WHOAMI /ALL
Chief Technical Architect – Microsoft Security
Most Valuable Professional
Microsoft Certified Trainer
Giac Certified Penetration Tester
Microsoft infrastructure and security expert (security researcher)
15 years+ with Microsoft technology
http://oddvar.moe
I like memes and gifs
@oddvarmoe
advania
nic
Microsoft
Domene
Users
Ola Normann

ODDVAR MOE'S BLOG

Notes from My adventures with Windows security

BYPASSING DEVICE GUARD UMCI USING CHM – CVE-2017-8625

Posted on 13 Aug 2017

TL;DR
You could/can bypass Device Guard user mode code integrity with a custom CHM and execute code.

The last 6 months I have done some security research on my (little) spare time, because I find that very interesting. During this time, I was lucky enough to find another valid Device Guard UMCI bypass (I found the bginfo.exe as well: https://msitpros.com/?p=3831) and reported this to the Microsoft Security Response Center (secure@microsoft.com).

After a dialog with MSRC they told me that this was already discovered by another security researcher and that this would become a CVE with my name on it as well. (This was big news to me)
MSRC could not tell who the other researcher was, but Matt Graeber knew (Love that guy) . The other researcher was Matt Nelson and he had found this bypass a while back. Awesome!

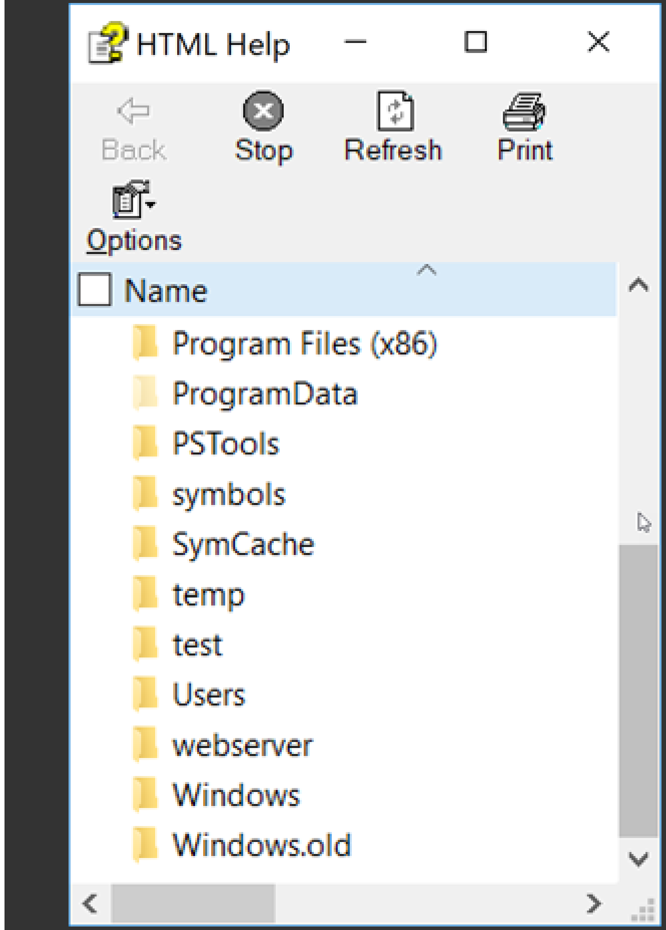
Anyways this blogpost looks into how I made the discovery and some PoC code as well.

I am not a hardcore reverse engineer (yet), so I will not do a full disassembly tour of the fix Microsoft has done, but rather focusing on how I found this.

For me this discovery started as I was looking through binary files in the windows and the system32 folder. I do this from time to time to discover new stuff.

I stumbled upon a binary file called hh.exe.
I went on and ran “hh.exe /?”. Assuming I either would get some help on the command or an error. Well, that is not what happened.

I got this instead:

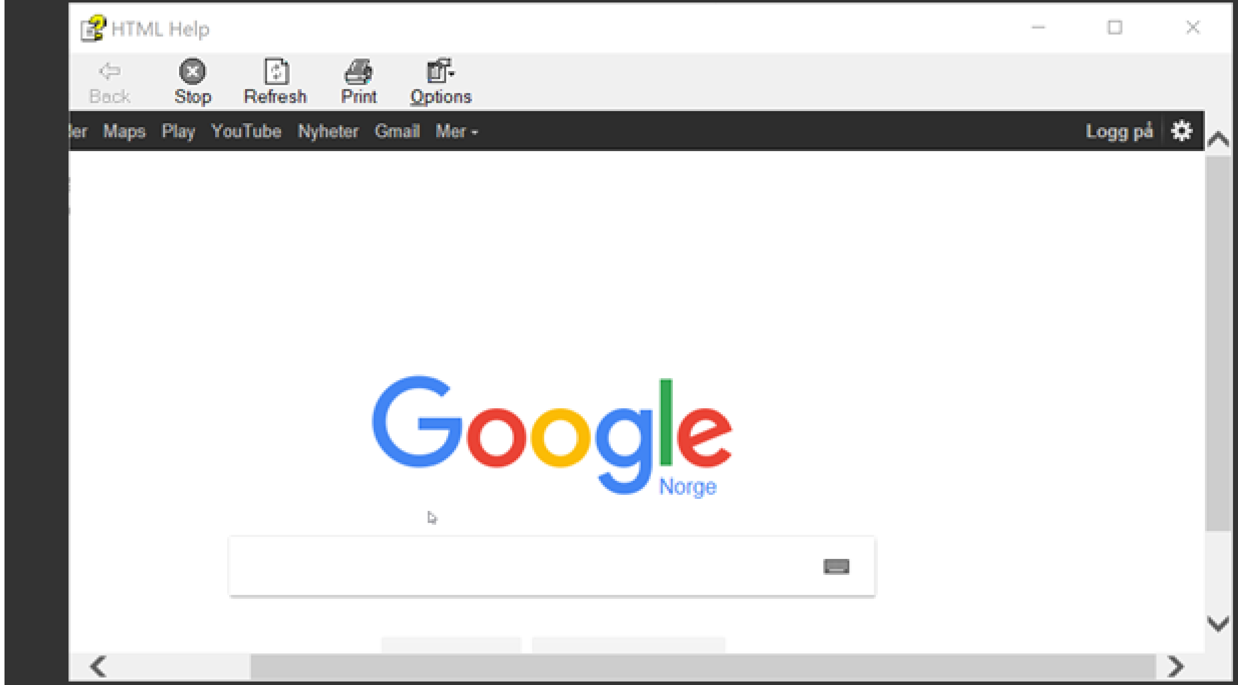


Name
Program Files (x86)
ProgramData
PSTools
symbols
SymCache
temp
test
Users
webserver
Windows
Windows.old

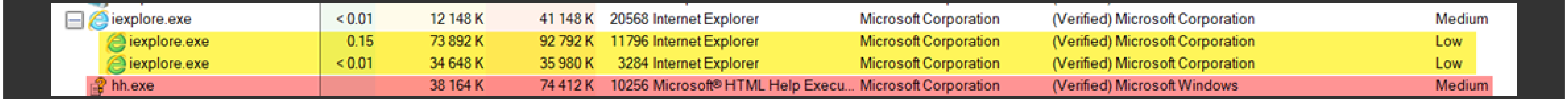
This triggered my curiosity to the max. This trick btw still works on the latest Windows 10 and I guess if you ever need to have an explorer in a locked down environment (Terminal server etc) this could do it. You could also run for instance “hh.exe c:”.

After trying a lot of different stuff, I realized you could also browse the internet through hh.exe. Just by typing “hh.exe http://www.google.com&#8221; proves this. (Still works in the latest Windows 10)

It looks like this:



The first thing that struck my mind was to check the integrity level of the hh.exe process and guess what….



iexplore.exe	< 0.01	12 148 K	41 148 K	20568 Internet Explorer	Microsoft Corporation	(Verified) Microsoft Corporation	Medium
iexplore.exe	0.15	73 892 K	92 792 K	11796 Internet Explorer	Microsoft Corporation	(Verified) Microsoft Corporation	Low
iexplore.exe	< 0.01	34 648 K	35 980 K	3284 Internet Explorer	Microsoft Corporation	(Verified) Microsoft Corporation	Low
hh.exe		38 164 K	74 412 K	10256 Microsoft® HTML Help Execu...	Microsoft Corporation	(Verified) Microsoft Windows	Medium

Yeah, my “browser” inside hh.exe is running in medium integrity mode and a normal iexplore process runs in Low. This should make it easier to exploit the browser inside hh.exe. (There is more research to be done here)

After trying a lot of different approaches, I thought I would try to create a custom help file with code inside, since HH.exe is primarily used for displaying help files.

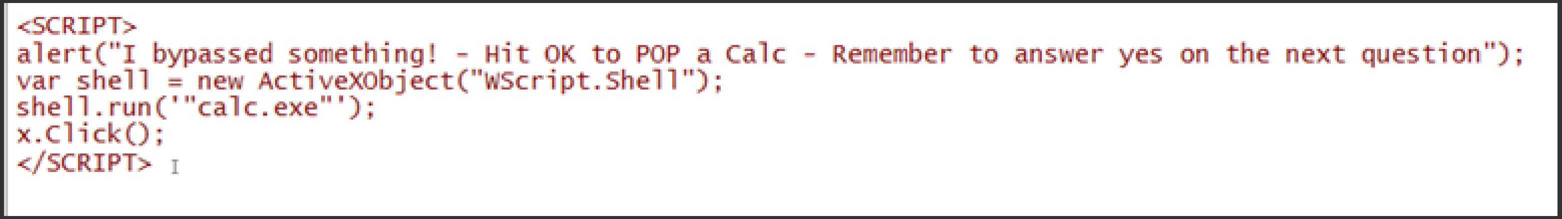
I searched the web for any valid sources of these kinds of projects and stumbled upon this:
https://raw.githubusercontent.com/samratashok/nishang/master/Client/Out-CHM.ps1
(thanks to Nikhil “SamratAshok” Mittal)

To run this script, I had to download and install the “HTML Help Workshop and Documentation” first:
http://www.microsoft.com/en-us/download/details.aspx?id=21138

I ended up creating my own version of Nikhil’s script, since I only wanted to prove my PoC and pop a Calculator.
My PoC script can be found here:
https://gist.githubusercontent.com/api0cradle/95ae3c7120f16255d94088bd8959f4b2/raw/fa25b85e85bbb64c5cf021adf92b125357086a6f/GenerateCHM_1.0.ps1

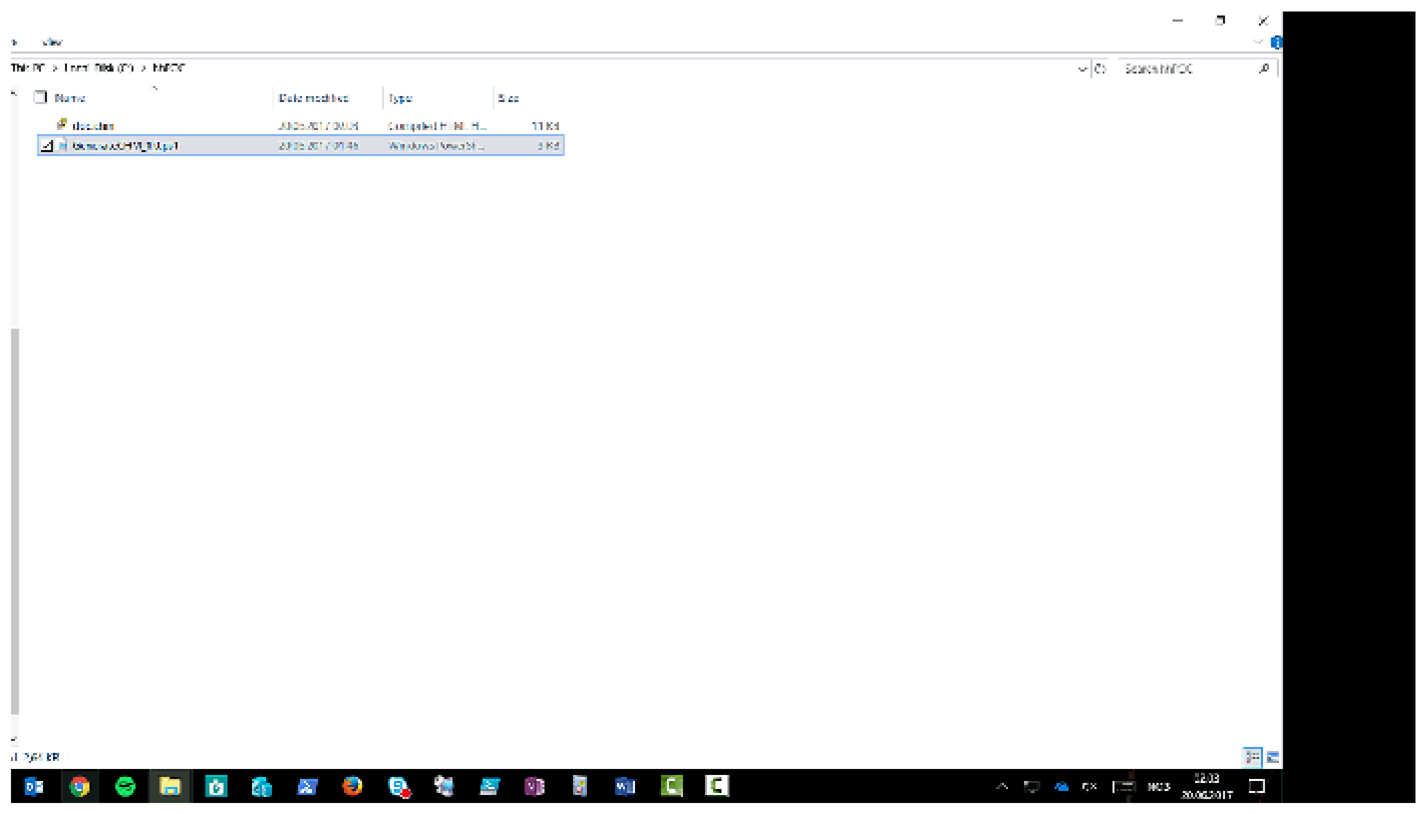
This script generates a simple .CHM that starts Calc.exe through ActiveX. There are certainly far more interesting things you can do.
For instance, running Cn33liz StarFighters to get an Empire agent going: https://github.com/Cn33liz/StarFighters/

The interesting part of my PoC script are these lines:



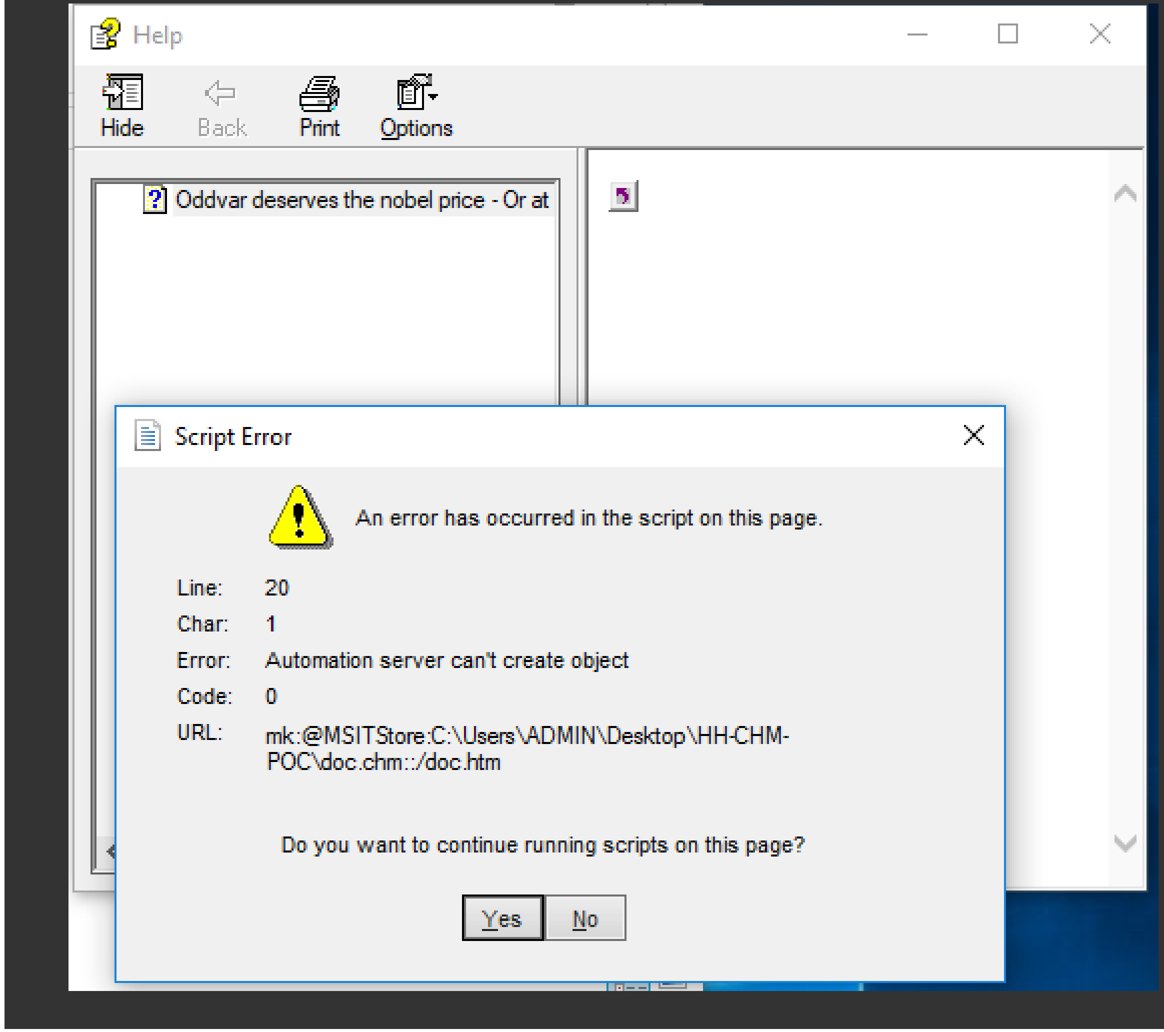
```
<SCRIPT>
alert("I bypassed something! - Hit OK to POP a Calc - Remember to answer yes on the next question");
var shell = new ActiveXObject("WScript.Shell");
shell.run('"calc.exe"');
x.Click();
</SCRIPT>
```

This gif shows you the Device Guard bypass in action:



This issue is fixed in the Windows 10 Creators update v1703 (aka Redstone 2) or if you want to patch it on older versions of Windows 10 you can find the correct patch here: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-8625

After I updated my machine to v1703 of Windows 10 I got this error when I tried to run the custom CHM file:



Script Error

An error has occurred in the script on this page.

```
Line:   20
Char:   1
Error:  Automation server can't create object
Code:   0
URL:    mk:@MSITStore:C:\Users\ADMIN\Desktop\HH-CHM-
        POC\doc.chm::/doc.htm
```

Do you want to continue running scripts on this page?

For you defenders out there, I would also monitor hh.exe and look at what the process is doing on your machines. Could be evil stuff going on.

That's it. Hope this was interesting to read and that it inspired you to conduct your own security research to make Windows even more secure. If you are interested in learning more about Device Guard I suggest reading Microsoft official documentation and some blogposts by Matt Graeber (they have helped me):

https://docs.microsoft.com/en-us/windows/device-security/device-guard/device-guard-deployment-guide

http://www.exploit-monday.com/2016/09/introduction-to-windows-device-guard.html
http://www.exploit-monday.com/2016/09/using-device-guard-to-mitigate-against.html
http://www.exploit-monday.com/2016/10/code-integrity-policy-reference.html
http://www.exploit-monday.com/2016/11/code-integrity-policy-audit-methodology.html
http://www.exploit-monday.com/2016/11/Effectiveness-of-Device-Guard-UMCI.html
http://www.exploit-monday.com/2016/12/updating-device-guard-code-integrity.html

Remember to send your discoveries to secure@microsoft.com and do not use your discoveries for evil. #WhiteHat4Life

SHARE THIS:

Twitter Facebook Google

★ Like

Be the first to like this.

RELATED

Windows Defender Attack Surface Reduction Rules bypass
In "Security"

Bypassing Application Whitelisting with BGInfo
In "Security"

AppLocker – Case study – How insecure is it really? – Part 1
In "Security"

PREVIOUS POST
Ping is okay? – Right?

NEXT POST
Research on CMSTP.exe

3 THOUGHTS ON “BYPASSING DEVICE GUARD UMCI USING CHM – CVE-2017-8625”

Pingback: [Sharing] Analysing simple tricks used in malicious documents | VXSecurity – something strange happens inside it

Pingback: 【技术分享】CVE-2017-8625：使用自定义CHM文件绕过Windows 10的Device Guard - 莹莹之色

Pingback: Cómo saltarse Device Guard en Windows 10 con CVE-2017-8625 - Blog Paginas Web Ciudad Real - Seguridad informática, Diseño web, Páginas web

LEAVE A REPLY

Enter your comment here...

Search ...

SEARCH

POWERED BY WORDPRESS.COM.