

Persistence – Windows Setup Script

When the Windows Operating system is installed via a clean installation or via an upgrade, the Windows Setup binary is executed. The Windows setup allows custom scripts to be executed such as the *SetupComplete.cmd* and *ErrorHandler.cmd* to enable the installation of applications or the execution of other tasks during or after the Windows setup process is completed. These scripts are stored in the following location:

```
%WINDIR%\Setup\Scripts\SetupComplete.cmd
```

```
%WINDIR%\Setup\Scripts\ErrorHandler.cmd
```

Using the *ErrorHandler.cmd* script it is possible to execute arbitrary code when the Windows operating system is upgraded. Even though it could be considered as an unconventional tactic, it could be combined with scheduled tasks for example to run Windows Setup and establish persistence. The following code can be used as a proof of concept of code execution that will display a message box when the Windows Setup binary is initiated:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Windows_setup1
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            string message = "Visit pentestlab.blog";
            string title = "Pentestlaboratories";
            MessageBox.Show(message, title);
        }
    }
}
```

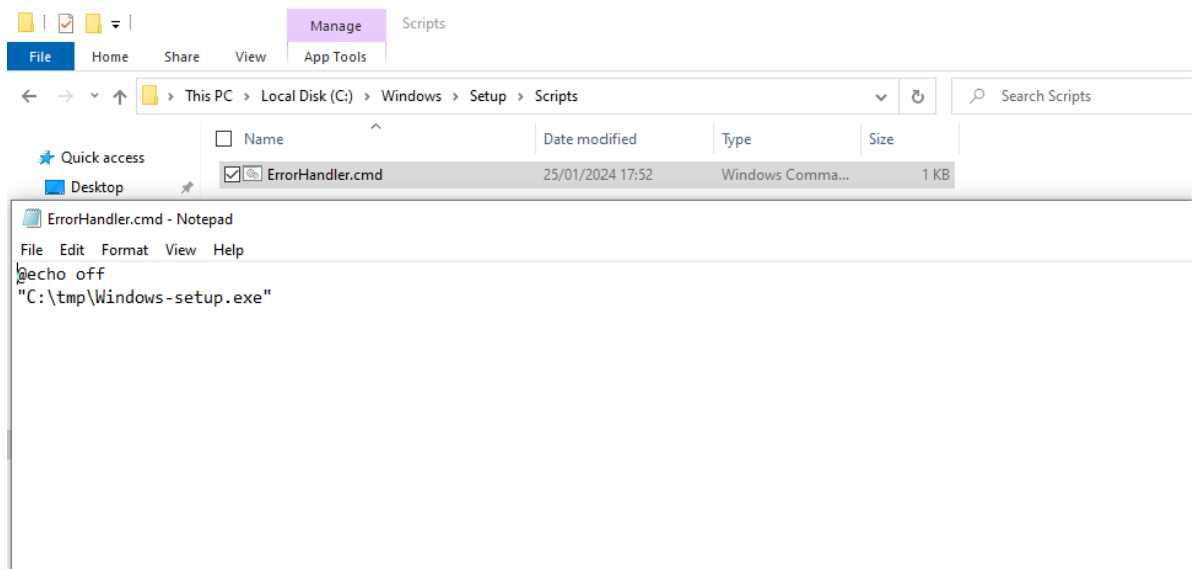
```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace Windows_setup1
8  {
9      0 references
10     internal static class Program
11     {
12         [STAThread]
13         0 references
14         static void Main()
15         {
16             string message = "Visit pentestlab.blog";
17             string title = "Pentestlaboratories";
18             MessageBox.Show(message, title);
19         }
20     }
21 }

```

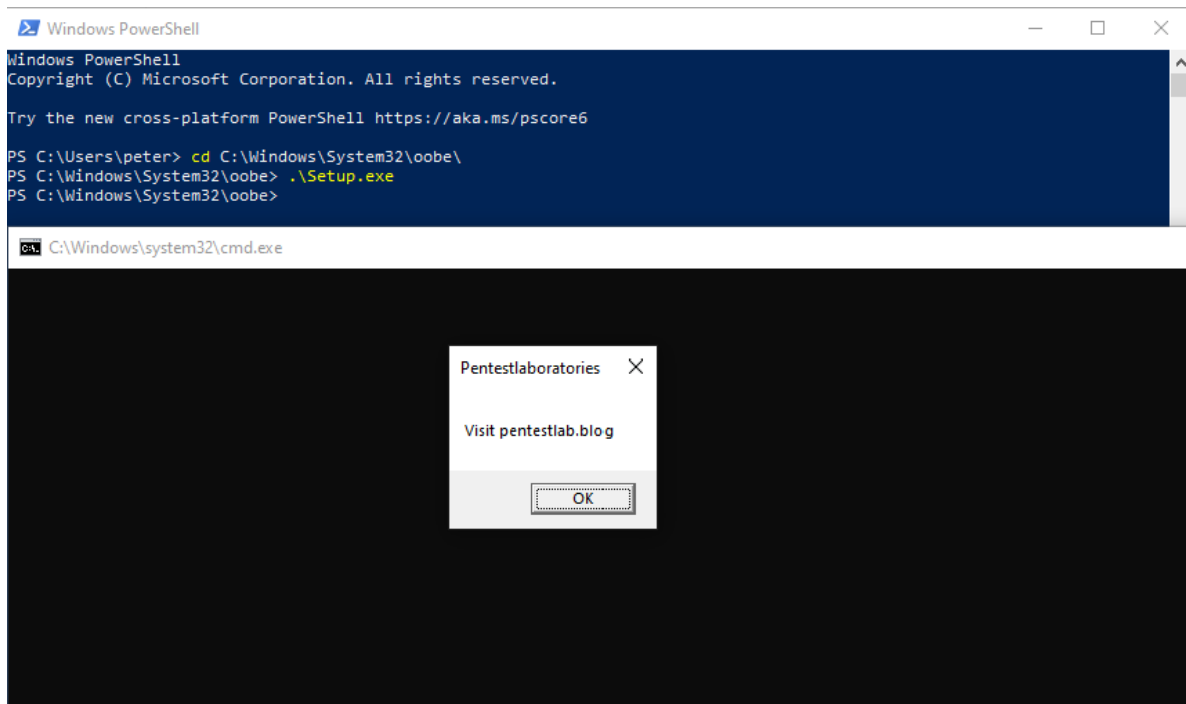
Windows Setup Script – Message Box Code

Since the Windows Setup will look during execution and when an error is caused in the setup process for the presence of *ErrorHandler.cmd* inside the *Scripts* folder, it is possible to use this script to execute arbitrary code.



Windows Setup Script Path

Running the *setup.exe* will cause an error which as a result will force the execution of *ErrorHandler.cmd* script.



Windows Setup Script – Message Box

Replacing the message box executable with an implant will allow a command and control session to be established.

The image shows the Havoc application interface. At the top is a menu bar with "Havoc", "View", "Attack", "Scripts", and "Help". Below the menu bar is a table with the following columns: "ID", "External", "Internal", "User", "Computer", "OS", "Process", "PID", "Last", and "Health". There is one row of data in the table. The table is set against a dark background.

ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
76632ff8	10.0.0.2	0.0.0.0	Administrator	WK01	Windows 10	demon.x64.exe	3980	2s	healthy

Windows Setup Script – C2

The process tree of the implant is specified below:

Setup.exe --> cmd.exe --> demon.x64.exe

Process Explorer - Sysinternals: www.sysinternals.com [RED\Administrator] (Administrator)

File Options View Process Find Users Help

<Filter by name>

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe	< 0.01	3,760 K	6,420 K	4444	Host Process for Windows S...	Microsoft Corporation
NisSrv.exe		3,816 K	872 K	5048		
SearchIndexer.exe		17,880 K	6,776 K	5200	Microsoft Windows Search I...	Microsoft Corporation
SearchProtocolHost.exe		2,548 K	7,572 K	1520	Microsoft Windows Search P...	Microsoft Corporation
SearchFilterHost.exe		1,692 K	6,172 K	5228	Microsoft Windows Search F...	Microsoft Corporation
SecurityHealthService.exe		3,452 K	8 K	3084		
SgmBroker.exe		3,300 K	908 K	3948		
svchost.exe		2,508 K	1,316 K	3200		
svchost.exe		2,564 K	0 K	5160		
svchost.exe		1,748 K	796 K	788		
TrustedInstaller.exe	< 0.01	1,780 K	1,012 K	5176	Windows Modules Installer	Microsoft Corporation
lsass.exe		5,952 K	6,364 K	664	Local Security Authority Proc...	Microsoft Corporation
fontdrvhost.exe		1,272 K	8 K	792	Usermode Font Driver Host	Microsoft Corporation
csrss.exe	< 0.01	3,272 K	7,648 K	508		
winlogon.exe		2,448 K	444 K	596	Windows Log-on Application	Microsoft Corporation
fontdrvhost.exe		3,080 K	2,220 K	800	Usermode Font Driver Host	Microsoft Corporation
dwm.exe	< 0.01	76,580 K	14,924 K	368	Desktop Window Manager	Microsoft Corporation
explorer.exe	0.72	68,212 K	47,064 K	4324	Windows Explorer	Microsoft Corporation
SecurityHealthSystray.exe		1,728 K	0 K	3652	Windows Security notificatio...	Microsoft Corporation
vmtoolsd.exe	< 0.01	17,872 K	4,144 K	5548	VMware Tools Core Service	VMware, Inc.
OneDrive.exe		8,720 K	1,028 K	5856	Microsoft OneDrive	Microsoft Corporation
powershell.exe	< 0.01	65,824 K	700 K	1976	Windows PowerShell	Microsoft Corporation
conhost.exe		4,172 K	908 K	3220	Console Window Host	Microsoft Corporation
Setup.exe		1,080 K	56 K	1072	Windows Installation and Set...	Microsoft Corporation
cmd.exe		3,688 K	68 K	3592	Windows Command Processor	Microsoft Corporation
conhost.exe		6,884 K	1,804 K	6124	Console Window Host	Microsoft Corporation
demon.x64.exe	< 0.01	3,292 K	2,748 K	3980		
procexp64.exe	2.87	22,384 K	33,916 K	3444	Sysinternals Process Explorer	Sysinternals - www.sysinter...

CPU Usage: 5.02% Commit Charge: 50.61% Processes: 90 Physical Usage: 38.38%

Windows Setup Script – Process Tree

References

1. <https://www.hexacorn.com/blog/2022/01/16/beyond-good-ol-run-key-part-135/>
2. <https://cocomelonc.github.io/persistence/2023/07/16/malware-pers-22.html>