



Microsoft Windows - Token Process Trust SID Access Check Bypass Privilege Escalation

EDB-ID: 44630	Author: Google Security Research	Published: 2018-05-16
CVE: CVE-2018-8134	Type: Local	Platform: Windows
Aliases: N/A	Advisory/Source: Link	Tags: Local
E-DB Verified: 	Exploit:  Download / View Raw	Vulnerable App: N/A

[« Previous Exploit](#)

[Next Exploit »](#)

```
1 Windows: Token Trust SID Access Check Bypass EOP
2 Platform: Windows 10 1709 (also tested current build of RS4)
3 Class: Elevation of Privilege
4
5 Summary: A token's trust SID isn't reset when setting a token after process creation allowing a user
6          process to bypass access checks for trust labels.
7
8 Description:
9
10 When a protected process is created it sets the protection inside the EPROCESS structure but also adds a
11    special trust SID to the primary token as part of SeSubProcessToken. Where the process protection is used
    for things such as what access rights to other processes the trust SID is used for direct access checks
    where a security descriptor has a process trust label. A good example is the \KnownDlls object directory
    which is labeled as PPL-WinTcb to prevent tampering from anything not at that protection level.
10 This trust SID isn't cleared during duplication so it's possible for a non-protected process to open the
11    token of a protected process and duplicate it with the trust SID intact. However using that token should
    clear the SID, or at least cap it to the maximum process protection level. However there's a missing edge
```

case, when setting a primary token through NtSetInformationProcess (specifically in PspAssignPrimaryToken). Therefore we can exploit this with the following from a normal non-admin process:

- 1) Create a protected process, werfaultsecure.exe is a good candidate as it'll run PP-WinTcb. It doesn't have to do anything special, just be created.
- 2) Open the process token (we get PROCESS_QUERY_LIMITED_INFORMATION) and duplicate it to a new primary token.
- 3) Create a new suspended process which will run the exploit code with the original token.
- 4) Set the protected process token using NtSetInformationProcess
- 5) Resume exploit process and do something which needs to pass the trust label check.

NOTE: There is also a related issue during impersonation and the call to SetTokenCanImpersonate. Normally the current process trust SID is checked against the impersonation token trust SID and if the process token's is lower a flag is returned to the caller which resets the new token's trust SID to the process one. This check occurs before the check for SeImpersonatePrivilege but after the check for an anonymous token authentication ID. Therefore if you're an admin you could craft a token with the anonymous token authentication ID (but with actual groups) and do a similar trick as with the process token to prevent the reset of the trust SID during impersonation. However I couldn't find an obvious use for this as the trust label seems to be based on the minimum between the impersonation and process token's trust SIDs and when impersonating over a boundary such as in RPC it looks like it gets reset to the process' protection level. But might be worth cleaning this up as well if you're there.

Proof of Concept:

I've provided a PoC as a C# project. It does the previous described trick to run a process which can then set the trust label on a new event object it creates (\BaseNamedObject\PPDEMO). If you run the poc with a command line parameter it will try and do the event creation but should print access denied.

- 1) Compile the C# project. It will need to grab the NtApiDotNet from NuGet to work.
- 2) Run the poc with no parameters as a normal user. It will capture the token and respawn itself to create the event.

Expected Result:

Setting the trust label returns accessdenied.

Observed Result:

The trust label is successfully set.

Proof of Concept:

<https://github.com/offensive-security/exploit-database-bin-splotts/raw/master/bin-splotts/44630.zip>














Related Exploits

Trying to match CVEs (1): CVE-2018-8134

Other Possible E-DB Search Terms: **Microsoft Windows**

Date	D	V	Title	Author
1999-01-07		✓	Microsoft Windows - 'April Fools 2001' Set Incorrect Date	Richard M. ...
2010-07-08		✓	Microsoft Windows - 'cmd.exe' Unicode Buffer Overflow (SEH)	bitform
2005-09-06		✓	Microsoft Windows - 'keybd_event' Local Privilege Escalation	AndrÃ©s Acu...
2005-08-01		✓	Microsoft Windows - 'LegitCheckControl.dll' Genuine Advantage Validation Patch	HaCkZaTaN
2017-06-22		✓	Microsoft Windows - 'win32k!ClientPrinterThunk' Kernel Stack Memory Disclosure	Google Secu...
2014-11-22		🕒	Microsoft Windows - 'win32k.sys' Denial of Service	Kedamsky
2006-02-12		✓	Microsoft Windows - ACLs Privilege Escalation (2)	Andres Tarasco
2012-11-29		✓	Microsoft Windows - AlwaysInstallElevated MSI (Metasploit)	Metasploit
2007-03-10		✓	Microsoft Windows - DCE-RPC svcctl ChangeServiceConfig2A() Memory Corruption	h07
2012-10-16		✓	Microsoft Windows - Escalate Service Permissions Privilege Escalation (Metasploit)	Metasploit
2012-10-10		✓	Microsoft Windows - Escalate UAC Execute RunAs (Metasploit)	Metasploit
2012-10-10		✓	Microsoft Windows - Escalate UAC Protection Bypass (Metasploit)	Metasploit
2017-08-22		✓	Microsoft Windows - Escalate UAC Protection Bypass (Via COM Handler Hijack) (Metasploit)	Metasploit
2016-08-19		🕒	Microsoft Windows - Fileless UAC Protection Bypass Privilege Escalation (Metasploit)	Pablo Gonzá...

Date	D	V	Title	Author
2010-08-21			Microsoft Windows - IcmpSendEcho2Ex Interrupting Denial of Service	I3D
2010-09-07			Microsoft Windows - Local Procedure Call (LPC) Privilege Escalation	yuange
2018-01-08			Microsoft Windows - Local XPS Print Spooler Sandbox Escape	Google Secu...
2013-01-25			Microsoft Windows - Manage Memory Payload Injection (Metasploit)	Metasploit
2006-01-15			Microsoft Windows - Metafile '.WMF' Arbitrary File Download (Generator)	darkeagle
2012-08-15			Microsoft Windows - Service Trusted Path Privilege Escalation (Metasploit)	Metasploit
2014-05-22			Microsoft Windows - Touch Injection API Local Denial of Service	Tavis Ormandy
2018-06-04			Microsoft Windows - UAC Protection Bypass (Via Slui File Handler Hijack) (Metasploit)	Metasploit
2010-08-17			Microsoft Windows - Win32k!xxxRealDrawMenuitem() Missing HBITMAP Bounds Checks	Tavis Ormandy
2005-05-02			Microsoft Windows - WINS Vulnerability + OS/SP Scanner	class101
2015-10-26			Microsoft Windows 10 - 'pcap' Driver Privilege Escalation	Rootkitsmm

© Copyright 2018 Exploit Database