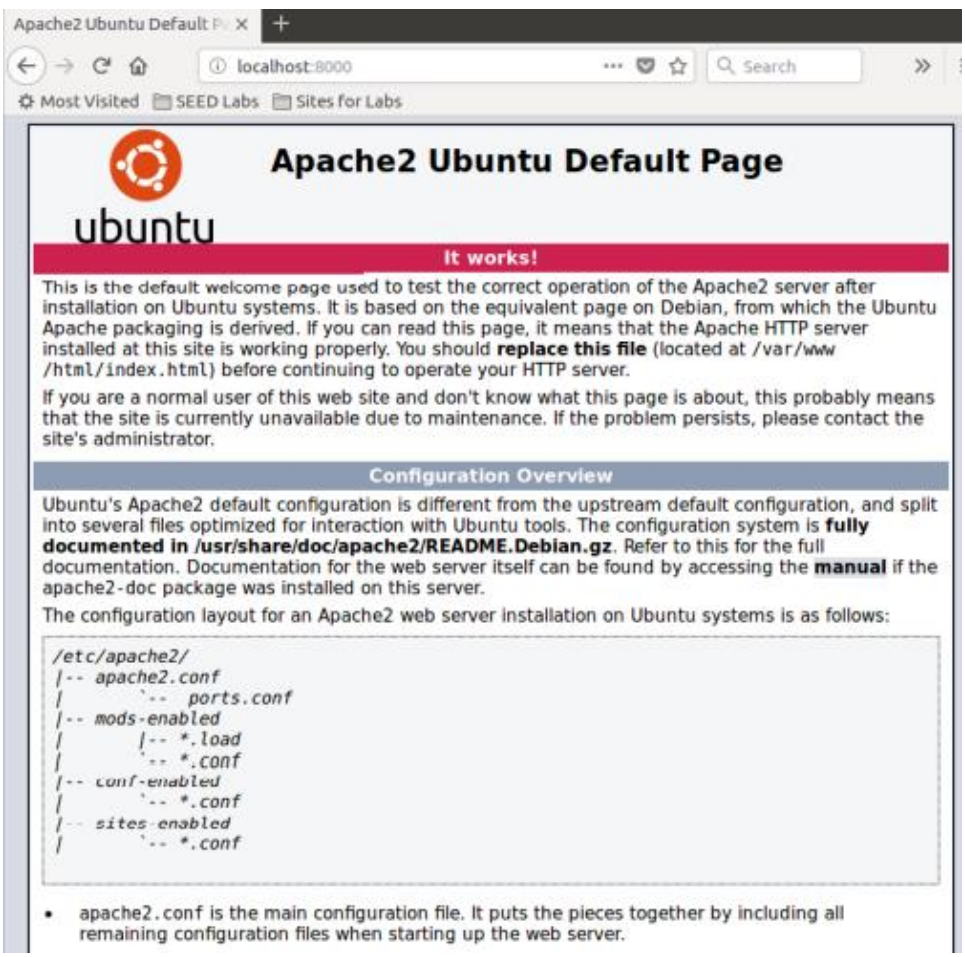
- Block Machine B from accessing port 22 and 80.

```
[10/30/22]seed@VM:~$ sudo ufw deny in from 10.0.2.8 to
10.0.2.7 port 22
Rule added
[10/30/22]seed@VM:~$ sudo ufw deny in from 10.0.2.8 to
10.0.2.7 port 80
Rule added
```

- Now used the command “ssh –R 8000:localhost:80 seed@10.0.2.8” to setup a reverse SSH on Machine A.

```
The authenticity of host '10.0.2.8 (10.0.2.8)' can't be establishe
d.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1
/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.8' (ECDSA) to the list of known
 hosts.
seed@10.0.2.8's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

- Now accessing the web page from Machine B available on Machine A.



IMPORTANT NOTE

I had to make new Machine B in due to some problems during work that's why there might be varying IP addresses.