NCL Vulnerable Environments Collection Initiative Documentation (Project Milkomeda)

Security researcher

Name: **Rayson Koh**

Vulnerability

CVE (if any): **2019-0708**
OSVDB Id (if any): **NIL**
EDB ID (if any): **NIL**
Other ID (if any): **NIL**

Environment

Operating system with version:
Windows Server 2008 R2 Enterprise (Vulnerable System), Ubuntu 18.04 (Malicious System)

Other environment information:
Username – [REDACTED], PW – [REDACTED]

Resources

Link to vulnerable source: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-0708
Link to exploit: https://github.com/Ekultek/BlueKeep
Useful Links:

- https://www.youtube.com/watch?v=AkXM2wywMN0
- https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/rdp-stands-for-really-do-patch-understanding-the-wormable-rdp-vulnerability-cve-2019-0708/

Description of Vulnerability

Short-Version:

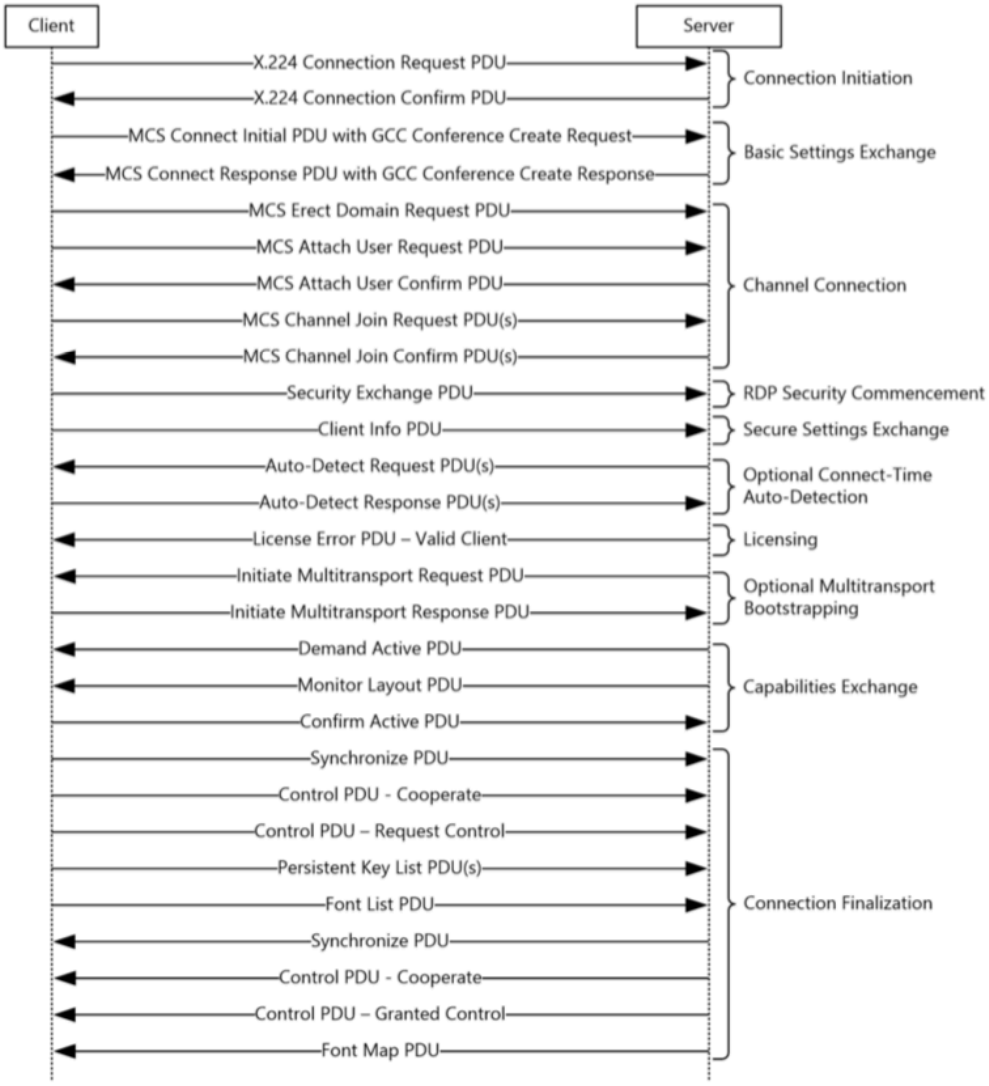
Before Authentication, Remote Desktop Protocol (RDP) negotiates various settings between Client and Server. This includes setting Static Virtual Channels for communication for different applications. These channels are set up before the security check is performed. In Windows XP, 7, Server 2008, there are 32 Static Virtual Channels by default. Windows bind Channel Names to Channel Numbers using _IcaBindVirtualChannels function in TermDD.sys.

Channel 31 is reserved for “MS_T120”. The exploit will bind “MS_T120” to another malicious Channel Number since _IcaBindVirtualChannels does not check for “MS_T120” and ensures it only binds to Channel 31.

When the attacker sends crafted packets through the malicious channel, TermDD.sys will attempt to close that channel, while the default Channel Number 31 remains open. This allows for a “use after free” vulnerability where the attacker would have Kernel-Level Privileges.

Long Version:

The connection sequence for RDP is as follows.



The vulnerability lies in **MCS Connect Initial PDU with GCC Conference Create Request** in the **Basic Settings Exchange** stage.

The **“MCS Connect Initial and GCC Create”** request contains security-related information, virtual channels creation information and other supported RDP client capabilities.

The typical **MCS Connect Initial PDU with GCC Conference Create Request** structure is as follows.

Offset (bytes)	Size	Description	
0x00	4	tpktHeader (TPKT header)	
0x04	3	x224 (Data TPDU)	
0x07	m	mcsCi	(variable)
0x07+m	n	gccCCrq	(variable)
0x07+m+n	variable	Settings Data Blocks	

Settings Data Blocks is a concatenation of one or more **Settings Data Block.** In particular, the **CS_NET Settings Data Block** (also called **clientNetworkData**) contains a list of virtual channels. Its structure is as follows.

Offset (bytes)	Size	Description
0x00	2	Type (0xC003)
0x02	2	Length
0x04	4	channelCount (N)
0x08	8	channelName_1
0x10	4	channelOption_1
0x14	8	channelName_2
0x1C	4	channelOption_2
...		
0x08+(N-1)*12	8	channelName_N
0x10+(N-1)*12	4	channelOption_N

RDP uses Static Virtual Channels as communication links for various RDP components and user extensions. For example, “rdpdr” (Redirection), “rdpsnd” (Sound), “cliprdr” (Clipboard Sharing). These channels are created before security check is performed.

Microsoft creates channel “MS_T120” by default and binds it to Channel 31. Clients are not expected to create “MS_T120” Channel.

Channels are created using IcaCreateChannel function in TermDD.sys which first checks if the name exists and if not, allocates a channel structure (**ChannelControlStructure**) to create it. A pointer to the channel structure is stored within a table (**ChannelPointerTable**) through the function **_IcaBindVirtualChannels** in TermDD.sys. It looks as follows.

```
Slot Number         ChannelControlStructure pointer
-----------         -------------------------------
0                   Empty
1                   Empty
2                   Empty
3                   Empty
4                   Empty
5                   Empty
6                   Empty
7                   Pointer to CTXTW
8                   Empty
...
0x1F                Pointer to MS_T120
```

Each slot can store a **ChannelControlStructure** pointer. When RDP client connects and attempts to open channels specified in the **clientNetworkData**, the corresponding **ChannelControlStructure** is created and the pointer is stored in the **ChannelPointerTable** through the function **_IcaBindVirtualChannels** in TermDD.sys. If a channel with the name "MS_T120" is specified in **clientNetworkData,** the pointer for its **ChannelControlStructure** is stored in the **ChannelPointerTable** through **_IcaBindVirtualChannels.**

```
43        v9 = IcaFindChannelByName(v4, (PERESOURCE)5, (char *)(v7 - 8));
44        v10 = v9;
45        if ( v9 )
46        {
47          IcaReferenceStack(v9);
48          KeEnterCriticalRegion();
49          ExAcquireResourceExclusiveLite((PERESOURCE)(v10 + 12), 1u);
50          _IcaBindChannel(v10, 5, *(_WORD *)v7, *(_DWORD *)(v7 + 2));
51          ExReleaseResourceLite((PERESOURCE)(v10 + 12));
52          KeLeaveCriticalRegion();
53          IcaDereferenceChannel((PVOID)v10);
54          IcaDereferenceChannel((PVOID)v10);
55          v4 = *(_DWORD *)(a1 - 468);
56        }
57        ++*(_DWORD *)(a1 - 456);
58        v7 += 14;
59      }
60      while ( *(_DWORD *)(a1 - 456) < *(_DWORD *)(a1 - 464) );
61    }
```

```
46    v3 = IcaFindChannelByName(v1, (PERESOURCE)5, (char *)v2 - 10);
47    v4 = v3;
48    if ( v3 )
49    {
50      IcaReferenceStack(v3);
51      KeEnterCriticalRegion();
52      ExAcquireResourceExclusiveLite((PERESOURCE)(v4 + 12), 1u);
53      v5 = __stricmp((const char *)(v4 + 88), "MS_T120");
54      v7 = *v2;
55      if ( v5 )
56        _IcaBindChannel(v4, 5, *((_WORD *)v2 - 1), v7);
57      else
58        _IcaBindChannel(v4, 5, 31, v7);
59      ExReleaseResourceLite((PERESOURCE)(v4 + 12));
60      KeLeaveCriticalRegion();
61      IcaDereferenceChannel((PVOID)v4);
62      IcaDereferenceChannel((PVOID)v4);
63      v1 = v15;
64    }
```

Figure 1 - Left: Before Patch, Right: After Patch

After the Patch, in the _IcaBindVirtualChannels function, it checks whether the channel name is MS_T120 and binds it to channel 31.

When the channels are opened after **MCS Channel Join Request,** if an attacker attempts to send specially crafted data into the **MS_T120** channel, TermDD.sys attempts to close the channel. This frees the pointer to the **MS_T120** channel in the **ChannelPointerTable** as well as its **ChannelControlStructure.** However the pointer at Channel 31 (0x1F) is not cleared. This leads to a use-after-free condition.

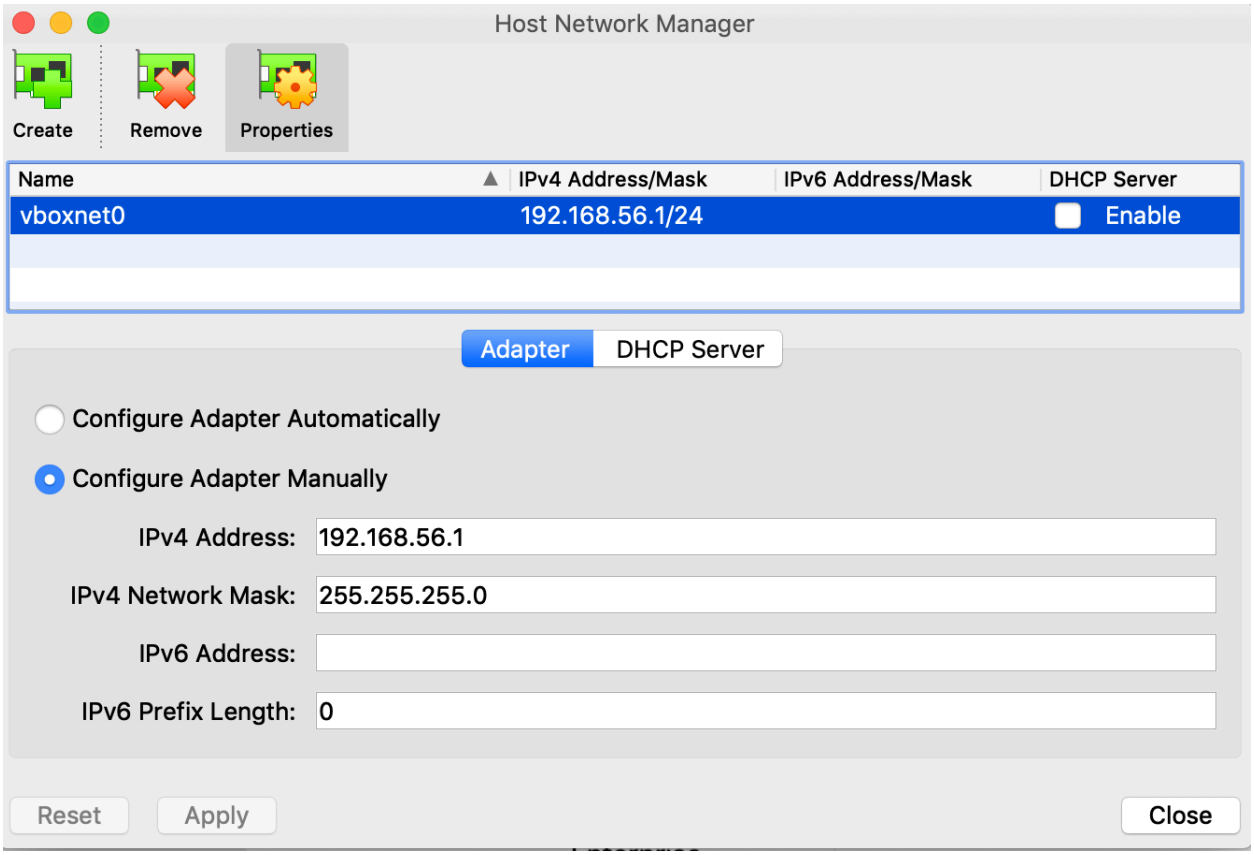
Environment Configuration/Installation

[Here we assume a fresh installed operating system]

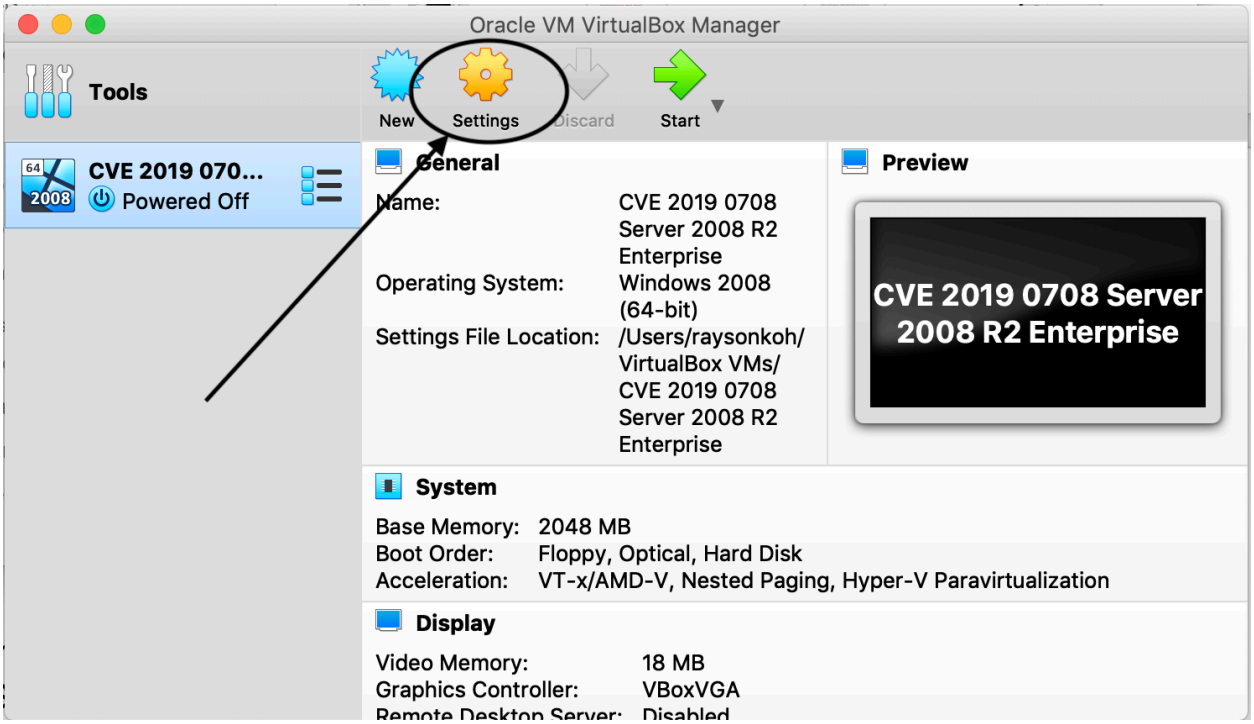
VirtualBox configuration for Windows Server 2008 R2 Enterprise (Vulnerable VM):

1. Import .ova file.
2. In the Virtualbox program, click on **File → Host Network Manager.**

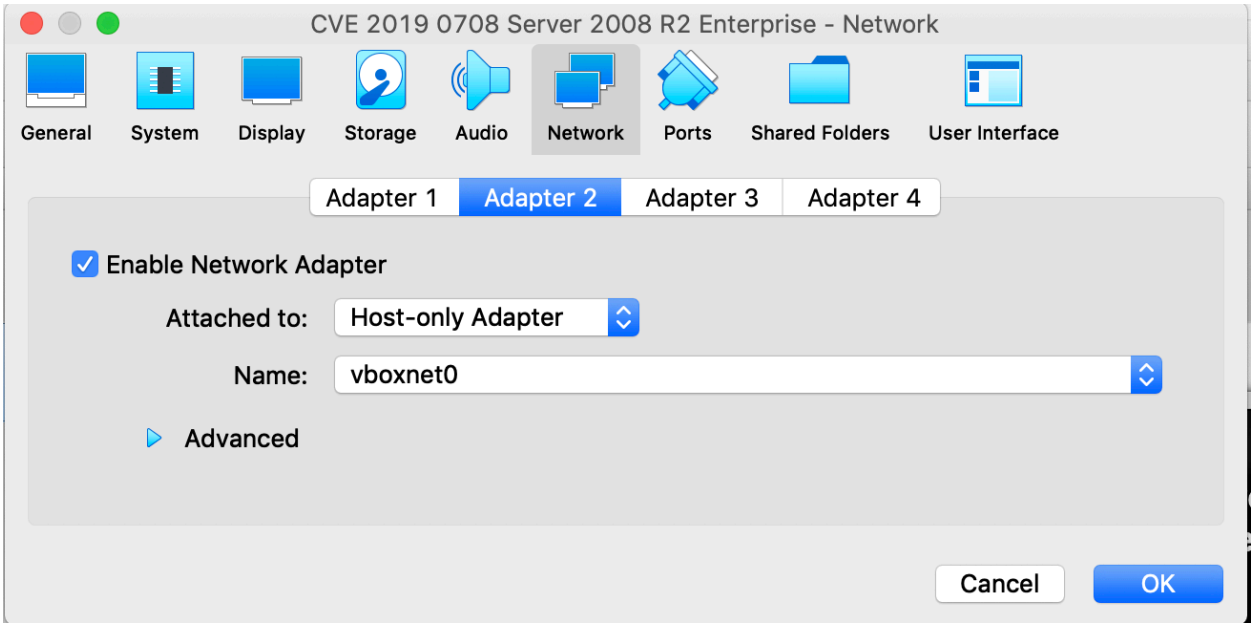
3. Click **Create** and **disable DHCP services.** See my host-only adapter set-up.



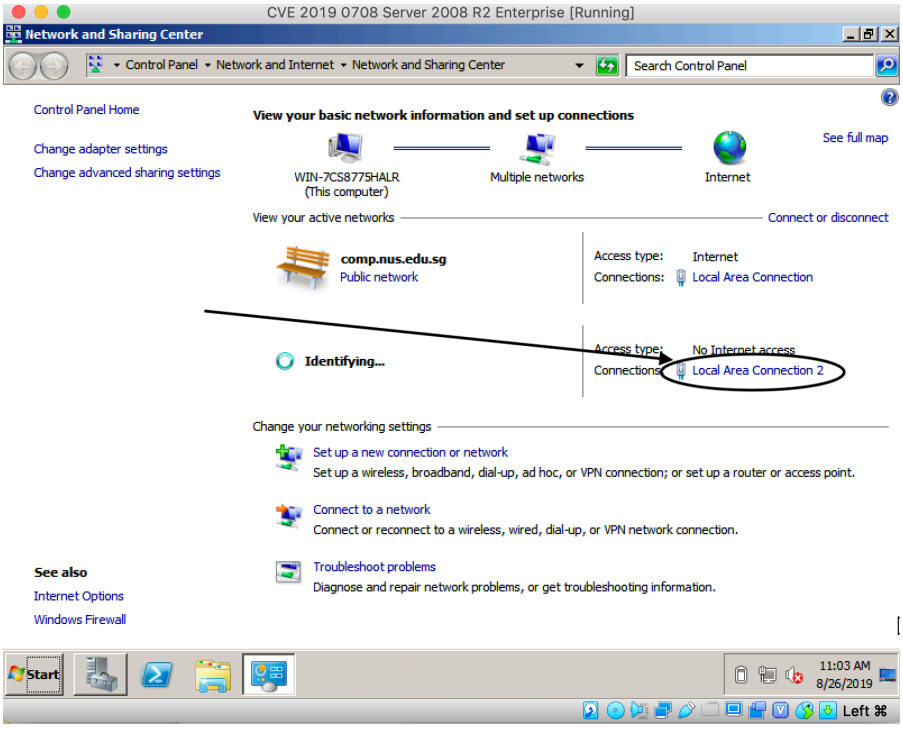
4. Go to Settings (ensure VM is shut-down)

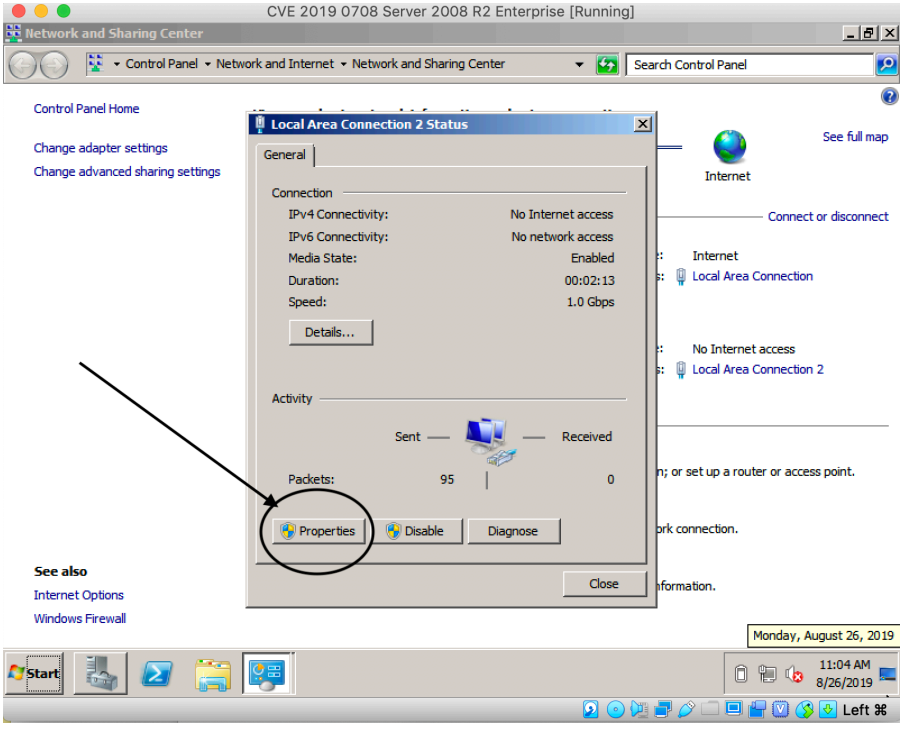


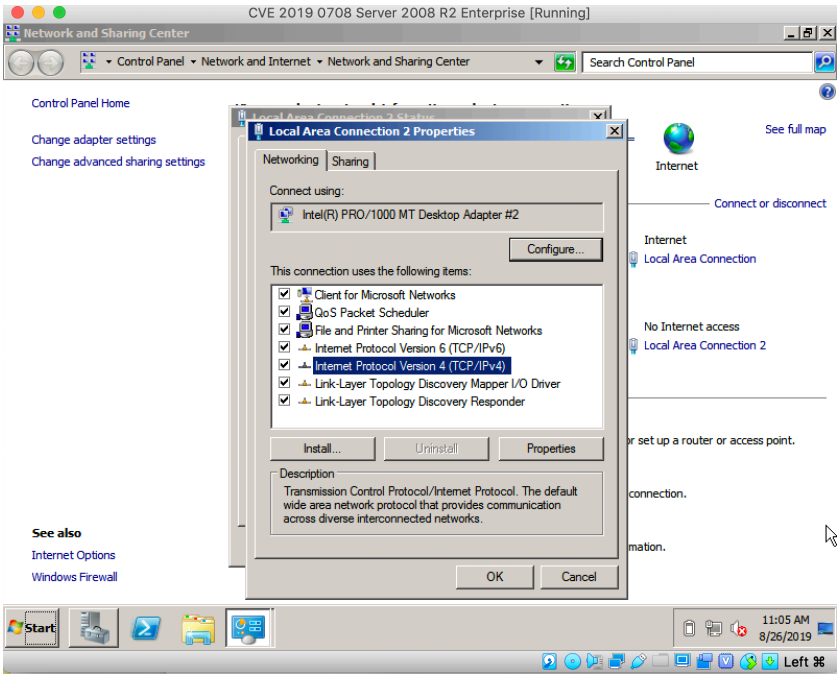
5. Go to **Adapter 2, Enable Network Adapter, Attached to: Host-only Adapter** and select the name of the host-only network that was created. Click **OK.**

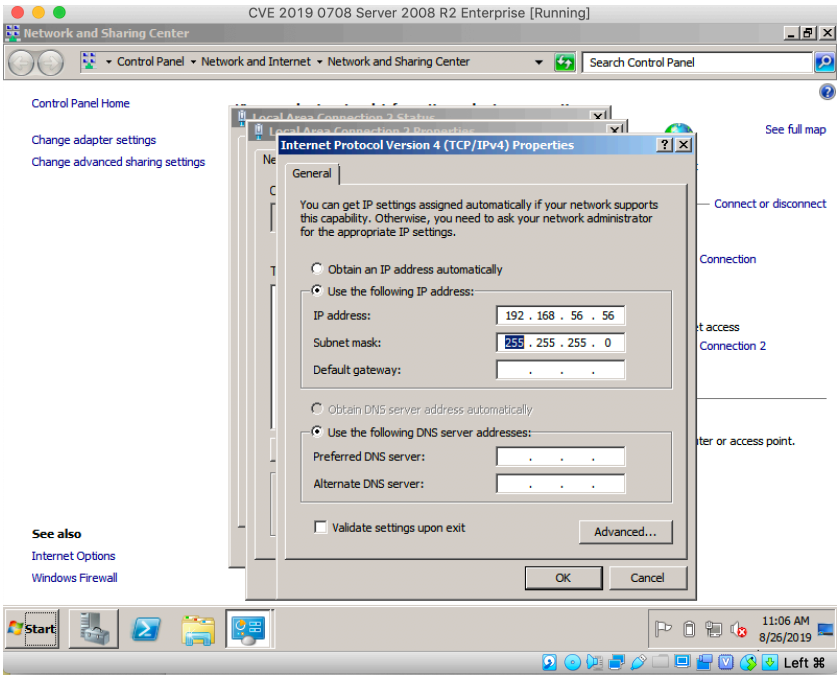


6. On the VM, setup static IP in the same IP subnet as the host. (In my case, it is 192.168.56.x).

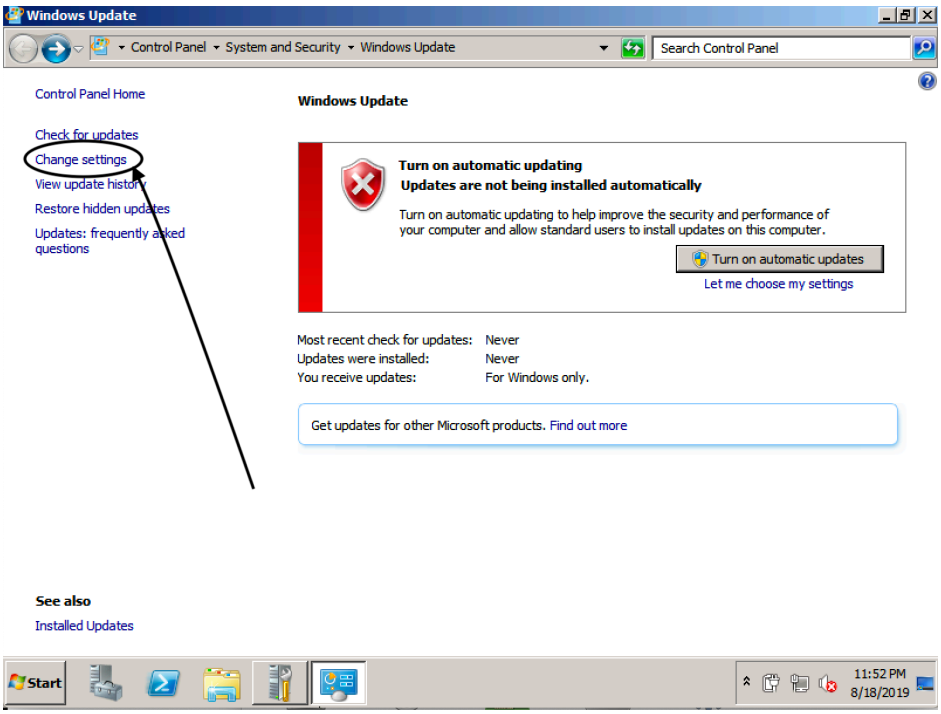


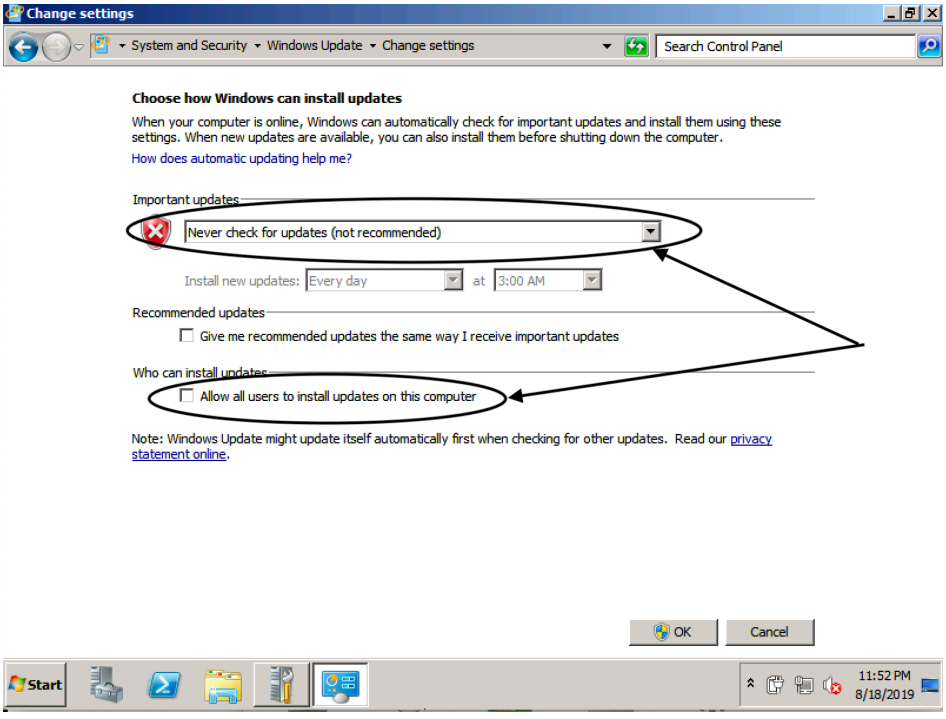



CVE 2019 0708 Server 2008 R2 Enterprise [Running]
Network and Sharing Center
Control Panel ▸ Network and Internet ▸ Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
See full map
Internet
Connect or disconnect
Internet
Local Area Connection
No Internet access
Local Area Connection 2
Local Area Connection 2 Properties
Networking
Sharing
Connect using:
Intel(R) PRO/1000 MT Desktop Adapter #2
Configure...
This connection uses the following items:
Client for Microsoft Networks
QoS Packet Scheduler
File and Printer Sharing for Microsoft Networks
Internet Protocol Version 6 (TCP/IPv6)
Internet Protocol Version 4 (TCP/IPv4)
Link-Layer Topology Discovery Mapper I/O Driver
Link-Layer Topology Discovery Responder
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel
r set up a router or access point.
connection.
mation.
See also
Internet Options
Windows Firewall
Start
11:05 AM
8/26/2019
Left ⌘

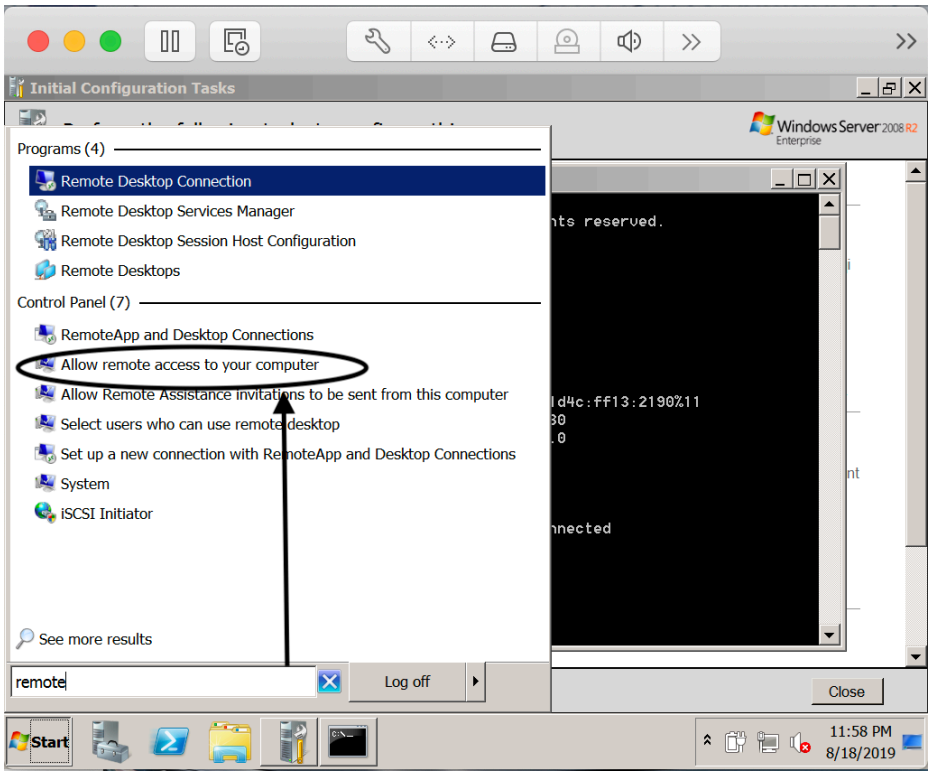

CVE 2019 0708 Server 2008 R2 Enterprise [Running]
Network and Sharing Center
Control Panel ▸ Network and Internet ▸ Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
See full map
Connect or disconnect
Connection
Connection 2
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
192 . 168 . 56 . 56
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel
ter or access point.
See also
Internet Options
Windows Firewall
Start
11:06 AM
8/26/2019
Left ⌘

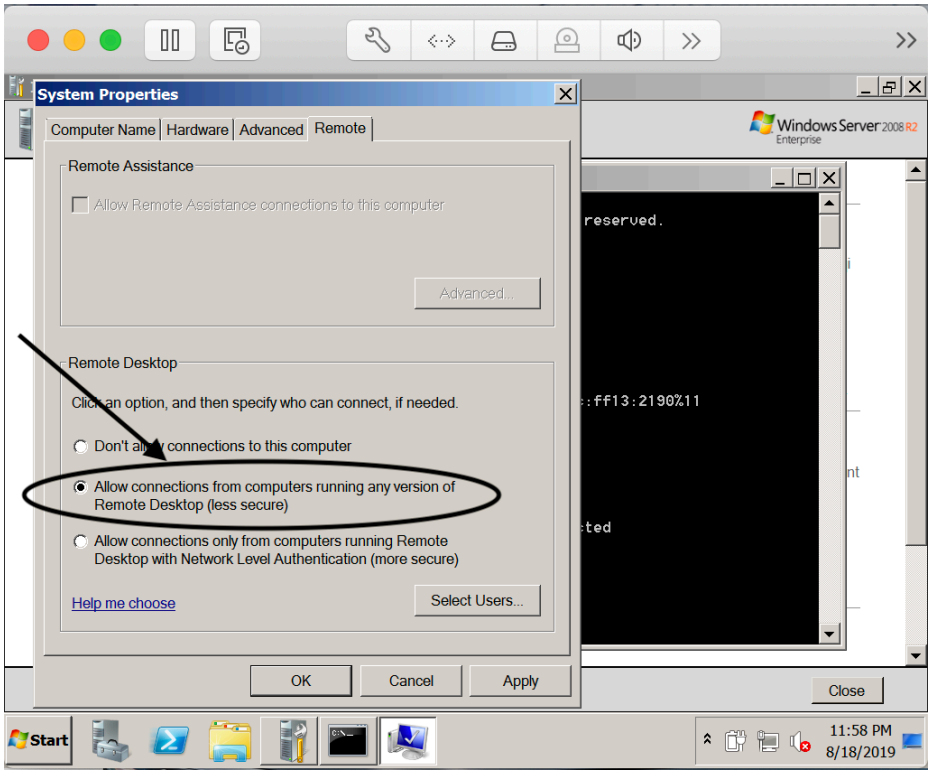
7. Set Windows Update to **Never Check for Updates.**



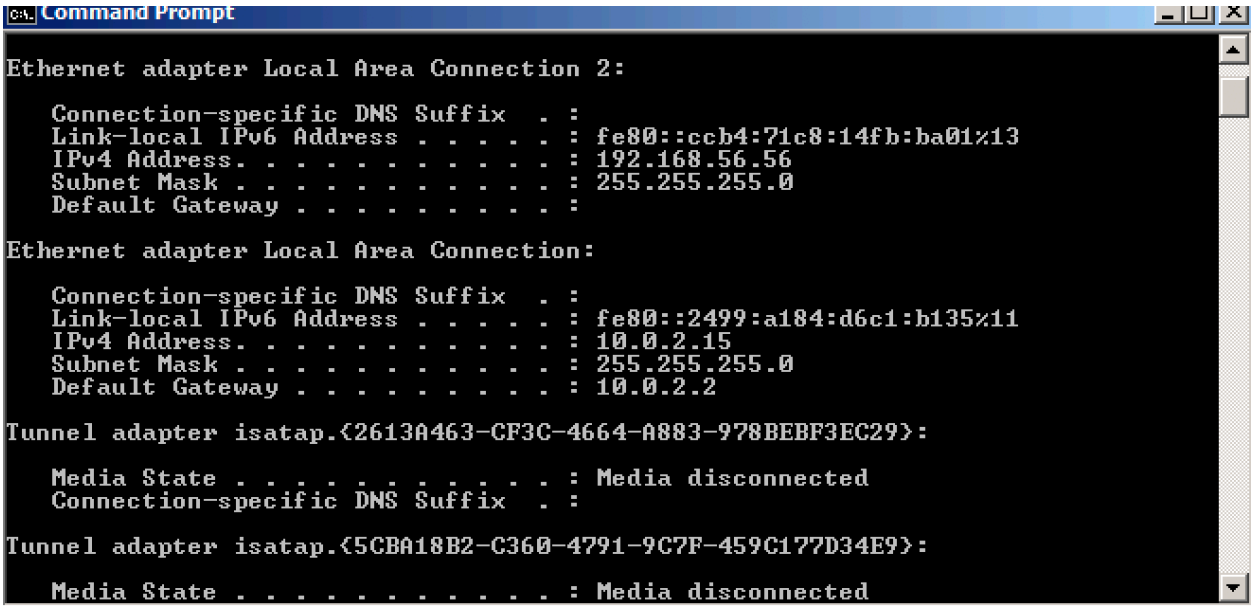


8. Set **Allow remote access to your computer** to **Allow connections from computers running any version of Remote Desktop.**



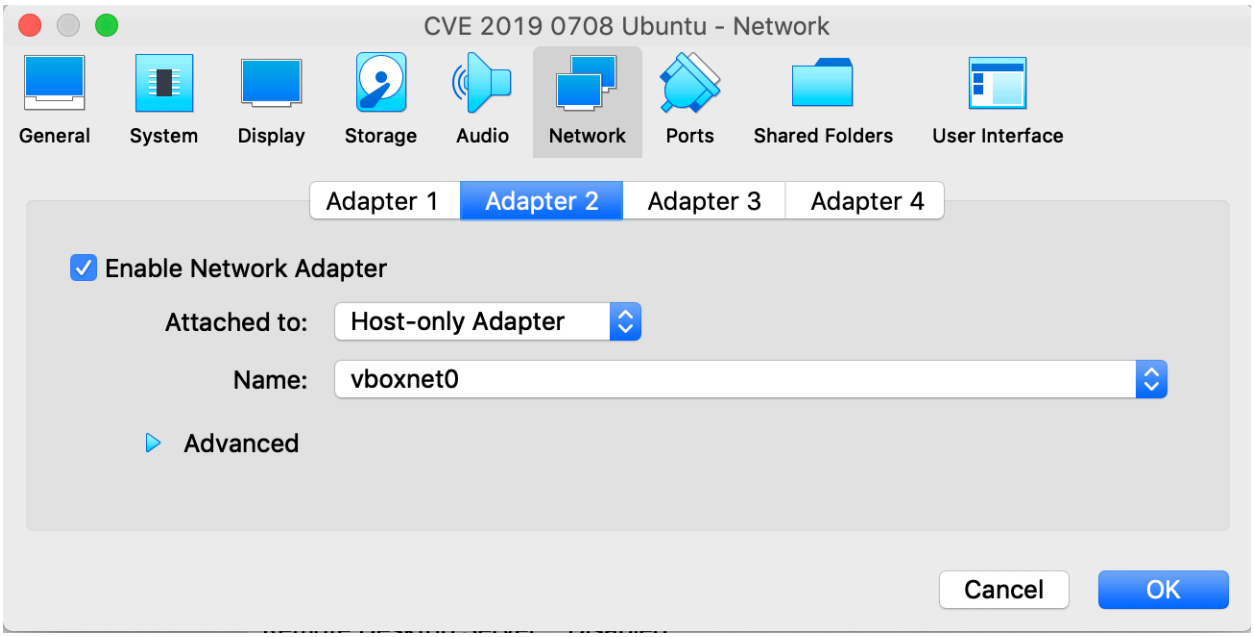


9. Check IP Address using **ipconfig** in **command prompt.**

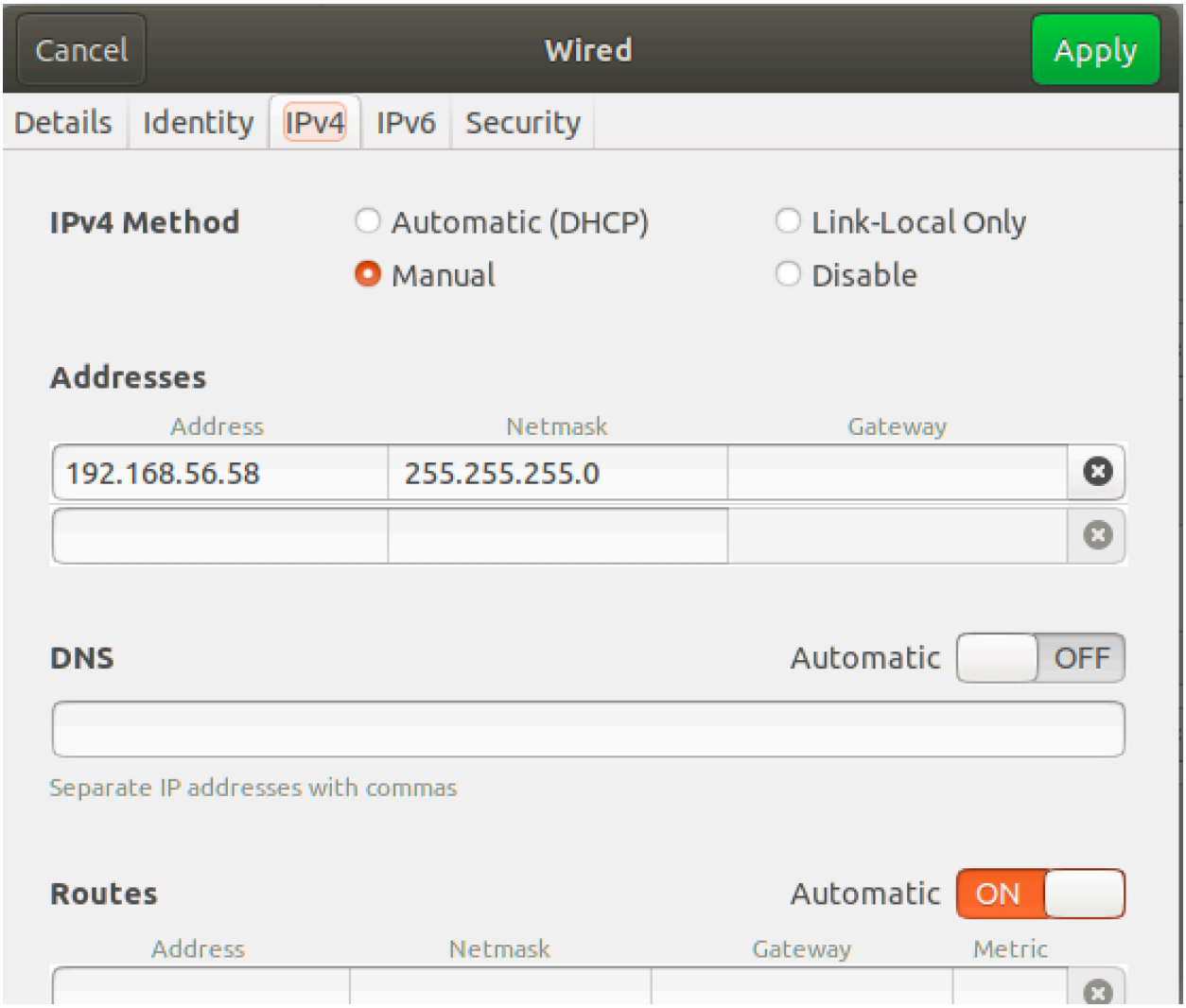


VirtualBox configuration for Ubuntu (Malicious VM):

1. Ensure Adaptor 1 is set to **NAT (default)** and Adaptor 2 is set to **Host-only Adapter** (should be the same as the adapter for the vulnerable VM)



2. Start the Ubuntu VM. Go to **Settings → Network.** Choose the Network that corresponds to the **Host-only Adapter.** Go to **ipv4 → Addresses** and add an IP address in the same subnet as the Vulnerable VM. (In this case, it is 192.168.56.x). Click **Apply.**

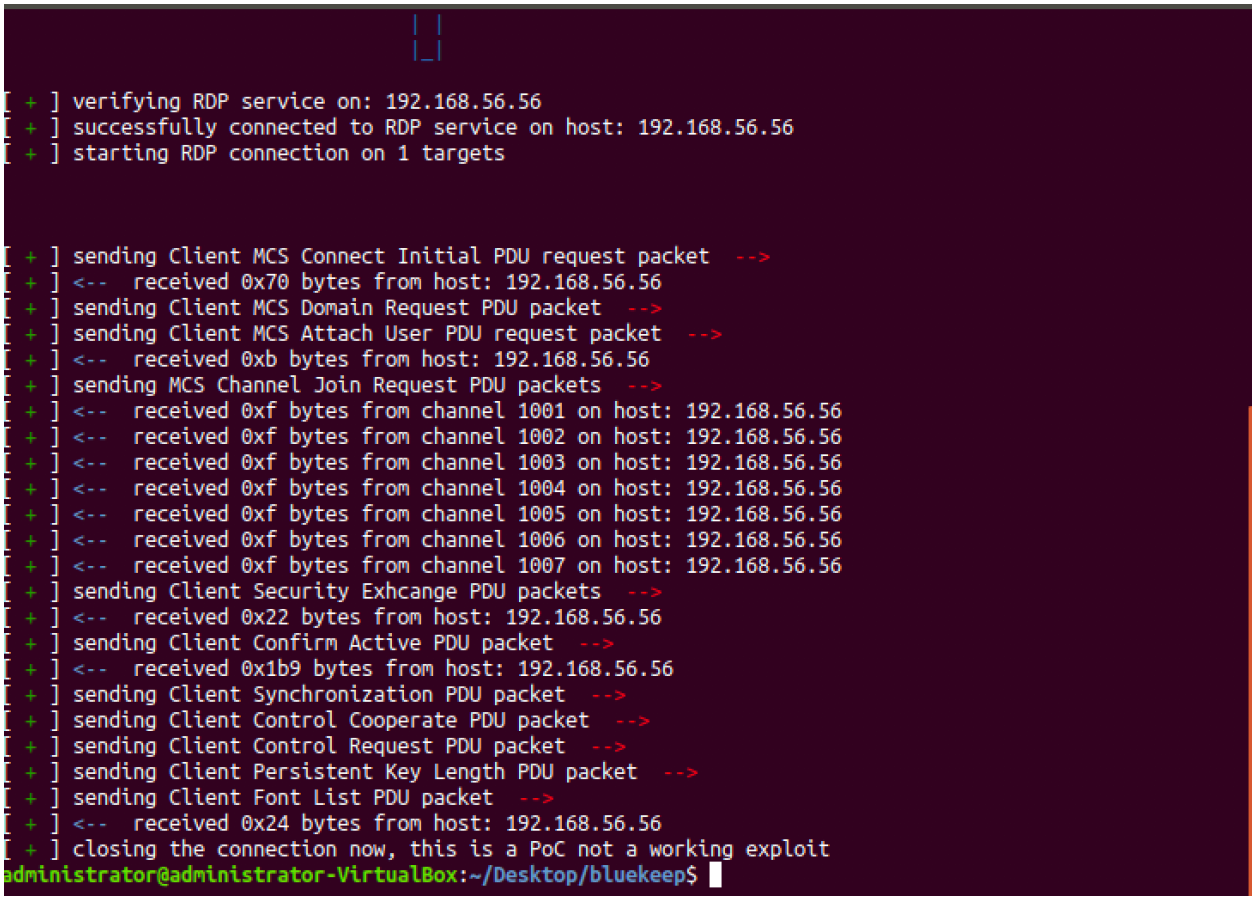


Exploitation steps **(run on Malicious Ubuntu VM)**

1. In Terminal, run **cd desktop/bluekeep**.

2. Run **`python bluekeep_poc.py –i 192.168.56.56 (ip_address of server in this case)`**

3. You should see this if successful.



4. To launch a DOS attack, run **`python bluekeep_weaponized_dos.py –i 192.168.56.56 (ip_address of server in this case)`**
5. You should see these if successful.

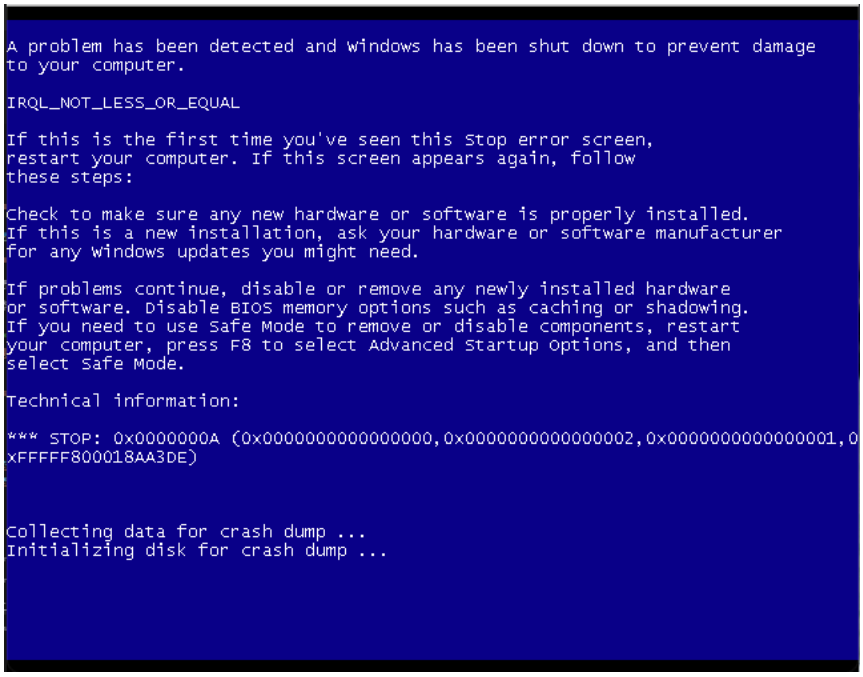
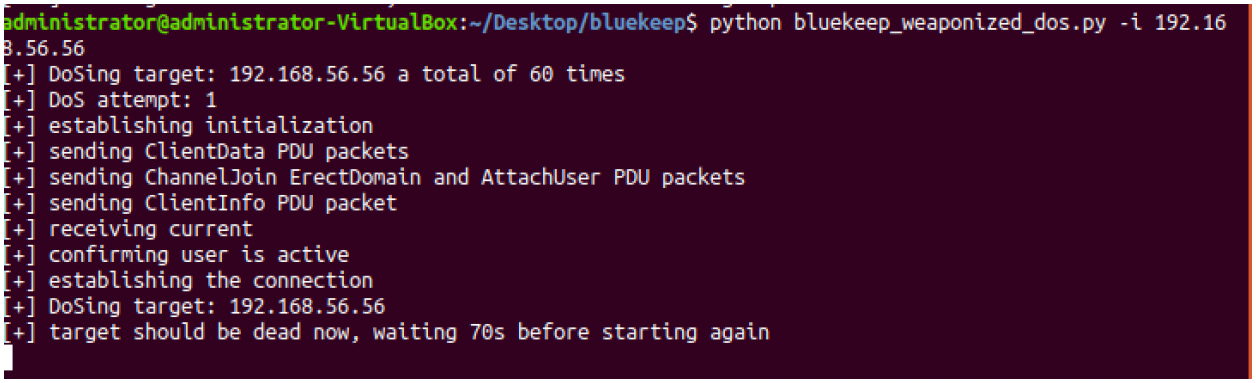


Figure 2 - Windows Server Crash



Other comments

1. To send a payload, edit bluekeep_poc.py under **start_rdp_connection** function, line 367.
2. Solution: Use Network-Level Authentication for RDP, block port 3389 (default RDP port) if RDP not in use and Update Windows to latest patch.