TP 1 : Windows système

1 Installation d'une station Windows 7

Attention
Pour l'installation de cette station, vous aurez besoin d'environ 7Go d'espace disque sur votre machine hôte.

- Commencez par installer Windows 7 dans une VM :
  - Pour installer la station, utilisez l'ISO que vous pouvez télécharger sur : `http://care.dlservice.microsoft.com/dl/download/evalx/win7/x64/EN/7600.16385.090713-1255_x64fre_enterprise_en-us_EVAL_Eval_Enterprise-GRMCENXEVAL_EN_DVD.iso`.
- Une fois l'installation réalisée, activez la license temporaire d'évaluation à l'aide de la commande suivante (vous aurez besoin de lancer un invité de commande en tant qu'administrateur) :

```
slmgr /rearm
```

- Pour le TP nous aurons besoin de plusieurs outils :
  - les outils de la suite *Sysinternals* (à télécharger sur microsoft.com) ;
  - WinSDK pour pouvoir utiliser le *kernel debugger*.
    - à télécharger sur `https://www.microsoft.com/en-us/download/details.aspx?id=8442` ;
    - choisir `GRMSDKX_EN_DVD.iso`.

2 Bases

2.1 Format PE

1. Sous linux, installez `hte` (HT Editor) ou équivalent ;
2. Ouvrez `ntoskrnl.exe` dans `hte` ;
   - on voit notamment que le fichier commence par "MZ" ;

— utilisez `hte` pour analyser le PE *header* (touche ESPACE puis "pe/header") ;
— jettez un œil à l'*entrypoint* (touche ESPACE puis "pe/image").

3. Faites de même pour `ntdll.dll`.
   — quelles différences voyez-vous ?

2.2 Modes d'exécution

1. En utilisant IDA ou un outil équivalent, retracez le chemin d'exécution entre l'appel à *ReadFile* de `kernel32.dll` à *NtReadFile* de `ntoskrnl.exe`. Voici quelques pistes pour vous aider :
   — Les fonctions importées de "API-MS-Win-Core-File-L1-1-0" ne sont pas importées de la DLL `API-MS-Win-Core-File-L1-1-0.dll`. "API-MS-Win-Core-File-L1-1-0" est un *API set*[1]. Les fonctions sont en réalité importées de la DLL `KernelBase.dll` ;
   — Pour faire le lien entre numéro d'appel système et fonction appelée, vous pouvez utiliser `livekd`[2] de *Sysinternals* :
      — utilisez la commande (de l'espace) :

```
kd> ln @@c++(((int*)@@(nt!KiServiceTable))[
<numéro d'appel système>] >> 4) +
nt!KiServiceTable
```

      — pour des explications sur cette commande, se référer à *Windows Internals, 6th edition, part 1.*

2. Visualisez le temps passé en *kernel mode* et en *user mode.*
   — Lancez le *Performance Monitor* ;
   — Ajoutez des compteurs (bouton vert en forme de '+') :
      — ajoutez "Privileged Time" et "User Time" (trouvable dans "Processor").
   — Lancez la commande :

```
dir c:\ /s
```

   — Comparez le résultat entre une exécution où la fenêtre de *cmd.exe* est au premier plan et une exécution où elle est cachée.

1. `https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-apisets`
2. pour plus de détails, allez sur `https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugger-commands`

2.3 Bases de registres

1. Lancez `regedit` ;
2. Changez le curseur de la souris en modifiant la base de registre (*hint* : `HKCU\ControlPanel\Cusors\Arrow`) ;
3. Annulez la modification avec l'utilitaire en ligne de commande `reg` ;
4. Configurez le lancement de `C:\Windows\System32\cmd.exe` au démarrage de la session de votre utilisateur (pour trouver la bonne clef de registre, Google est votre ami) ;
   > Il s'agit d'une technique souvent utilisée par les attaquants pour la persistance.
5. Au lieu de lancer `C:\Windows\System32\cmd.exe` au démarrage de la session de votre utilisateur, lancez un script BAT qui désactive le pare-feu.
   — Pour déboguer, vous pouvez utiliser la commande "whoami /groups > test.txt".

2.4 Services

1. Lancez l'outil graphique d'administration de service `services.msc` :
   — soit Windows+R et tappez `services.msc` ;
   — soit depuis le *Control Panel* ;
   — soit depuis MMC (où il faut ajouter le *snap-in* Services).

   — vous pouvez voir et modifier la configuration des services ;
   — vous pouvez aussi démarrer ou arrêter les services ;
2. Retrouvez la configuration des services dans la base de registre ;
3. À l'aide de *process explorer* de *Sysinternals* regardez les services enregistrés par les processus :
   — double clique sur un processus, puis onglet "Services" ;
   — regardez par exemple le cas du processus *lsass.exe*.
4. Créez un service qui désactive le pare-feu et installez le avec la commande `sc`.
   — N'hésitez pas à demander de l'aide pour la création du service ;
   — Quels avantages voyez-vous par rapport au script BAT lancez au démarrage de la session de votre utilisateur ?

2.5 Tâches planifiées

1. Lancez l'outil graphique d'administration de tâches planifiées `taskschd.msc` :

— créez une tâche planifiée (qui montre un message lors de la connexion par exemple).

2. À l'aide de l'outil en ligne de commande *schtasks* lancez votre tâche puis supprimez la.

2.6 Objets

1. Lancez *winobj* de *Sysinternals* :
   — ne permet de voir que les objets ayant un nom, on ne peut donc pas y voir les processus par exemple.
2. Lancez *livekd* de *Sysinternals* :
   — nous allons l'utiliser pour analyser les objets de type processus ;
   — listez les processus :

```
kd> !process 0 0
```

   — récupérez les informations sur un des processus :

```
kd> !object <adresse de l'objet processus>
```

   — récupérez le *header* de l'objet :

```
kd> dt nt!_OBJECT_HEADER <adresse du header
(ObjectHeader)>
```

3. Lancez *process explorer* de *Sysinternals* :
   — vous pouvez l'utiliser pour voir les *handles* sur des objets qu'ont les processus ;
   — allez dans "View->Low Panel View" et sélectionnez "Handles".
   — lancez un *cmd.exe* et regarder l'évolution des *handles* quand vous vous déplacez dans l'arborescence de fichiers.
4. Il est aussi possible de voir les *handles* d'un processus avec *livekd* de *Sysinternals* :
   — lancez *livekd* ;
   — listez les processus :

```
kd> !process 0 0
```

   — récupérez la liste d'*handles* d'un processus :

```
kd> !handle 0 3 <CID du processus>
```

5. Nous allons lister les fichiers ouverts de "C :" :

— lancez *livekd* ;
— récupérez le nom du *device* correspondant à la lettre "C" :

```
kd> !object \Global??\C:
```

— récupérez l'adresse de l'objet *device* :

```
kd> !object \Device\HarddiskVolume1
```

— récupérez la liste des fichiers ouverts :

```
kd> !devhandles <adresse du device>
```

— la liste est longue, vous pouvez arrêter l'exécution avant la fin.

3 Contrôle d'accès

1. À l'aide de `psgetsid` de *Sysinternals* retrouvez le SID de votre compte et celui du compte administrateur ;
2. À l'aide de *whoami /all* analysez votre jeton d'accès ;
3. Analysez les jetons d'accès des processus à l'aide de *process explorer* :
   — clique droit sur un processus puis "Properties>Security".
4. Analysez les jetons d'accès des processus à l'aide de *livekd* :
   — récupérez l'adresse du jeton d'accès :

```
kd> !process <CID du processus> 1
```

   — utilisez `!token` pour voir le jeton :

```
kd> !token <adresse du jeton>
```

5. Analysez les descripteurs de sécurité à l'aide d'*explorer* :
   — clique droit sur un fichier puis "Properties>Security".
6. Analysez les descripteurs de sécurité à l'aide de *livekd* :
   — récupérez l'adresse du *header* de l'objet :

```
kd> !object <adresse de l'objet>
```

   — récupérez le *header* de l'objet :

```
kd> dt nt!_OBJECT_HEADER <adresse du header
(ObjectHeader)>
```

   — analysez le descripteur de sécurité de l'objet :

```
kd> !sd <adresse du descripteur de sécurité
(SecurityDescriptor)> & -10
```

7. Tentez d'accèder à la clef de registre `HKLM\SAM` ;
   — analysez le problème et contournez-le
8. Avec l'outil *User account* créez un nouvel utilisateur non administrateur ;
9. Avec l'outil *Local Security Policy* supprimez le privilège de changement de *time zone* au groupe *Users* ;
   — Tentez de changer de fuseau horaire avec l'utilisateur non administrateur ;
      — clique droit sur l'horloge ;
      — puis bouton "Change time zone...".
   — Analysez le jeton de l'utilisateur administrateur et celui de l'utilisateur non administrateur.
10. Volez le jeton d'accès du processus lsass.exe (permettant d'obtenir un jeton d'accès de l'utilisateur Système) et donnez-le à un cmd.exe s'exécutant avec l'utilisateur normal :
    — Les processus sont des objets de type EPROCESS. Pour récupérer l'*offset* du champ *token* dans la structure EPROCESS vous pouvez utiliser la commande suivante :

      ```
      kd> dt nt!_EPROCESS <adresse du processus> Token
      ```

    — Pour récupérer la valeur d'un pointeur, utilisez la fonction *poi(<adresse du pointeur)* ;
    — Pour modifier un pointeur, utilisez la commande *ep* (*edit pointer*) :

      ```
      kd> ep <adresse du pointeur> <nouvelle valeur>
      ```

    — *livekd* permet de travailler seulement en lecture, ce n'est pas suffisant dans notre cas. Nous allons utiliser *kd* en mode local :
      — il faut d'abord activer une option de démarrage :

        ```
        bcdedit /debug on
        ```

      — puis redémarrer ;
      — on peut alors utiliser *kd* en mode local :

        ```
        kd -kl
        ```

4 Authentification

Dans la partie précédente, nous avons vu comment obtenir un cmd.exe Système en volant un jeton d'accès. Il existe d'autres façons d'en obtenir. Nous allons en essayer quelques unes.

1. Nous avons déjà créé un service pour désactiver le pare-feu. Inspirez-vous en pour obtenir un `cmd.exe` Système.
   — D'après vous, à quoi correspond le changement d'environnement graphique ?
2. L'outil `psexec.exe` de *Sysinternals* permet d'exécuter des commandes à distance. Il peut aussi être utiliser localement et possède une option pour s'exécuter en Système. Utilisez-le pour obtenir un cmd.exe Système ;
3. Sur l'écran de login, la pression de la touche MAJ répétée 5 fois entraine le lancement de `C:\Windows\System32\sethc.exe`, en tant que Système. Utilisez ce mécanisme pour obtenir un cmd.exe Système.

Bonus : changez le *shell* de votre utilisateur de *explorer.exe* à *cmd.exe*.

5 Mécanismes de protection

5.1 Niveaux d'intégrité

1. Lancez un *cmd.exe* normalement puis en tant qu'administrateur :
   — à l'aide de *livekd*, comparez les jetons d'accès des deux *cmd.exe*.
2. À l'aide de l'outil *string* de *Sysinternals*, analysez le manifest de `regedit.exe` ;
3. Lancez *Internet Explorer* ;
   — à l'aide de *process explorer*, analysez les niveaux d'intégrité des processus d'*Internet Explorer* ;
      — allez dans "View>Select column..." puis cochez "Integrity level".
   — en analysant les jetons d'accès des processus d'*Internet Explorer*, retrouvez leurs niveaux d'intégrité ;
   — trouvez le niveau d'intégrité de l'objet processus d'*Internet Explorer* (en analysant son descripteur de sécurité) ;
4. Créez un fichier et configurez son niveau d'intégrité à *Low* avec `icacls` ;

5.2 Autres

1. Ajoutez une règle de pare-feu pour interdire l'utilisation d'*Internet Explorer* ;
2. Créez les règles par défaut d'*Applocker* :
   — pour fonctionner *Applocker* a besoin du service *Application Identity*, démarrez le ;
   — lancez *Local Security Policy* et allez dans "Applications Control Policies" ;
   — clique droit sur "Executable Rules" puis "Create Default Rules" ;
   — avec l'utilisateur non administrateur, copiez *cmd.exe* sur votre bureau et tentez de le lancer ;
   — dans *Event Viewer*, regardez les logs d'*Applocker*.
3. Utilisez *Bitlocker to Go* pour chiffrer un clef USB.