Process Doppelgänging

Franck Jeannot

Montréal, Canada, Août 2018, R532, v1.1

Abstract

A review of the code injection technique called **Process Doppelgänging** that is getting more and more attention for its usage in MS Windows malwares and by its ability to launch processes thats looks legitimate.

Keywords: Doppelgänging, code injection, transactional NTFS, Ensilo, MS Windows, SynAck, ransomware, BlackHat, Kaspersky, AtomBombing

1. Introduction

La technique de ***Process Doppelgänging*** ou « *technique du process sosie*[1] » est une technique d'**injection de code** et d'**évasion** de mécanismes de sécurité (antivirus, ... etc) similaire à la technique de ***Process Hollowing***[2] mais qui utilise **les mécanismes de transactions NTFS** [1]. Cette technique est plus largement documentée depuis la présentation[3] de sa découverte à la conférence Blackhat Europe de Décembre 2017 [2] par des chercheurs d'**Ensilo** [3]. C'est donc une technique utilisée notamment dans les malwares, ransomwares et assimilés qui vise les plateformes Windows, de part l'usage de mécanismes NTFS ("*transactional NTFS*"). En mai 2018 un article[4] de Microsoft recommandait désormais de **ne plus utiliser les transactions NTFS** mais *des solutions alternatives* [4]... "*Microsoft strongly recommends developers utilize alternative means to achieve your application s needs. Many scenarios that TxF was developed for can be achieved through simpler and more readily available techniques. Furthermore, TxF may not be available in future versions of Microsoft Windows. For more information, and alternatives to TxF, please see Alternatives to using Transactional NTFS.*]"

1. https://fr.wiktionary.org/wiki/Doppelg%C3%A4nger
2. https://www.bleepingcomputer.com/news/security/-process-doppelg-nging-attack-works-on-all-windows-versions/
3. https://youtu.be/Cch8dvp836w
4. https://docs.microsoft.com/en-us/windows/desktop/FileIO/about-transactional-ntfs

2. Historique

Les premières identifications de l'usage de cette technique ont été faites au travers de l'analyse de variantes du ransomware **SynAck** [5] en Août/Septembre 2017 [6]. Bleepingcomputer remontait dans leur article du 5 Septembre 2017 des exemples de messages du ransomware [7] montrant l'usage de chiffrement fort avec des algorithmes de type [ecc-secp192r1 aes-ecb-256]. Suite à la publication de Ensilo à la Blackhat, de nouvelles implémentations se sont développées. Le 7 mai 2018 une publication [8] a été faite par des chercheurs de Kaspersky [9] [10] mettant en avant la complexité et l'avancement de dernières variantes du ransomware SynAck et notamment l'usage de la technique de **Process Doppelgänging** dans une nouvelle variante de ce ransomware.

3. Transactions NTFS

Les mécanismes de transaction NTFS existent depuis Windows Vista. Dans l'exemple plus bas en C (Source codeguru.com [5]) on initie la création de transaction via une fonction de **Kernel Transaction Manager (KTM)** :

```
#include <ktmw32.h>
#pragma comment(lib, "KtmW32.lib")
HANDLE hTrans = CreateTransaction(NULL,0, 0, 0, 0, NULL, _T("My NTFS
    Transaction"));
if (hTrans == INVALID_HANDLE_VALUE)
{
cerr << "CreateTransaction failed" << endl;
return 1;
}
```

Listing 1: Exemple d'initialisation de CreateTransaction

5. https://malwareless.com/how-to-remove-synack-ransomware-detailed-removal-guide/
6. https://www.bleepingcomputer.com/news/security/synack-ransomware-sees-huge-spike-in-activity/
7. https://www.bleepstatic.com/images/news/u/986406/Ransomware/SynAck/SynAck-ransom-3.png
8. https://usa.kaspersky.com/about/press-releases/2018_synack-doppelganging
9. https://securelist.com/synack-targeted-ransomware-uses-the-doppelganging-technique/85431/
10. https://threatpost.com/variant-of-synack-malware-adopts-doppelganging-technique/131760/

On peut utiliser la fonction `CreateFileTransacted` comme exemple :

```
USHORT view = 0xFFFE; // TXFS_MINIVERSION_DEFAULT_VIEW
HANDLE hFile = CreateFileTransacted(_T("test.file"), GENERIC_WRITE, 0,
    NULL, CREATE_ALWAYS, 0, NULL, hTrans, &view, NULL);
if (hFile == INVALID_HANDLE_VALUE)
{
     cerr << "CreateFileTransacted failed" << endl;
     return 1;
}
```

Listing 2: Exemple d'utilisation de CreateTransaction

Après que le fichier ait été créé dans le contexte d'une transaction on peut commencer à écrire un fichier. L'exemple suivant [5] écrit une simple chaîne dans un fichier et le ferme :

```
DWORD dwWritten = 0;
string str = "Test String";
WriteFile(hFile, str.c_str(), str.length(), &dwWritten, NULL);
CloseHandle(hFile);
```

Listing 3: Exemple d'écriture de fichier

4. Technique Doppelgänging

Le but principal de la technique de Doppelgänging est d'utiliser les transactions NTFS pour lancer un process malicieux depuis le fichier en transit de telle manière que le processus semble légitime.

4.1. Les 4 grandes étapes de cette technique

- Transaction - Créé une transaction TxF en utilisant un exécutable légitime et réécrit le fichier avec du code malicieux. Ces changements vont être isolés et seulement visibles dans le contexte de la transaction
- Chargement (load) - Création d'une section partagée dans la mémoire et charge l'exécutable malicieux
- Rollback - Fait un retour arrière des modifications en enlevant le code malicieux du système de fichiers
- Anime - Créé un process depuis la section modifiée de mémore et initie l'exécution

4.2. Implémentation C++ / MFC de hasherezade

Voire l'implémentation [11] **complète** mais dite expérimentale (C++ et Visual Studio MFC) de **hasherezade** [12] et son excellent article technique explicatif [13].

4.3. Autres implémentations C/C++

D'autres implementations se trouvent sur Github mais avec divers problèmes de codage, elles restent d'intérêt pour la compréhension de la technique. [14] [15] [16] [17]

4.4. Analyses de binaires compilés utilisant cette technique

Exemple : [18]. Voir un binaire similaire correspondant uploadé sur reverse.it [19], on peut y voir les Warnings sur les API suspicieuses.

4.5. Contre-mesures

hasherezade [6], commentait en Décembre 2017 sur les possibilités de protection par diverses **comparaisons** "*Although this technique may look dangerous, it can be easily detected with the help of any tool that compares if the image loaded in the memory matches the corresponding file on the disk*".

Au 11 Mai 2018, McAfee annonçait disposer de solutions de protections (ENS 10.5.4, released [20] April 24, 2018) [21] et confirmait que des protections avaient été mises en place par Microsoft depuis la version Windows 10 Spring Creators Update (Redstone 4)(RS4)(1803)(April, 30, 2018).

---

11. https://github.com/hasherezade/process_doppelganging
12. https://twitter.com/hasherezade
13. https://hshrzd.wordpress.com/2017/12/18/process-doppelganging-a-new-way-to-impersonate-a-process/
14. https://sivapremv.wordpress.com/2018/02/01/process-doppelganging/
15. https://gist.github.com/hfiref0x/a9911a0b70b473281c9da5daea9a177f
16. https://github.com/3gstudent/Inject-dll-by-Process-Doppelganging
17. https://raw.githubusercontent.com/hfiref0x/UACME/master/Source/Shared/ntos.h
18. http://kali-km.tistory.com/entry/Process-Doppelganging-1?category=490391
19. https://www.reverse.it/sample/b82aebc12360493788d54af88f5d4c5b749364b9b67d110bbd9cd443da234238?environmentId=100
20. https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/27000/PD27598/en_US/ens_1054_rn_ePOCloud_0-00_en-us.pdf
21. https://securingtomorrow.mcafee.com/mcafee-labs/mcafee-protects-against-doppelganging-technique/

4.6. Tests : procmon, proc_ doppel.exe, PE-Sieve, mimikatz

4.6.1. Compilation de proc_ doppel.exe

Compilation dans Visual studio

```
cmake_minimum_required (VERSION 2.8)
project (proc_doppel)
set(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} /MT")
set (srcs
main.cpp
util.cpp
ntdll_undoc.cpp
pe_hdrs_helper.cpp
)
set (hdrs
util.h
ntdll_undoc.h
ntdll_types.h
ntddk.h
pe_hdrs_helper.h
)
add_executable (proc_doppel ${hdrs} ${srcs})
```

Listing 4: Exemple du cmake (CMakeLists.txt) de hasherezade utilisé pour la compilation du binaire d'exemple d'implémentation appelé proc_doppel.exe

On réalise alors la compilation du binaire de test nommé proc_doppel.exe :

```
cmd.exe /C "cd . && "C:\Program Files (x86)\Microsoft Visual Studio
    \2017\Community\Common7\IDE\CommonExtensions\Microsoft\CMake\CMake\
    bin\cmake.exe" -E vs_link_exe --intdir=CMakeFiles\proc_doppel.dir --
    manifests  -- C:\PROGRA~2\MIB055~1\2017\COMMUN~1\VC\Tools\MSVC
    \1414~1.264\bin\Hostx86\x86\link.exe /nologo CMakeFiles\proc_doppel.
    dir\main.cpp.obj CMakeFiles\proc_doppel.dir\util.cpp.obj CMakeFiles\
    proc_doppel.dir\ntdll_undoc.cpp.obj CMakeFiles\proc_doppel.dir\
    pe_hdrs_helper.cpp.obj  /out:proc_doppel.exe /implib:proc_doppel.lib
     /pdb:proc_doppel.pdb /version:0.0  /machine:X86 /debug /INCREMENTAL
     /subsystem:console  kernel32.lib user32.lib gdi32.lib winspool.lib
    shell32.lib ole32.lib oleaut32.lib uuid.lib comdlg32.lib advapi32.
    lib && cd ."
Build succeeded.
```

Listing 5: Exemple de la compilation de proc_doppel.exe

4.6.2. Compilation de Mimikatz

On compile Mimikatz [22] (Version 2.1.1.0). La compilation du binaire devrait déclencher tout Antivirus heuristique installé..., accepter l'exception....

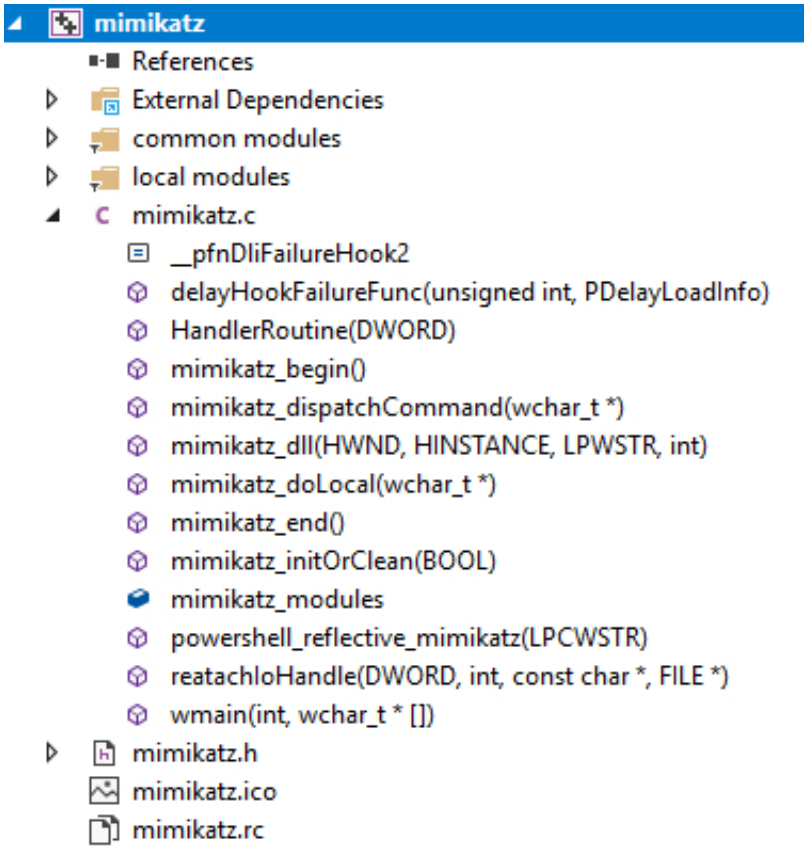

FIGURE *(1): On compile Mimikatz V 2.1.1.0 dans Visual Studio...*

22. https://github.com/gentilkiwi/mimikatz

4.6.3. Tests des binaires

On exécute alors proc_doppel.exe avec mimikatz et un fichier texte en argument...

```
>proc_doppel.exe mimikatz32.exe c:\test\test.txt
```

Listing 6: Execution de proc_doppel.exe : on passe en argument un fichier texte de test

L'outil **PE-Sieve**[23], développé par *hasherezade* permet de scanner un process donné et de chercher des modules chargés manuellement ou modifiés "*Detects inline hooks and other in-memory PE modifications*". Quand l'outil en détecte, il dumpe les PE suspects et fournit un rapport en correspondance au format JSON.

4.6.3.1. Antivirus. Si proc_doppel.exe n'est pas détecté, a contrario, **mimikatz**, étant un binaire connu, est détecté par des Antivirus à jour comme "binaire suspicieux".

4.6.3.2. Windows 7. A l'exécution de la commande, du point de vue du moniteur de processus (**Procmon**) n'est alors visible qu'un **bénigne process du nom du fichier texte**, **l'injection réussit de manière transparente** !

4.6.3.3. Windows 10 - Avant Red Stone 3. L'exécution génère un BSOD (Blue Screen of Death) (dans la mesure de présence de l'ensemble des DLL VStudio pour ce cas précis) sur **NtCreateProcessEx** en raison d'un bug Microsoft (ptr dereference)[24]. Il faut noter aussi que la compilation de l'implémentation de proc_doppel.exe se base sur les DLL de Visual Studio. La simple exécution de proc_doppel.exe sur un PC « standard » sans installation de Visual Studio (exemple Windows 10 Redstone 2 (RS2) 15063.674) et des DLL associées va échouer avec une erreur de type "*The code execution cannot proceed because MSVCP14D.dll was not found...* " puis "*The application was unable to start correctly(0xc000007b). Click OK to close the application.*"

4.6.3.4. Windows 10 Red Stone 4 et supérieur. Il faut noter qu'avec une Version Windows 10 RS4 et +, le test va échouer comme ci-dessous (du fait d'un patch de Microsoft en Win 10 RS4 1803)...

```

>proc_doppel.exe mimikatz.exe C:\test\test.txt
NtCreateProcessEx failed
[-] Failed!
```

Listing 7: Depuis Windows 10 RS4, la tentative échoue

23. https://github.com/hasherezade/pe-sieve/releases
24. https://gist.github.com/hfiref0x/a9911a0b70b473281c9da5daea9a177f

Références

[1] Microsoft, About Transactional NTFS.
URL https://docs.microsoft.com/en-us/windows/desktop/FileIO/about-transactional-ntfs

[2] BlackHat, Black Hat Europe 2017 - Twitter feed.
URL https://twitter.com/hashtag/BHEU2017?src=hash

[3] Tal Liberman and Eugene Kogan, Lost in Transaction: Process Doppelgänging, BlackHat.
URL https://www.blackhat.com/docs/eu-17/materials/eu-17-Liberman-Lost-In-Transaction-Process-Doppelganging.pdf

[4] Microsoft, Alternatives to using Transactional NTFS.
URL https://docs.microsoft.com/en-us/windows/desktop/FileIO/deprecation-of-txf

[5] Marc Grégoire, Introducing the Windows Kernel Transaction Manager, Transactional NTFS and Transactional Registry, codeguru.
URL https://www.codeguru.com/cpp/article.php/c18309/Introducing-the-Windows-Kernel-Transaction-Manager-Transactional-NTFS-and-Transactional-Registry.htm

[6] hasherezade, Process Doppelgänging – a new way to impersonate a process.
URL https://hshrzd.wordpress.com/2017/12/18/process-doppelganging-a-new-way-to-impersonate-a-process/