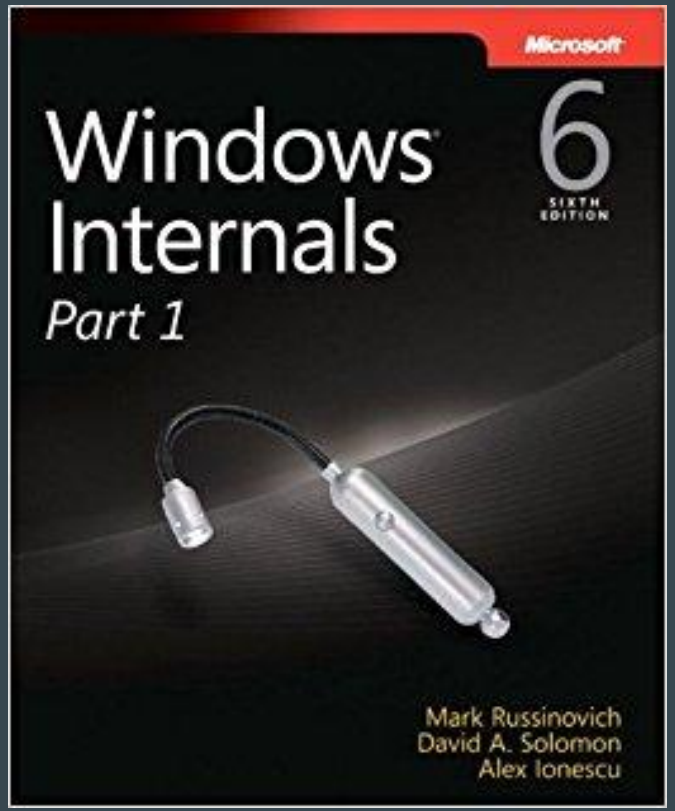
Les voies du SYSTEM sont pénétrables

...

Gestion des privilèges admin sous windows

Les vacances c’est chouette pour faire ...

... un peu de lecture



... et surtout de la veille

La voie royale

- Compromission d’un compte utilisateur du domaine
- Compromission d’un compte admin local
- Compromission d’un compte admin de domaine

La voie royale

- Compromission d’un compte utilisateur du domaine

- Compromission d’un compte admin local

- Compromission d’un compte admin de domaine

Mais pourquoi vouloir être SYSTEM?

Limitation d'un admin authentifié

- Impossible d'écrire dans le dossier windows
- Impossible de réaliser des opérations d'injection
- Impossible de mimikatzer
- Impossible de lire la base SAM
- Impossible de créer un service

Limitation d'un admin

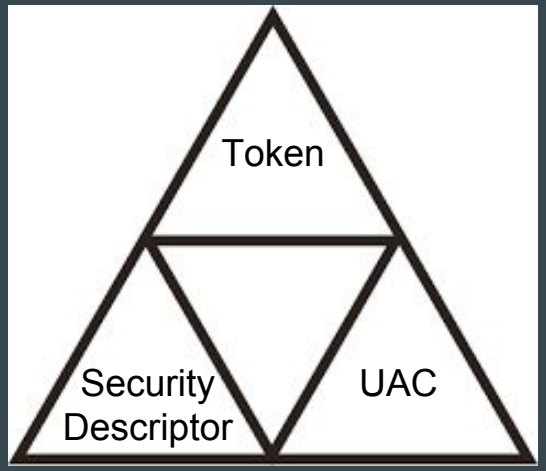
- Impossible d'écrire dans le dossier windows
- Impossible de réaliser des opérations d'injection
- Impossible de mimikatzer
- Impossible de lire la base SAM
- Impossible de créer un service

Integrity level
UAC pour les intimes...

Modèle de sécurité de windows

La triforce de windows

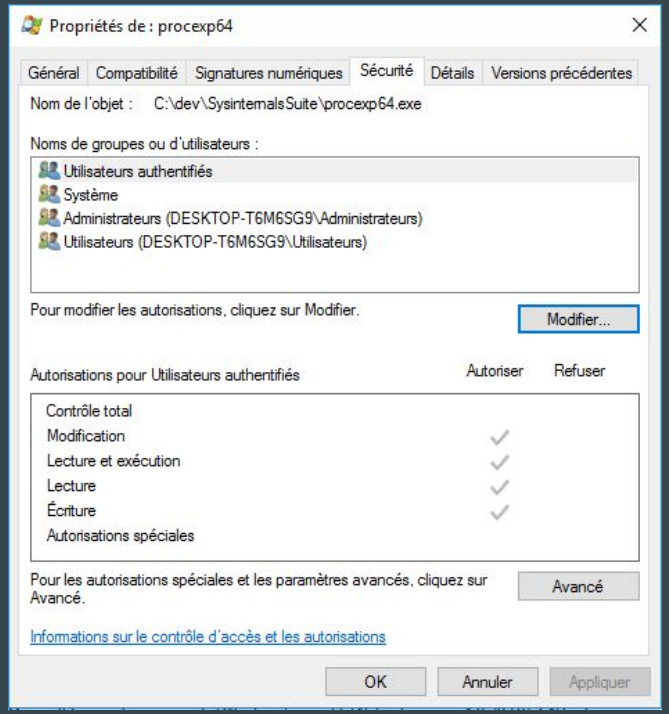
- Access token
  - Process
  - Thread
- Security Descriptor
  - Tout objets du système
- Integrity Level
  - User Access Control



Security Descriptor

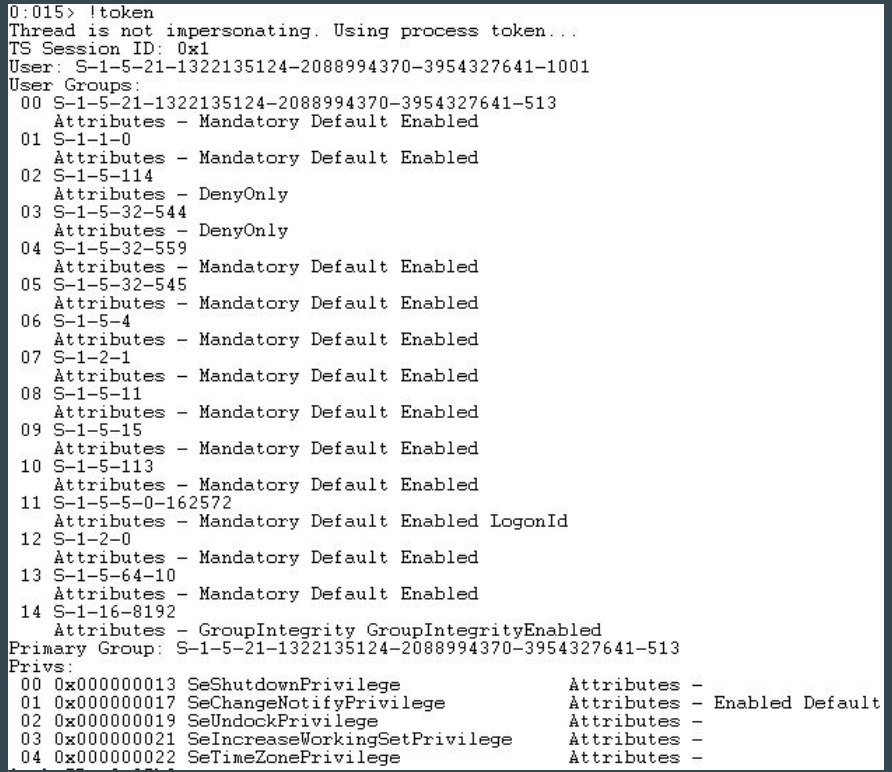
- Gestion des droits d'accès à l'objet
  - DACL
  - SACL

Attention à l'ordre des ACL !!!



Token Access

- Utilisateur et groupes associés
  - Gestion des SIDs
  - S-1-**2**-0 LOCAL account
  - S-1-**5**-**32**-544 Administrateur
  - S-1-**5**-**32**-545 Utilisateur
- Privilèges
  - SeDebugPrivilege
  - SeImpersonatePrivilege



```
0:015> !token
Thread is not impersonating. Using process token...
TS Session ID: 0x1
User: S-1-5-21-1322135124-2088994370-3954327641-1001
User Groups:
 00 S-1-5-21-1322135124-2088994370-3954327641-513
    Attributes - Mandatory Default Enabled
 01 S-1-1-0
    Attributes - Mandatory Default Enabled
 02 S-1-5-114
    Attributes - DenyOnly
 03 S-1-5-32-544
    Attributes - DenyOnly
 04 S-1-5-32-559
    Attributes - Mandatory Default Enabled
 05 S-1-5-32-545
    Attributes - Mandatory Default Enabled
 06 S-1-5-4
    Attributes - Mandatory Default Enabled
 07 S-1-2-1
    Attributes - Mandatory Default Enabled
 08 S-1-5-11
    Attributes - Mandatory Default Enabled
 09 S-1-5-15
    Attributes - Mandatory Default Enabled
 10 S-1-5-113
    Attributes - Mandatory Default Enabled
 11 S-1-5-5-0-162572
    Attributes - Mandatory Default Enabled LogonId
 12 S-1-2-0
    Attributes - Mandatory Default Enabled
 13 S-1-5-64-10
    Attributes - Mandatory Default Enabled
 14 S-1-16-8192
    Attributes - GroupIntegrity GroupIntegrityEnabled
Primary Group: S-1-5-21-1322135124-2088994370-3954327641-513
Privs:
 00 0x000000013 SeShutdownPrivilege                Attributes -
 01 0x000000017 SeChangeNotifyPrivilege            Attributes - Enabled Default
 02 0x000000019 SeUndockPrivilege                  Attributes -
 03 0x000000021 SeIncreaseWorkingSetPrivilege      Attributes -
 04 0x000000022 SeTimeZonePrivilege                Attributes -
```

Integrity level

Integrity level

Integrity level

- Apparu avec Windows VISTA (paix à son âme)
- Chaque Token possède un un niveau d'intégrité (via le système de groupes)
- Chaque objet possède un label d'intégrité
- Permet de contrôler les supers pouvoirs du compte administrateur local
    - Pour comprendre les droits que lui demande une opération

Integrity level

- Untrusted
  - Aucun accès
- Low
  - Pas d'accès en écriture (même sur le système de fichier)
- Medium
  - Un administrateur local avec privilège limité
- High
  - Un administrateur avec tous les privilège
- System
  - YOLO

Sécurité ?


FAUX

Enfin presque ...

Technique de Bypass

- C’est un peu le bizutage du chercheur en sécurité
- Beaucoup de techniques connues...
- ... peut être encore plus d’inconnues

Technique de Bypass “classique”

1. Détecter un DLL hijack dans le répertoire System32 d’un programme dit auto élevé!!!
    - Lister l’ensemble des programme auto élevés
        - Il doit être signé par microsoft
        - Il doit se trouver dans le répertoire System32
        - Il doit être marqué comme auto elevate

Technique de Bypass “classique”

```
c:\dev\SysinternalsSuite>sigcheck.exe -m c:\Windows\System32\eventvwr.exe

Sigcheck v2.54 - File version and signature viewer
Copyright (C) 2004-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\windows\system32\eventvwr.exe:
        Verified:       Signed
        Signing date:   11:33 16/07/2016
        Publisher:      Microsoft Windows
        Company:        Microsoft Corporation
        Description:    Event Viewer Snapin Launcher
        Product:        Microsoft« Windows« Operating System
        Prod version:   10.0.14393.0
        File version:   10.0.14393.0 (rs1_release.160715-1616)
        MachineType:    64-bit
        Manifest:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Copyright (c) Microsoft Corporation -->
<assembly xmlns="urn:schemas-microsoft-com:asm.v1"  xmlns:asmv3="urn:schemas-microsoft-com:asm.v3" manifestVersion="1.0">
<assemblyIdentity
    version="5.1.0.0"
    processorArchitecture="amd64"
    name="Microsoft.Windows.Eventlog.EventVwr"
    type="win32"
/>
<description>Event Viewer Snapin Launcher</description>

<trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
        <requestedPrivileges>
            <requestedExecutionLevel
                level="highestAvailable"
                uiAccess="false"
            />
        </requestedPrivileges>
    </security>
</trustInfo>
<asmv3:application>
   <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">
        <autoElevate>true</autoElevate>
   </asmv3:windowsSettings>
</asmv3:application>
</assembly>
```

Technique de Bypass “classique”

2. Créer une DLL avec la même interface
3. Copier notre DLL à côté de l’exécutable...
    - ... Mais je ne peux pas car il me faut les privilèges que je recherche...


IFileOperation

IFileOperation

- Objet COM permettant des réaliser des copies de fichier etc...
- C'est aussi un objet COM auto elevate suivant quel programme l'appel
  - Un programme signé par microsoft par exemple
  - Ex: powershell.exe et OUAIS!

```
PS C:\dev\powershell> Invoke-IFileOperation
PS C:\dev\powershell> $IFileOperation.CopyItem("C:\dev\README.md" , "C:\Windows\System32\", "toto.dll")
PS C:\dev\powershell> $IFileOperation.PerformOperations()
PS C:\dev\powershell>
```

IFileOperation

- En fait il ne vérifie pas le binaire...
- En fait il ne vérifie pas la signature...
- ... En fait ... heu ... bin en fait il vérifie le nom du binaire dans le PEB
- ... Bin je peux changer le nom de mon PEB ... (Masquerade PEB)
- Et voilà voilà....

- Ou alors je m'injecte dans explorer.exe (toujours possible) ...
- Et...
- ... c'est tout

Par contre côté red team on laisse un fichier

EventViewer

- Technique découverte en avril dernier
- Technique dite file less (sans fichier pour les non anglophones)
- Le binaire eventvwr.exe permet de lancer la console windows et utilisant le plugin associé
- Ce dernier utilise un clé de registre pour savoir le path du plugin
  - **HKCU\Software\Classes\mscfile\shell\open\command**
  - **HKCR\mscfile\shell\open\command**
- ... Bon bin on utilise cette dernière pour lancer le programme que l'on veut...

Et maintenant le SYSTEM...

- Maintenant que nous avons les privilèges on va tenter de devenir system
  - Impersonation (Windows API) permettant d'éxécuter un processus dans le security context de l'appelant
    - Créer un serveur de PIPE nommées
    - Créer un service avec l'utilisateur SYSTEM qui va venir se connecter sur le serveur
    - Utiliser la fonction ImpersonateNamedPipeClient
- Maintenant que l'on est SYSTEM on peut faire joujou avec les tokens de touts les utilisateurs de la machine...
  - OpenProcess
  - OpenProcessToken
  - CreateProcessWithToken
  - NtQueryInformations


HEAVEN
dreamstime.com

Alors c’est pas vraiment un système de sécurité ???