2019: a local hacking odyssey
The MITM attack against password manager

Fujitsu System Integration Laboratories Limited
Security Meister - High Master
Soya Aoyama
My Profile

SOYA Aoyama
Security Researcher @ Fujitsu System Integration Laboratories Ltd
Organizer @ BSides Tokyo

1992 ~ 2015
software developer of Windows.

2015 ~
security researcher
- 2016 AVTOKYO
- 2017 BSides Las Vegas
- 2018 GrrCON / ToorCon / DerbyCon / AVTOKYO
- 2019 HackMiami / LeHack

2018 ~
BSides Tokyo Organizer
- 2018 first BSides in East Asia
What is a Local Hacking?

• Attacks that can be done
  – after breaking into local PC
  – with or without administrator privileges
My research history

2016

Jump the AirGap

2017

Way to escalate privileges to administrator
- BSides Las Vegas

2018

Way to evading the ransomware protection
- GrrCON
- ToorCon
- DerbyCon
- Hack Miami
- LeHack

2019
How to escalate privileges to administrator in latest Windows

- Basic Concept
- Detail
It all started...

When I execute the CompMgmtLauncher
CompMgmtLauncher.exe

– does not display the UAC screen
– executes with administrator privileges
– loads 3\textsuperscript{rd} Party dll (in Program Files Folder)

What processes can access programmer's folder?
– Explorer.exe?
So...

I checked the Explorer
I found a way to get administrator privileges.

Which means...

Administrator Privileges
Basic Concept
Replace the correct dll with malicious one

Pass through the function to the correct dll
What functions the dll has

I used the dumpbin command
The four functions API

Describe on Microsoft web site.

- HRESULT DllCanUnloadNow(void);
- HRESULT DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID *ppv);
- HRESULT DllRegisterServer(void);
- HRESULT DllUnregisterServer(void);

https://docs.microsoft.com/en-us/windows/desktop/api/_com/
So...

Only need to implement four functions
To realize concept
the implementation necessary

1. Load the correct DLL and get its handle
   hModule = LoadLibraryEx(lpLibFileName, hFile, dwFlags);

2. Get address of each function using handle
   Address = GetProcAddress(hModule, lpProcName);

3. When called from EXE, call the corresponding function with the correct arguments
   return Address(arg1, arg2, …);
#include <Shobjidl.h>

typedef HRESULT(__stdcall *CUN)(void);
typedef HRESULT(__stdcall *GCO)(REFCLSID rclsid, REFIID riid, LPVOID *ppv);
typedef HRESULT(__stdcall *RS)(void);
typedef HRESULT(__stdcall *US)(void);
CUN CanUnloadNow;
GCO GetClassObject;
RS RegisterServer;
US UnregisterServer;

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    if (DLL_PROCESS_ATTACH == ul_reason_for_call) {
        WCHAR dll[MAX_PATH + 1] = { 0 };
        GetModuleFileName(hModule, dll, MAX_PATH);
        wcscat(dll, L"_");
        HINSTANCE hDllInstance = ::LoadLibraryEx(dll, NULL, LOAD_WITH_ALTERED_SEARCH_PATH);
        CanUnloadNow = (CUN)GetProcAddress(hDllInstance, "DllCanUnloadNow");
        GetClassObject = (GCO)GetProcAddress(hDllInstance, "DllGetClassObject");
        RegisterServer = (RS)GetProcAddress(hDllInstance, "DllRegisterServer");
        UnregisterServer = (US)GetProcAddress(hDllInstance, "DllUnregisterServer");
    }
    return TRUE;
}

STDAPI DllCanUnloadNow(void) { return CanUnloadNow(); }
STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID *ppv) { return GetClassObject(rclsid, riid, ppv); }
STDAPI DllRegisterServer(void) { return RegisterServer(); }
STDAPI DllUnregisterServer(void) { return UnregisterServer(); }
Demonstration video 1
Mechanism

User executes malicious program
Mechanism

Malicious program replaces correct dll with itself

- **Process**
  - System Integrity Level
  - High Integrity Level
  - Medium Integrity Level
    - Malicious.bat
  - Low Integrity Level

- **File**
  - Any Folder
    - Malicious.bat
  - Malicious.dll
  - %UserProfile\...\OneDrive
    -FileSyncShell64.dll
    -FileSyncShell64.dll_
Mechanism

Explorer loads malicious dll

File

Any Folder
- Malicious.bat
- Malicious.dll

%UserProfile%\...\OneDrive
- FileSyncShell64.dll
- FileSyncShell64.dll_

Process

System Integrity Level

High Integrity Level

Medium Integrity Level
- Explorer.exe
-FileSyncShell64.dll

Low Integrity Level
Mechanism
Malicious program replaces correct dll with itself

<table>
<thead>
<tr>
<th>Process</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Integrity Level</td>
<td>%UserProfile%...\OneDrive</td>
</tr>
<tr>
<td>High Integrity Level</td>
<td>FileSyncShell64.dll</td>
</tr>
<tr>
<td>Medium Integrity Level</td>
<td>FileSyncShell64.dll_</td>
</tr>
<tr>
<td>Low Integrity Level</td>
<td>FileSyncShell64.dll</td>
</tr>
<tr>
<td></td>
<td>c:\Program Files\WinMerge</td>
</tr>
<tr>
<td>Explorer.exe</td>
<td>ShellExtensionX64.dll</td>
</tr>
<tr>
<td></td>
<td>ShellExtensionX64.dll_</td>
</tr>
</tbody>
</table>

UAC bypass
Need administrator privileges

Malicious program replaces correct dll with itself
Mechanism

Malicious program executes CompMgmtLauncher

Process
- System Integrity Level
- High Integrity Level
- Medium Integrity Level: Explorer.exe
- Low Integrity Level

File
- %UserProfile%\System32: CompMgmtLauncher.exe
- c:\windows\system32:FileSyncShell64.dll
- c:\Program Files\WinMerge: ShellExtensionX64.dll

Mechanism
- Malicious program executes CompMgmtLauncher
Malicious program gets administrative privileges

**Mechanism**

UAC bypass

Need administrator privileges

Yay!
Microsoft Bug Bounty Program

Microsoft strongly believes close partnerships with researchers make customers more secure. Security researchers play an integral role in the ecosystem by discovering vulnerabilities missed in the software development process. Each year we partner together to better protect billions of customers worldwide.

The Microsoft Bug Bounty Program is designed to supplement and encourage research in certain technologies to better protect our customers and the broader ecosystem. Through targeted and ongoing bounty programs, we reward researchers for submitting their findings to one of our eligible bounty programs and for partnering with us through Coordinated Vulnerability Disclosure. If you are a security researcher that has found a vulnerability in a Microsoft product, service, or device we want to hear from you. If it is within scope of a bounty program you can receive bounty award according to the program descriptions. Even if it is not covered under an existing bounty program, we will publicly acknowledge your contributions when we fix the vulnerability. Both categories of submission are counted in our annual Top 100 Researcher leaderboard.

Click here to submit a security vulnerability

I submitted a vulnerability report
MSRC said…
Hello,

Thank you for contacting the Microsoft Security Response Center (MSRC). Upon investigation, we have determined that this does not meet the bar for security servicing as binary planting in the application directory would already indicate the user is compromised.

For an in-depth discussion of what constitutes a product vulnerability please see the following:

"Definition of a Security Vulnerability"


Again, we appreciate your report.

Regards,
However…

This issue has been fixed

Microsoft stealthy fixed!!
Attacks that can be done

- after breaking into local PC
- with or without administrator privileges

➡️ The Local Hacking
An Inconvenient Truth: *Evading the Ransomware Protection in Windows 10*

- Basic Concept
- Detail
I noticed ...

There is a defect in Ransomware Protection

- Allowed Apps: Explorer.exe, cmd.exe
- Protected Folders: %UserProfile%\Documents, %UserProfile%\Pictures
- File: c:\windows\system32

Controlled folder access
- Protect files, folders, and memory areas on your device from unauthorized changes by unfriendly applications.
It all started ...

Why does Explorer load OneDrive dll?
I searched in the registry

It was found in HKEY_CLASSES_ROOT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default)</td>
<td>REG_SZ</td>
<td>C:\Users\saoyama\AppData\Local\Microsoft\OneDrive\17.3.6381.0405\amd64\FileSyncShell64.dll</td>
</tr>
<tr>
<td>ThreadingModel</td>
<td>REG_SZ</td>
<td>Apartment</td>
</tr>
</tbody>
</table>
What HKEY_CLASSES_ROOT?

HKCR is that merges HKLM with HKCU

The RegOpenUserClassesRoot function provides a merged view for processes, such as services, that are dealing with clients other than the interactive user. In this case, the HKEY_CLASSES_ROOT key provides a view of the registry that merges the information from HKEY_LOCAL_MACHINE\Software\Classes with the information from HKEY_CURRENT_USER\Software\Classes.

The system uses the following rules to merge information from the two sources:

- The merged view includes all subkeys of the HKEY_CURRENT_USER\Software\Classes key.
- The merged view includes all immediate subkeys of the HKEY_LOCAL_MACHINE\Software\Classes key that do not duplicate the subkeys of HKEY_CURRENT_USER\Software\Classes.
- At the end of this topic is a list of subkeys that are found in both HKEY_LOCAL_MACHINE\Software\Classes and HKEY_CURRENT_USER\Software\Classes. The immediate subkeys of these keys from the HKEY_LOCAL_MACHINE tree are included in the merged view only if they are not duplicates of immediate subkeys from the HKEY_CURRENT_USER tree. The merged view does not include the HKEY_LOCAL_MACHINE contents of duplicate subkeys.

Which dll meets the requirements?

I found it
What functions the dll has

I used the dumpbin command
EXE
- Call LoadLibraryEx()
- Call DllGetClassObject()
- Call Func_A()
- Call Func_B()
- Call Func_C()

Malicious DLL
- DllMain()
- Call LoadLibraryEx()
- DllGetClassObject()
- Call DllGetClassObject()

DLL
- DllMain()
- DllGetClassObject()
- Return ClassObject pointer
- ClassObject
- Func_A()
- Func_B()
- Func_C()

However...
Only need to implement four export functions
Demonstration video 2
Mechanism

User executes malicious program

File

- Malicious. bat
- Malicious.dll

Any Folder

Process

- Low Integrity Level
- Medium Integrity Level
- High Integrity Level
- System Integrity Level

- Malicious. bat
Mechanism
Malicious program writes malicious path to HKCU

Process
- System Integrity Level
- High Integrity Level
- Medium Integrity Level
- Low Integrity Level

Registry
- HKCR: Any Folder\Malicious.dll
- HKCU\Software\Classes: Any Folder\Malicious.dll
- HKLM\Software\Classes: %SysteRoot%\system32\shell32.dll
Mechanism

Explorer loads malicious dll

Registry
- HKCR
  - Any Folder\Malicious.dll
- HKCU\Software\Classes
  - Any Folder\Malicious.dll
- HKLM\Software\Classes
  - %SystemRoot%\system32\shell32.dll

Process
- System Integrity Level
- High Integrity Level
- Medium Integrity Level
  - Explorer.exe
  - Malicious.dll
- Low Integrity Level
Malicious dll encrypts files in Protected Folders

- **Process**
  - System Integrity Level
  - High Integrity Level
  - Medium Integrity Level: Explorer.exe
  - Low Integrity Level

- **File**
  - c:windows\%
  - c:\Program Files
  - c:\Program Files (x86)

- **Protected Folders**
  - %UserProfile%\Documents
  - %UserProfile%\Pictures

Yay!
It’s revenge
I submitted a vulnerability report
MSRC said…
Hello,

Thank you for contacting the Microsoft Security Response Center (MSRC). If I am interpreting your report correctly, this report is predicated on the attacker having login access to the target's account already. This is demonstrated by planting a dll through registry modifications. Since you are only able to write to HKCU, you will not be able to effect other users, just the target you have already compromised through other means. There also does not appear to be an escalation of privileges and you already had the same access level as the target. It would appear the attacker would not gain anything from this attack and could already do anything that the planting could trigger. As your report as written, this does not meet the bar for security servicing.

As such, this thread is being closed and no longer monitored.

If you believe this to be a misunderstanding of the report, submit a new email to secure@microsoft.com without a CRM number in the subject line. Please include:
Microsoft says…

- Local Hacking is not a vulnerability

It does not matter if it's a vulnerability or not.
You Ain’t Seen Nothin’ Yet!
Do you use the password long enough and complex?
A password for the Hawaii emergency agency was hiding in a public photo, written on a Post-it note.

Don't tell me you reuse password!!
Our memory is not so good…

You use a password Manager, right?
There are various products

https://www.pcmag.com/roundup/300318/the-best-password-managers

<table>
<thead>
<tr>
<th>Product</th>
<th>Keeper Password Manager &amp; Digital Vault</th>
<th>Dashlane</th>
<th>Sticky Password Premium</th>
<th>LastPass Premium</th>
<th>Password Boss</th>
<th>LogMeOnce Password Management Suite</th>
<th>RoboForm 8 Everywhere</th>
<th>AgileBits 1Password</th>
<th>True Key by Intel Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoho Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Oh my...

There are exe and dll in Users folder
xxx.dll

Does not have following export functions

- DllCanUnloadNow()
- DllGetClassObject()
- DllRegisterServer()
- DllUnregisterServer()
Can steal the information of Functions
Implementation of all functions is impossible
Need a generic DLL

Does not depend on

- number of export functions
- number of arguments

EXE
- Call LoadLibraryEx()
- Call Func_A()
- Call Func_B()
- Call Func_C()

Malicious DLL
- DllMain()
- DummyFunc()
- Call Func_xxx()

DLL
- DllMain()
- Func_A()
- Func_B()
- Func_C()
Generate export table dynamically

Exe gets address of function via GetProcAddress

Exe gets address of function via GetProcAddress

```
Func();
GetProcAddress(hDllInstance, "Func_C");
Func_C();
```

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>1</td>
</tr>
<tr>
<td>NumberOfFunctions</td>
<td>4</td>
</tr>
<tr>
<td>NumberOfNames</td>
<td>3</td>
</tr>
<tr>
<td>AddressOfFunctions</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNames</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNameOrdinals</td>
<td>&lt;RVA&gt;</td>
</tr>
</tbody>
</table>

```
Func_A()   
Func_B()   
Func_C()
```
Generate export table dynamically

Copy export table from correct dll

**Malicious DLL**

```
<table>
<thead>
<tr>
<th>Func_A</th>
<th>Func_B</th>
<th>Func_C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
```

```
AddressOfFunctions: <RVA>
AddressOfNames: <RVA>
AddressOfNameOrdinals: <RVA>
NumberOfFunctions: 4
NumberOfNames: 3
```

**DLL**

```
<table>
<thead>
<tr>
<th>Func_A</th>
<th>Func_B</th>
<th>Func_C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
```

```
AddressOfFunctions: <RVA>
AddressOfNames: <RVA>
AddressOfNameOrdinals: <RVA>
NumberOfFunctions: 4
NumberOfNames: 3
```

**Copy export table from correct dll**

```
GetProcAddress(hDllInstance, "Func_C");
Func_C();
```

**Malicious DLL**

```
Member Name | Value | Base | NumberOfFunctions | NumberOfNames | AddressOfFunctions | AddressOfNames | AddressOfNameOrdinals |
------------|-------|------|-------------------|---------------|-------------------|---------------|----------------------|
Func_A      |       |      |                   |               |                   |               |                      |
Func_B      |       |      |                   |               |                   |               |                      |
Func_C      |       |      |                   |               |                   |               |                      |
```

```
0xXXXX movzx eax, word ptr[n];
0xXXXX inc ax
0xXXXX mov word ptr[n], ax
0xXXXX movzx eax, word ptr[n];
0xXXXX inc ax
0xXXXX mov word ptr[n], ax
0xXXXX JMP Func[n]
```
Generate export table dynamically

Exe gets incorrect address from malicious dll

**EXE**
- `Func()`
- `GetProcAddress(hDllInstance, "Func_C");`
- `Func_C()`

**Malicious DLL**
- **Member**
  - Name: `<RVA>`
  - Base: 1
  - NumberOfFunctions: 4
  - NumberOfNames: 3
  - AddressOfFunctions: `<RVA>`
  - AddressOfNames: `<RVA>`
  - AddressOfNameOrdinals: `<RVA>`

**DummyFunc()**
- `movzx eax, word ptr[n]`
- `inc ax`
- `mov word ptr[n], ax`
- `movzx eax, word ptr[n]`
- `inc ax`
- `mov word ptr[n], ax`
- `JMP Func[n]`

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0XXXXXX</td>
<td>0XXXXXX</td>
<td>0XXXXXX</td>
</tr>
</tbody>
</table>

**Export Table**

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>1</td>
</tr>
<tr>
<td>NumberOfFunctions</td>
<td>4</td>
</tr>
<tr>
<td>NumberOfNames</td>
<td>3</td>
</tr>
<tr>
<td>AddressOfFunctions</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNames</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNameOrdinals</td>
<td>&lt;RVA&gt;</td>
</tr>
</tbody>
</table>
### Malicious DLL

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>1</td>
</tr>
<tr>
<td>NumberOfFunctions</td>
<td>4</td>
</tr>
<tr>
<td>NumberOfNames</td>
<td>3</td>
</tr>
<tr>
<td>AddressOfFunctions</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNames</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNameOrdinals</td>
<td>&lt;RVA&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>1</td>
</tr>
<tr>
<td>NumberOfFunctions</td>
<td>4</td>
</tr>
<tr>
<td>NumberOfNames</td>
<td>3</td>
</tr>
<tr>
<td>AddressOfFunctions</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNames</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNameOrdinals</td>
<td>&lt;RVA&gt;</td>
</tr>
</tbody>
</table>

### DLL

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>1</td>
</tr>
<tr>
<td>NumberOfFunctions</td>
<td>4</td>
</tr>
<tr>
<td>NumberOfNames</td>
<td>3</td>
</tr>
<tr>
<td>AddressOfFunctions</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNames</td>
<td>&lt;RVA&gt;</td>
</tr>
<tr>
<td>AddressOfNameOrdinals</td>
<td>&lt;RVA&gt;</td>
</tr>
</tbody>
</table>

### Code Snippet

```assembly
movzx eax, word ptr[n]
inc ax
mov word ptr[n], ax
movzx eax, word ptr[n]
inc ax
mov word ptr[n], ax
JMP Func[n]
```

---

**Generate export table dynamically**

### Diagram

- Malicious DLL
- DLL
- Code Snippet

- Functional members:
  - Func_A
  - Func_B
  - Func_C

- RVAs:
  - Func_A
  - Func_B
  - Func_C

- AddressOrdinals:
  - Func_A
  - Func_B
  - Func_C
Demonstration video 3
Mechanism

User executes malicious program

File

Any Folder
- Malicious.bat
- Malicious.dll

Process

System Integrity Level
- Malicious.bat

High Integrity Level
- Malicious.bat

Medium Integrity Level
- Malicious.bat

Low Integrity Level
- Malicious.bat
Mechanism
Malicious program replaces correct dll with itself

<table>
<thead>
<tr>
<th>Process</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Integrity Level</td>
<td>Any Folder</td>
</tr>
<tr>
<td>High Integrity Level</td>
<td>Malicious.bat</td>
</tr>
<tr>
<td>Medium Integrity Level</td>
<td>Malicious.dll</td>
</tr>
<tr>
<td></td>
<td>%UserProfile\1password\app\7</td>
</tr>
<tr>
<td>Low Integrity Level</td>
<td>1password.dll</td>
</tr>
<tr>
<td></td>
<td>1password.dll_</td>
</tr>
</tbody>
</table>
Mechanism

1Password loads malicious dll
Mechanism

The information between exe to dll is plain text
Bug Bounty Program

1Password
1Password is the world's best password manager. Perfect for protecting your business, team, and family.

Points – $5,000 per vulnerability
Up to $100,000 maximum reward
Managed by Bugcrowd

https://bugcrowd.com/agilebits
It’s revenge again
I submitted a vulnerability report
1Pssword said…
Hi SoyaAoyama,

We appreciate your work on this submission. This issue was reported before by other researchers as well. However, after this was reviewed along with 1Password team, it does not qualify for a bounty as it is an attack that depends on a compromise of the user's operating system and environment. That is actually considered as out of scope on this program. As you reported this once on the other program, I am closing it as N/A this time.

Best regards,
- trim_bugcrowd
However, after 3 hours pay the reward

Jeffrey_Goldberg rewarded SoyaAoyama $100
23 Mar 2019 04:36:53 JST

As noted earlier, we can't defend against local compromises, but we are looking at ways to further reduce the risk of such attacks from. And so because your report resulted in actions and potential changes, we are rewarding it.

Jeffrey_Goldberg rewarded SoyaAoyama $400
23 Mar 2019 04:39:08 JST

Although local attacks are out of scope, I wanted to add a bonus as we are more systematically reviewing what practically can be done.
1Password releases beta version

7.3.701.BETA (build #701) – released 2019-07-05 – download

NEW

- Added support for an upcoming feature with 1Password memberships. Stay tuned for more details. {OPW-3904}

IMPROVEMENTS

- Added temporary support for Opera 60 stable version with an expired key within a valid certificate. {OPW-4001}
- 1Password will notify if an attempt to run 1Password with administrator rights is made. Instead, run 1Password normally and it'll request it when needed. {OPW-3775}

SECURITY

- 1Password will now require administrator rights to install and to update. Thanks to @SoyaAoyama for his reports. {OPW-3959, OPW-3887}
- Inform Windows to limit our DLL search to the 1Password's app directory only and not look for it elsewhere in the default list of known locations. Thanks to @SoyaAoyama for his reports. {OPW-3833}
- Added 1Password.brain.exe to be opt'ed out of Windows Error Reporting. {OPW-3804}
- Removed the --database-path support from 1Password.exe as it could be abused to redirect 1Password to an unexpected location. We recommend using Group Policy to set the database path instead. Thanks @zemnmez! {OPW-3776, OPW-3778}
I just want to say one word!!

Only $500?
security impact

- Does not require administrator privileges
- Remote impersonation is possible

Used by over 40,000 businesses with hundreds of glowing reviews
Why is local hacking neglected?

• Because, you got already hacked.

Are there nothing to consider after such an invasion?
In Local

- A lot of important data
- Many things attackers can do without becoming administrator

You would take local hacking after intrusion into account
@SoyaAoyama

https://www.facebook.com/soya.aoyama.3

https://www.slideshare.net/SoyaAoyama