UAC Bypass by Mocking Trusted Directories

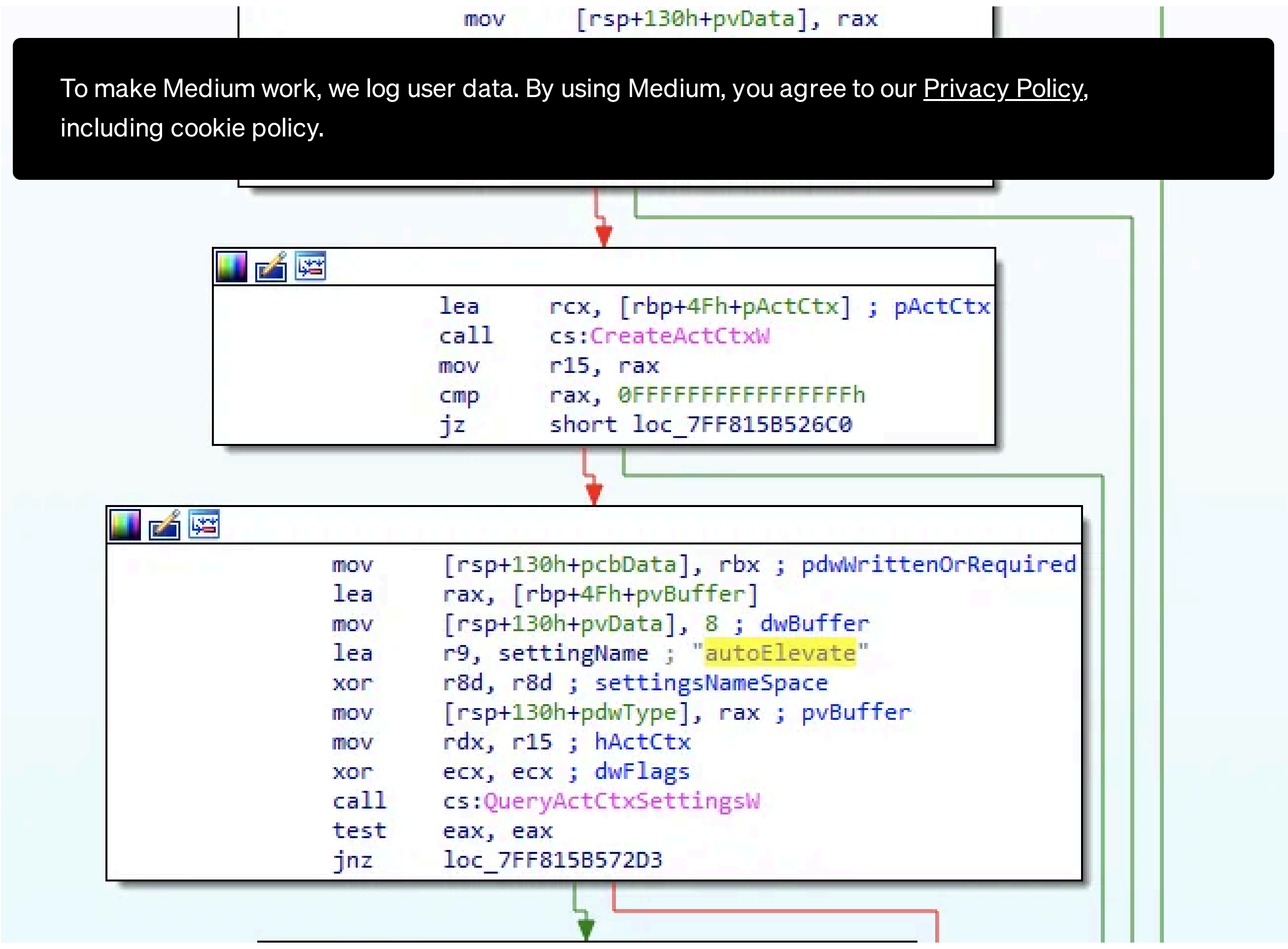
David Wells · Follow
Published in Tenable TechBlog · 6 min read · Nov 9, 2018

Hello Everyone,

During research for some new User Account Control (UAC) bypass techniques, I discovered what I believe to be a new bypass method (at the time of this writing). It is worth mentioning that Microsoft doesn't consider UAC a security boundary, however we still reported the bug to Microsoft and want to share its details here. This method was successfully tested against Windows 10 Build 17134. Before I dive into the details of the bypass, I will first offer a short primer on UAC and its quirks for those unfamiliar with some of it's inner-workings.

UAC Primer

When a user that is part of the Administrators group wants to execute a process that requires elevation, the UAC prompt is presented to confirm process elevation to the user. This UAC prompt however, is not popped for



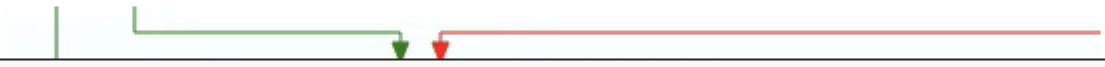
```
mov     [rsp+130h+pvData], rax
```

```
lea     rcx, [rbp+4Fh+pActCtx] ; pActCtx
call    cs:CreateActCtxW
mov     r15, rax
cmp     rax, 0FFFFFFFFFFFFFFFFh
jz      short loc_7FF815B526C0
```

```
mov     [rsp+130h+pcbData], rbx ; pdwWrittenOrRequired
lea     rax, [rbp+4Fh+pvBuffer]
mov     [rsp+130h+pvData], 8 ; dwBuffer
lea     r9, settingName ; "autoElevate"
xor     r8d, r8d ; settingsNameSpace
mov     [rsp+130h+pdwType], rax ; pvBuffer
mov     rdx, r15 ; hActCtx
xor     ecx, ecx ; dwFlags
call    cs:QueryActCtxSettingsW
test    eax, eax
jnz     loc_7FF815B572D3
```

Figure 1 — Reading Executable’s Manifest for potential “autoElevate” key

If found and the value is “True,” it will be considered an auto elevating executable which will be ran elevated and bypass any UAC dialog (provided it passed the next requirements mentioned later). There is one exception to this “autoElevate” rule however. Regardless of manifest, if the file name itself matches one of the whitelisted EXE names, it will also be considered an “auto elevating” executable. Below you’ll see a *bsearch* call after this manifest check to see if the file name exists in a list of whitelisted executable names. If the exe name matches one of these executable names, then auto elevation will be attempted regardless of manifest.



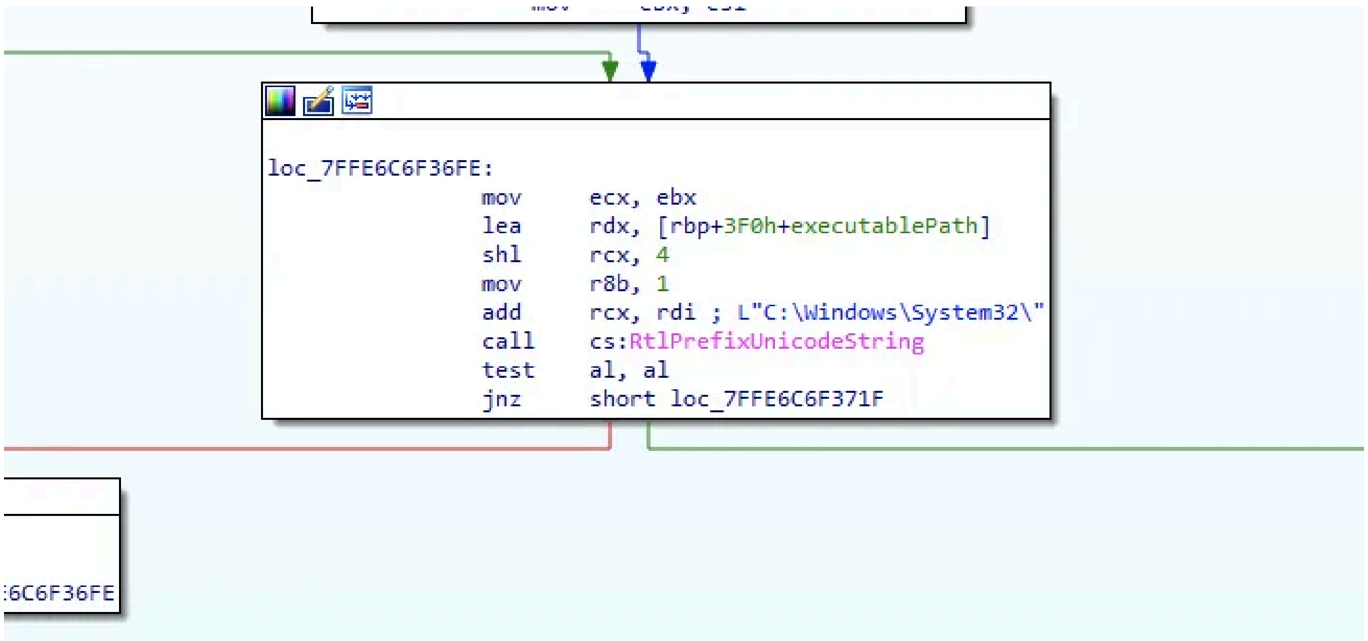
Requirement 2: Properly Signed

As

elevating,” it will now do a signature check using *wintrust!WTGetSignatureInfo*. This means an attacker won’t be able to simply craft their own “autoElevating” manifest or executable file name to get auto elevation to succeed, as the attacker’s binary is most likely not properly signed and it also probably doesn’t pass the last requirement, which is Executing from Trusted Directory.

Requirement 3: Executing from Trusted Directory

The last auto elevating requirement is that the target executable resides in a “trusted directory,” such as “C:\Windows\System32”. Figure 3 shows AIS doing this check on a path requesting elevation, in this case one of the paths its considering “trusted” is “C:\Windows\System32”.



```
loc_7FFE6C6F36FE:
                mov     ecx, ebx
                lea     rdx, [rbp+3F0h+executablePath]
                shl     rcx, 4
                mov     r8b, 1
                add     rcx, rdi ; L"C:\Windows\System32\"
                call    cs:RtlPrefixUnicodeString
                test    al, al
                jnz     short loc_7FFE6C6F371F
```

Figure 3

The name of this write up is “*Bypassing UAC by Mocking Trusted Directories*,” *RtlPrefixUnicodeString* check of course, and I'll also mention that this is so

W

Using the CreateDirectory API however, and prepending a "\\?\" to the directory name I want to create, we can bypass some of these naming filter rules and send the directory creation request directly to file system.

Figure 4

This results in a bit of an awkward directory happily coexisting on the filesystem alongside the real "C:\Windows\" (except for when you try to do anything with it in Windows Explorer).

Figure 5 — Directory deletion requests silently fail and unable to rename directory to remove trailing space.

Now that we have a "C:\Windows \" directory, we can create a "system32" directory in it and copy one of the signed, auto elevating executables from

When this awkward path is sent to AIS for an elevation request, the path is pa

“C

the string that trusted directory checks are performed against (using *RtlPrefixUnicodeString)* for the rest of the routine. The beauty is that after the trusted directory check is done with this converted path string, it is then freed, and rest of checks (and final elevated execution request) are done with the original executable path name (with the trailing space). This allows all other checks to pass and results in appinfo.dll spawning my winSAT.exe copy as auto elevated (since it is both properly signed and whitelisted for auto elevation).

To actually elevate attacker code through this, I simply dropped a fake WINMM.dll (imported by winSAT.exe) in its current directory “C:\Windows\System32\” for a local dll hijack. The full concept can be seen in figure below.

Written by David Wells

737 Followers · Writer for Tenable TechBlog

Follow

More from David Wells and Tenable TechBlog

David Wells in Tenable TechBlog

Bypass Windows 10 User Group Policy (and more) with this One…

I'm going to share an (ab)use of a Windows feature which can result in bypassing User…

Feb 18, 2020 688 6

Clément Notin [Tenable] in Tenable TechBlog

SMB "Access is denied" Caused by Anti-NTLM Relay Protection

Explanations of the "Microsoft network server: Server SPN target name validation…

Jan 11, 2023 24 2

Recommended from Medium

Mark Manson

40 Life Lessons I Know at 40 (That I Wish I Knew at 20)

Today is my 40th birthday.

Sep 23 👏 22K 💬 454

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

✦ Jun 1 👏 25K 💬 484

Lists

General Coding Knowledge
20 stories · 1693 saves

Coding & Development
11 stories · 881 saves

Stories to Help You Grow as a Software Developer
19 stories · 1452 saves

ChatGPT
21 stories · 855 saves

Desiree Peralta in Publishous

OnlyFans is Finally Dead

And I'm happy about it.

✦ Oct 8 17.5K 352

Jonathan Mondaut

How ChatGPT Turned Me into a Hacker

Discover how ChatGPT helped me become a hacker, from gathering resources to tackling...

✦ Jun 18 1.6K 54

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams