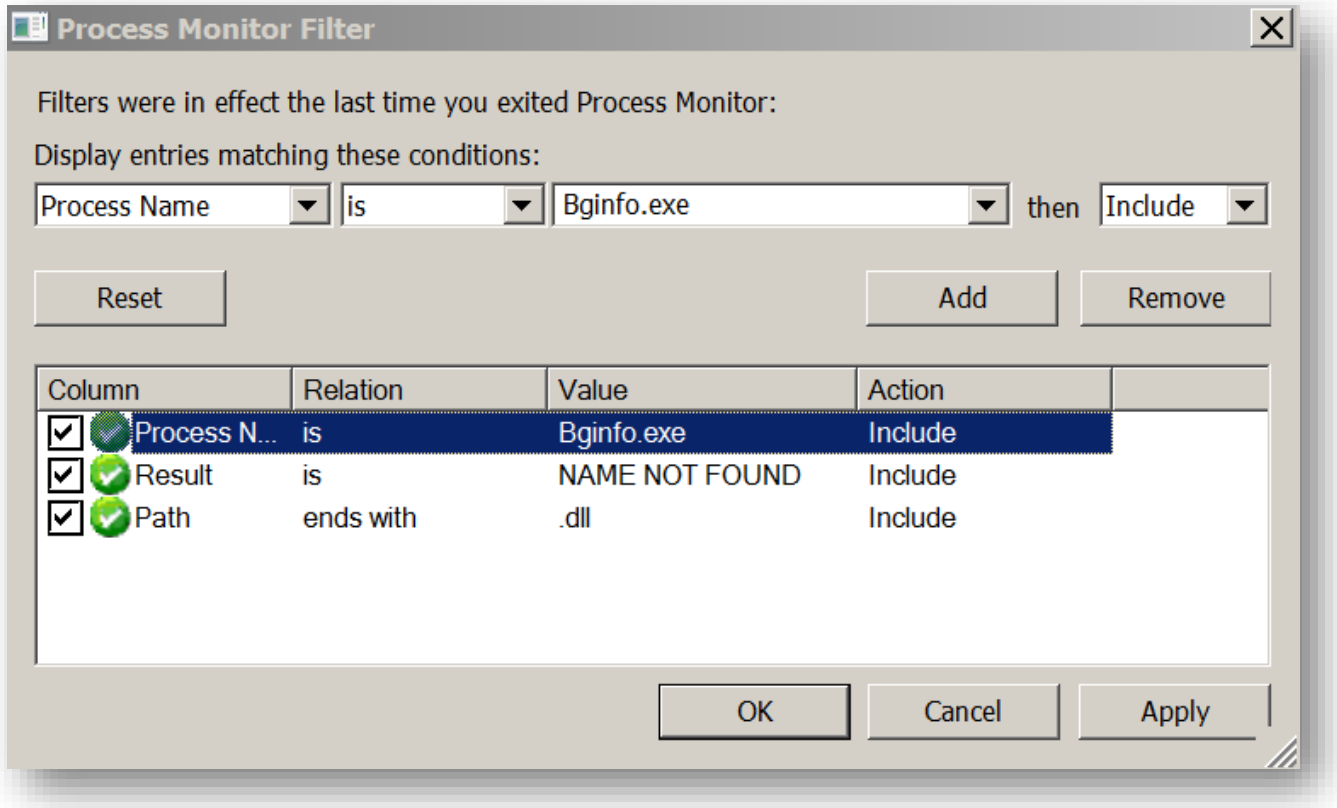
DLLirant

Parce que c'est DéDéLirant!

Explications basiques du DLL Hijacking

- Méthode utilisée en opérations Red-Team ainsi que par des acteurs malveillants
- Elle consiste à exécuter du code malveillant sous forme de librairie (DLL) via un programme légitime et potentiellement signé, via les fonctions du programme de façon « exportées »
- En principe, moins détecté que via un simple binaire qui va exécuter une charge malveillante
- C’est bien beau mais faut le trouver ce programme légitime vulnérable

Première méthode, la méthode chiante



- Via des filtres sur l'exécutable en utilisant Process Monitor
- Inconvénient, c'est long et fastidieux et DéDé il a pas le temps

La seconde méthode, via dumpbin (parsing IAT)

```
Dump of file nslookup.exe

File Type: EXECUTABLE IMAGE

  Section contains the following imports:

    msvcrt.dll
             14000D448 Import Address Table
             1400113D8 Import Name Table
                     0 time date stamp
                     0 Index of first forwarder reference

                         4A0 putchar
                         4B9 sprintf_s
                         461 gmtime
                         432 exit
                         363 _vsnprintf
                         304 _strnicmp
                         439 fflush
                         486 malloc
                         ...
    DNSAPI.dll
             14000D238 Import Address Table
             1400111C8 Import Name Table
                     0 time date stamp
                     0 Index of first forwarder reference

                        45 DnsFreeConfigStructure
                        75 DnsQueryConfigAllocEx
```

Troisième méthode, DLLirant

- Dumpbin aurait pu suffire mais cela demande Visual Studio d'installé
- Il faut passer par mal de temps à tester chaque module et chaque fonction en recompilant des PoCs en boucle et tester
- DLLirant permet de parser l'IAT en python via la librairie pefile
- Il recompile ensuite le code à tester en C++, et le teste lui-même
- Si une DLL Hijacking est possible, il affiche une MessageBox et enregistre les fonctions à exporter dans un fichier de logs
- Enfin, il est possible de transformer la DLL Hijacking en DLL Proxying automatiquement et ainsi éviter de couper l'exécution du programme original pour exécuter sa payload

Les possibilités d'évolution

- Créer la DLL from scratch avec Python (plus besoin de clang comme compilateur, merci à **@_hugsy_** pour le tips)
- Peut être faire un GUI directement en .NET pour que ça soit du one-click payload à partir d'un .bin
- Il est l'heure de la démo car DéDé s'impatiente

DEMO

- Merci à **@Geluchat** pour la relecture, également merci au staff **@BeeRumP_Paris** et bonne binouze à tous et à toutes !

- Références si vous voulez aller plus loin :

https://github.com/Sh0ckFR/DLLirant

https://sh0ckfr.com/pages/martine-a-la-recherche-de-la-dll-hijacking-perdue/

https://sh0ckfr.com/pages/martin-et-le-dll-proxying-de-cristal/

https://hijacklibs.net/