# SSE - Sécurité pour les applications

# TP5

Master 2 Cybersécurité

2018 - 2019

Encadré par :
Josselin MARIETTE

Réalisé par :
Manon DEROCLES
Alexis LE MASLE

# Table Des Matières

# ASLR

## Linux

| | | demo_sansPIE | demo_avecPIE |
|---|---|---|---|
| 1er lancement | Section exécutable de demo_nnnnPIE | 00400000-00401000 | 5599fd7c8000-5599fd7c9000 |
| | Section exécutable de la libc (libc-2.24.so) | 7f26a5c43000-7f26a5dd8000 | 7fedec050000-7fedec1e5000 |
| | [heap] | 02245000-0266000 | 5599ff5f8000-5599ff619000 |
| | [stack] | 7fff264f6000-7fff26517000 | 7ffcbf29b000-7ffcbf2c000 |
| 2è lancement | Section exécutable de demo_nnnnPIE | 00400000-00401000 | 55bb1e23b000-55bb1e23c000 |
| | Section exécutable de la libc (libc-2.24.so) | 7ff5bbd81000-7ff5bbf16000 | 7f4d0b57c000-7f4d0b711000 |
| | [heap] | 00b3d000-00b5e000 | 55bb1f076000-55bb1f097000 |
| | [stack] | 7ffe8ceceb000-7ffe8ed0c000 | 7ffe3a884000-7ffe3a8a5000 |
| Après redémarrage du système | Section exécutable de demo_nnnnPIE | 00400000-00401000 | 55f25dc7b000-55f25dc7c000 |
| | Section exécutable de la libc (libc-2.24.so) | 7facfc5b9000-7facfc74e000 | 7f52bfe8000-7f52bff9d000 |
| | [heap] | 00816000-00837000 | 55f25e9aa000-55f25e9cb000 |
| | [stack] | 7ffd23717000-7ffd23738000 | 7ffe9904f000-7ffe9907000 |

Tableau d'adresses des zones mémoire des exécutables demo_xxxxPIE

**Les 2 binaires se comportent-ils de la même manière au niveau des adresses de chargement du programme en lui-même, des bibliothèques qu'ils utilisent, de leur pile et de leur tas ? Quel est l'effet d'un redémarrage sur ces adresses ?**

Pour demo_sansPIE:
Après deux exécutions, l'adresse d'exécution du programme lui-même n'a pas changé, les bibliothèques, leur pile et leur tas eux ont changés. Lors du redémarrage du système, les adresses d'exécutions du programme lui-même ne changent pas non plus.

Pour demo_avecPIE:
Après les deux exécutions les adresses ont changé, pour toutes les parties ainsi qu'après le redémarrage.

**Les zones de mémoires affectées au tas et à la pile sont-elles exécutables ? Quel principe est appliqué ici ?**

Les zones affectées au tas et à la pile ne sont pas exécutables, il y'a les droits "rw-p" sur la pile et le tas, le principe du "Write xor Exec" est ici appliqué.

# Windows

| | | DemoASLR1 | DemoASLR2 |
|---|---|---|---|
| 1er lancement | DemoASLRn.exe | 0x7FF7E8CA0000 | 0x140000000 |
| | Kernel32.dll | 0x7FFE936B0000 | 0x7FFE936B0000 |
| | Ntdll.dll | 0x7FFE95BA0000 | 0x7FFE95BA0000 |
| 2è lancement | DemoASLRn.exe | 0x7FF7E8CA0000 | 0x140000000 |
| | Kernel32.dll | 0x7FFE936B0000 | 0x7FFE936B0000 |
| | Ntdll.dll | 0x7FFE95BA0000 | 0x7FFE95BA0000 |
| Après redémarrage du système | DemoASLRn.exe | 0x7FF7E28B0000 | 0x140000000 |
| | Kernel32.dll | 0x7FFFE2D10000 | 0x7FFFE2D10000 |
| | Ntdll.dll | 0x7FFFE3BC0000 | 0x7FFFE3BC0000 |

Tableau des différentes adresses mémoire

**Les 2 binaires se comportent-ils de la même manière au niveau des adresses de chargement du programme et des bibliothèques qu'ils utilisent ? Quel est l'effet d'un redémarrage sur ces adresses ?**

Malgré l'activation de l'ASLR, entre deux lancements des binaires nous obtenons les mêmes adresses. Or avec l'ASLR, nous devons avoir les adresses changeante pour les binaires ainsi que leurs dépendances. À chaque démarrage de ces binaires les adresses doivent être prisent aléatoirement.

Lorsque nous démarrons le système, les adresses des bibliothèques changent, ainsi que l'adresses du binaire DemoASLR1.

**Récupérer l'application PE Studio et analyser les 2 binaires. Quelle information vous permet d'expliquer le comportement observé ?**

| | |
|---|---|
| processor-32bit | false |
| relocation-stripped | false |
| large-address-aware | **true** |
| uniprocessor-only | false |
| system-image | false |
| dynamic-link-library | false |
| executable | **true** |
| debug information stripped | false |
| if on a removable media, copy and run from the swap | false |
| if on a Network, copy and run from the swap | false |

| | |
|---|---|
| NumberOfRvaAndSizes | 16 |
| address-space-layout-randomization (ASLR) | **true** |
| Code Integrity | false |
| data-execution-prevention (DEP) | **true** |
| Image Isolation | **true** |
| structured-exception-handling (SEH) | **true** |
| Image Bound | **true** |
| windows-driver-model (WDM) | false |
| terminal-server-aware | **true** |
| control-flow-guard (CFG) | false |

Informations sur DemoASLR1

Les adresses des bibliothèques ne sont pas dynamique. C'est à dire que les adresses des dépendances sont statique nous avons donc la même adresse pendant toutes l'exécution du système. Lors du redémarrage du système les adresses des bibliothèques sont changées.

| | |
|---|---|
| processor-32bit | false |
| relocation-stripped | **true** |
| large-address-aware | **true** |
| uniprocessor-only | false |
| system-image | false |
| dynamic-link-library | false |
| executable | **true** |
| debug information stripped | false |
| if on a removable media, copy and run from the swap | false |
| if on a Network, copy and run from the swap | false |

| | |
|---|---|
| LoaderFlags | 0x00000000 |
| NumberOfRvaAndSizes | 16 |
| address-space-layout-randomization (ASLR) | false |
| Code Integrity | false |
| data-execution-prevention (DEP) | **true** |
| Image Isolation | **true** |
| structured-exception-handling (SEH) | **true** |
| Image Bound | **true** |
| windows-driver-model (WDM) | false |
| terminal-server-aware | **true** |
| control-flow-guard (CFG) | false |

Informations sur DemoASLR2

ASLR est à “false”, c’est pour cela que l’adresse du binaire ne change pas malgré le redémarrage du système.

Windows gère les ASLR uniquement lors du redémarrage du système. C’est pour cela que l’adresses du binaire de la DemoASLR1 ne change pas alors que nous relançons les programmes.

**Le fonctionnement du mécanisme ASLR est-il identique sur les deux systèmes d'exploitation considérés ? Qu'en pensez-vous ?**

Non, le fonctionnement du mécanisme ASLR n’est pas la même entre Linux et Windows.
Linux, lorsque l’ASLR est activé, gère aléatoirement les adresses des binaires ou des bibliothèque à chaque lancement d'application.
Windows, lui, génère les adresses aléatoire uniquement lors du démarrage du système.

## Mandatory ASLR (Windows)


**Le comportement du programme DemoASLR2 est-il différent ? Qu'en pensez-vous ?**

Non, le comportement du programme DemoASLR2 reste inchangé. En effet la configuration du binaire prime sur la configuration de Windows. Nous le savons puisque en forçant l'ASLR pour le fichier, nous n'obtenons toujours pas un changement d'adresse.

# Contrôle applicatif sous Windows

```
PS C:\Windows\system32> Set-AppLockerPolicy -xml C:\tp\tp-applocker.xml
PS C:\Windows\system32> secpol.msc
PS C:\Windows\system32> Test-AppLockerPolicy -xml C:\tp\tp-applocker.xml -Path C:\Windows\notepad.exe

FilePath               PolicyDecision MatchingRule
--------               -------------- ------------
C:\Windows\notepad.exe        Allowed (Default Rule) All files located in the Windows folder


PS C:\Windows\system32> Test-AppLockerPolicy -xml C:\tp\tp-applocker.xml -Path C:\tp\mimikatz\mimikatz.exe

FilePath                    PolicyDecision MatchingRule
--------                    -------------- ------------
C:\tp\mimikatz\mimikatz.exe DeniedByDefault
```

Tests d'autorisation d'application selon des règles prédéfinies

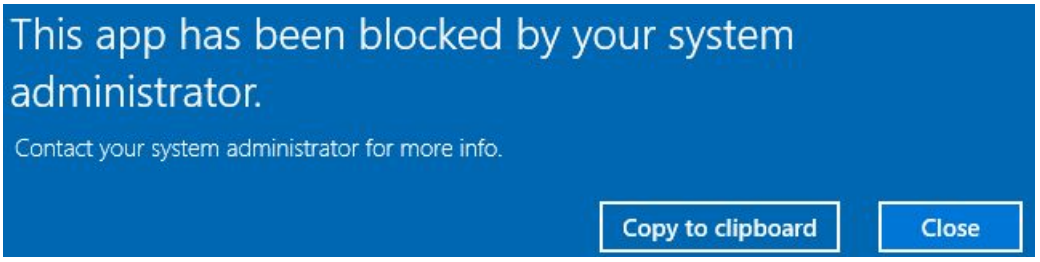
Avec la commande Test-AppLockerPolicy, nous testons si un programme donné peut s'exécuter ou non. Nous pouvons observer que notepad est autorisé par les règles données par tp-applocker.xml et que mimikatz est refusé.

```
PS C:\Windows\system32> sc.exe config AppIdSvc start=auto
[SC] ChangeServiceConfig SUCCESS
PS C:\Windows\system32> sc.exe start AppIdSvc

SERVICE_NAME: AppIdSvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1712
        FLAGS              :
```

**Pouvez-vous lancer les applications Notepad et Mimikatz ciblée précédemment avec la cmdlet Test-AppLockerPolicy ? Est-ce conforme la stratégie déployée ?**

Avec un utilisateur non privilégié nous pouvons lancer l'application Notepad. Nous ne pouvons pas exécuter l'application mimikatz.




Message d'erreur lors du lancement de l'application Mimikatz

**Avec une invite de commande non-élevée, taper les commandes suivantes :**

```
cp C:\tp\mimikatz.exe C:\Windows\temp
C:\Windows\temp\mimikatz.exe
```

**Que se passe-t-il ? Comment peut-on expliquer ce comportement.**

```
PS C:\Users\util_std> cp C:\tp\mimikatz\mimikatz.exe C:\Windows\temp
PS C:\Users\util_std> C:\Windows\temp\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Aug 13 2017 17:27:53
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 21 modules * * */

mimikatz #
```

Lancement de l'application mimikatz

Nous pouvons lancer l'application mimikatz à la suite de ces deux commandes.

Lorsque nous avons créé la règle qui interdit le lancement de mimikatz nous l'avons interdit sur l'application a un chemin donnée. En changeant de dossier et le mettre dans le dossier temporaire du système nous pouvons avoir accès à l'application.


Nouvelle règle pour mimikatz.exe

**Tester le lancement de Mimikatz à partir des différents répertoires où il a été copié. La règle est-elle efficace ?**

```
PS C:\Users\IEUser> C:\Windows\temp\mimikatz.exe
Program 'mimikatz.exe' failed to run: This program is blocked by group policy. For more information, contact your
system administratorAt line:1 char:1
+ C:\Windows\temp\mimikatz.exe
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~.
At line:1 char:1
+ C:\Windows\temp\mimikatz.exe
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
```

Refus du système de lancé l'application mimikatz

Avec la nouvelle règle créée nous pouvons plus lancé l'application, ni dans son dossier d'origine, ni dans le dossier temp de Windows.

**Les informations apportés peuvent-elles être utiles du point de vue de la sécurité ?**

```
PS C:\tp\mimikatz> Get-AppLockerFileInformation -EventLog -EventType Denied -Statistics


FilePath       : %OSDRIVE%\BGINFO\BGINFO.EXE
FilePublisher  : O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\BGINFO\BGINFO.EXE,4.20.0.0
FileHash       : SHA256 0xA78AE13C9E5B8AADD3FB97322A0CBECDF50B123E23CF8DA39362D724F5F320C2
PolicyDecision : Denied
Counter        : 2

FilePath       : %OSDRIVE%\USERS\UTIL_STD\APPDATA\LOCAL\MICROSOFT\ONEDRIVE\17.3.6816.0313\FILESYNCCONFIG.EXE
FilePublisher  : O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\MICROSOFT
                 ONEDRIVE\FILESYNCCONFIG.EXE,17.3.6816.313
FileHash       : SHA256 0x7BE00D40A19ACF6476DA2D723968CD1564DC0A2EA3A03A529FCFFAAFB4D0BCCB
PolicyDecision : Denied
Counter        : 1

FilePath       : %OSDRIVE%\USERS\UTIL_STD\APPDATA\LOCAL\MICROSOFT\ONEDRIVE\17.3.6816.0313_1\FILESYNCCONFIG.EXE
FilePublisher  : O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US\MICROSOFT
                 ONEDRIVE\FILESYNCCONFIG.EXE,17.3.6816.313
FileHash       : SHA256 0x7BE00D40A19ACF6476DA2D723968CD1564DC0A2EA3A03A529FCFFAAFB4D0BCCB
PolicyDecision : Denied
Counter        : 1

FilePath       : %REMOVABLE%\WINDOWS\PASSWORDFILTERSERVICE.EXE
FilePublisher  :
FileHash       : SHA256 0x241BC559CFEA4070EA49208B686C23A086FFA35D9C29E82AA87C776F14C114E8
PolicyDecision : Denied
Counter        : 1

FilePath       : %OSDRIVE%\TP\MIMIKATZ\MIMIKATZ.EXE
FilePublisher  :
FileHash       : SHA256 0x02C86C9977C85A08F18AC1DAE02F1CDDA569EABA51EC6D17AED6F4EBC2ADAF21
PolicyDecision : Denied
Counter        : 6

FilePath       : %WINDIR%\TEMP\MIMIKATZ.EXE
FilePublisher  :
FileHash       : SHA256 0x02C86C9977C85A08F18AC1DAE02F1CDDA569EABA51EC6D17AED6F4EBC2ADAF21
PolicyDecision : Denied
Counter        : 3
```

Fichier d'information sur AppLocker

Les informations données peuvent être utilisées car nous pouvons voir comme information le nombre de tentative de connexion sur une application interdite comme mimikatz. Nous pouvons voir également sur quelle chemin nous avons essayé d'accéder à l'application.

**Cette méthode consistant à ajouter des règles de refus ciblant des condensats cryptographiques de fichiers vous paraît-elle efficace, notamment vis-à-vis d'une règle de chemin ?**

Devoir ajouter des règles vis à vis des chemins n'est pas sur. Il suffit de copier l'application pour l'exécuter. Pour sécuriser il faut couvrir tous les chemins potentiellement. Cette méthode peut laisser place à un oubli (erreur humaine) de la part de l'administrateur de sécurité.