

"Catch me if you can" ou la détection des intrusions réseaux

Jean-Marc Pouchoulon

octobre 2024



1 Avant de commencer

1.1 Pré-requis, recommandations et notation du TP.

Vous travaillerez individuellement. Deux QCM (analyse de pcap) permettront de vous évaluer sur cette partie "Hunting".

1.2 Logiciels à installer pour commencer

- "Zui" (Téléchargement via <https://www.brimdata.io/download/>). Zui est un logiciel permettant de lire plus rapidement et d'analyser plus facilement des "dumps" réseaux au format "pcap".
- Nushell

```
# Installez d'abord rust
curl --proto '==https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

# N'oubliez pas d'ajouter le répertoire ~/.cargo/bin à votre PATH:

export PATH="$~/.cargo/bin:$PATH"
source .zshrc

# Puis installez NuShell et les plugins choisis:

~/cargo/bin/cargo install nu
~/cargo/bin/cargo install nu_plugin_formats
~/cargo/bin/cargo install nu_plugin_polars
~/cargo/bin/cargo install nu_plugin_query

# Après avoir relancé NuShell référez les plugins Polars et Query auprès de NuShell:

nu
plugin add ~/.cargo/bin/nu_plugin_polars
```

```

plugin add ~/.cargo/bin/nu_plugin_query
plugin add ~/.cargo/bin/nu_plugin_formats
exit
nu
plugin list

```

- zeek "Zeek" est un concurrent de "Suricata" qui permet de générer des traces protocolaires au format json. Ce sont ces traces que nous allons analyser à l'aide de "jq" et "nushell".

json est un format de choix pour les développeurs mais il est aussi de plus en plus utilisé dans le domaine de la sécurité. (fichier eve.json de Suricata par exemple). Le pcap a été transformé à l'aide de Zeek en fichier json afin de permettre une analyse à l'aide de requête jq.

A l'aide de requêtes jq et des fichiers au format json¹ issu du pcap et générés à l'aide de zeek, retrouvez les éléments d'analyse de l'article.² Générez les traces au format json à partir du pcap issu de: <https://github.com/pan-unit42/wireshark-tutorial-Emotet-traffic> en utilisant Zeek.

Vous pouvez installer zeek via docker. Voir <https://hub.docker.com/r/zeek/zeek> Depuis la directory où se trouvent le pcap, générez les traces zeek au format json via les commandes:

```

docker run -v ./:/ -it docker.io/zeek/zeek:latest bash
mkdir -p output
cd output
zeek -r /usr/local/zeek/logs/Example-1-2021-01-06-Emotet-infection.pcap LogAscii::use_json=T

```

- "jq" (installation via apt). jq est un parseur de "chunks" au format json extrêmement populaire et qui peut servir pour analyser aussi des traces de sécurité au format json.

2 Analyse des traces réseaux du botnet Emotet à partir de fichier Zeek

On va analyser les traces réseaux afin de connaître son "modus operandi".

2.1 Analyse post mortem de traces du virus Emotet au format json

1. Affichez l'adresse ip des machines communiquant entre elle via http ainsi que les "uri".
2. Dans le trafic http sélectionnez les communications relatives aux sites "hangarlastik.com", "padrees-
capes.com", sarture.com et "seo.udaipurkart.com". Affichez les "IP sources", les IP de "destinations"
et les "uri" sur la sélection.
3. Sélectionnez les communications relatives à la ddl contenant Emotet "nDUrg8uFD5hldll" (files)
depuis "files.log".
4. Sélectionnez les requêtes de type "POST" sur les ports de destination 80 ou 8080.

2.2 Analysez le premier pcap d'Emotet à l'aide de Zui

Ce pcap est le témoin de l'infection d'un serveur par Emotet. L'analyste vous précédent a vérifié avec nrich que les sites "hangarlastik.com", "padrees-
capes.com", "sarture.com" et "seo.udaipurkart.com" sont des sites vulnérables et probablement malicieux.

```

#hangarlastik.com
89.252.164.58
#padrees-
capes.com
66.153.205.191
#sarture.co
173.255.195.246

```

1. fichiers sur moodle

2. voir <https://www.sans.org/blog/parsing-zeek-json-logs-with-jq/>

#seo.udaipurkart.com

103.92.235.25

echo 89.252.164.58 | nrich -

89.252.164.58

Ports: 22, 53, 80, 110, 443, 465, 993, 995, 2082, 2083, 2086, 2087

Tags: starttls

CPEs: cpe:/a:exim:exim:4.94.2, cpe:/a:openbsd:openssh:7.4, cpe:/a:apache:http_server

Vulnerabilities: CVE-2017-15906, CVE-2018-15919

nrich sites.txt

66.153.205.191

Ports: 80, 443

CPEs: cpe:/a:jquery:jquery, cpe:/a:jquery:jquery_ui, cpe:/o:microsoft:windows, cpe:/a:getbootstrap:bootstrap

89.252.164.58

Ports: 22, 53, 80, 110, 443, 465, 993, 995, 2082, 2083, 2086, 2087

Tags: starttls

CPEs: cpe:/a:exim:exim:4.94.2, cpe:/a:openbsd:openssh:7.4, cpe:/a:apache:http_server

Vulnerabilities: CVE-2017-15906, CVE-2018-15919

173.255.195.246 (li205-246.members.linode.com)

Ports: 21, 53, 80, 110, 143, 443, 465, 587, 993, 2082, 2083, 2087, 2525, 3306

Tags: database, starttls, cloud

CPEs: cpe:/a:pureftpd:pure-ftpd, cpe:/a:exim:exim:4.94.2, cpe:/a:apache:http_server, cpe:/a:mysql:mysql

103.92.235.25 (server27.hostingraja.org)

Ports: 21, 53, 80, 443, 465, 587, 995, 2079, 2082, 2087, 3306

Tags: database, starttls

CPEs: cpe:/a:pureftpd:pure-ftpd, cpe:/a:mysql:mysql:5.6.51, cpe:/a:exim:exim:4.94.2, cpe:/a:php:php, cpe:/a:apache:http_server, cpe:/a:jquery:

Ces requêtes de bases données par l'analyste vont vous permettre de comprendre le fonctionnement du langage ZQL de Zui.

Testez-les:

```
count() by _path | sort -r
_path=="dns" | count() by query | sort -r
_path matches smb* OR _path=="dce_rpc"
_path=="http" | cut id.orig_h, id.resp_h, id.resp_p, method, host, uri | uniq -c
_path=="conn" | cut id.orig_h, id.resp_p, id.resp_h | sort | uniq
filename!=null | cut _path, tx_hosts, rx_hosts, conn_uids, mime_type, filename, md5, sha1
method=="POST" | cut ts, uid, id, method, uri, status_code
_path=="conn" | put classnet := network_of(id.resp_h) | cut classnet | count() by classnet | sort -r
event_type=="alert" | count() by alert.severity, alert.category | sort count
event_type=="alert" | alerts := union(alert.category) by src_ip, dest_ip
event_type=="alert" | alerts := union(alert.category) by network_of(dest_ip)
```

1. Retrouver le "GET" sur "seo.udaipurkart.com" et le nom de la librairie "dll" téléchargée en réponse au post. Regardez la corrélation que trouve Zui avec "files" et les alertes Suricata. Ouvrez la sélection dans Wireshark depuis Zui.
2. Depuis Zui lancez Wireshark pour en extraire la dll.
3. Retrouvez les requêtes relatives au trafic "C2" (Control & Command). En extraire les réseaux qui hébergent les "C2".

3 Défacement d'un site web par une vilaine grenouille

Description du Challenge

source du challenge: first 2015 par Erik Hjelmvik, Swedish Armed Forces CERT

Le défacement de `www.pwned.se` a eu lieu le 12 Mars à 12:58 UTC. L'attaquant a "uploadé" une image de grenouille `www.pwned.se/skyblue/fr.jpg` Le réseau est le suivant:

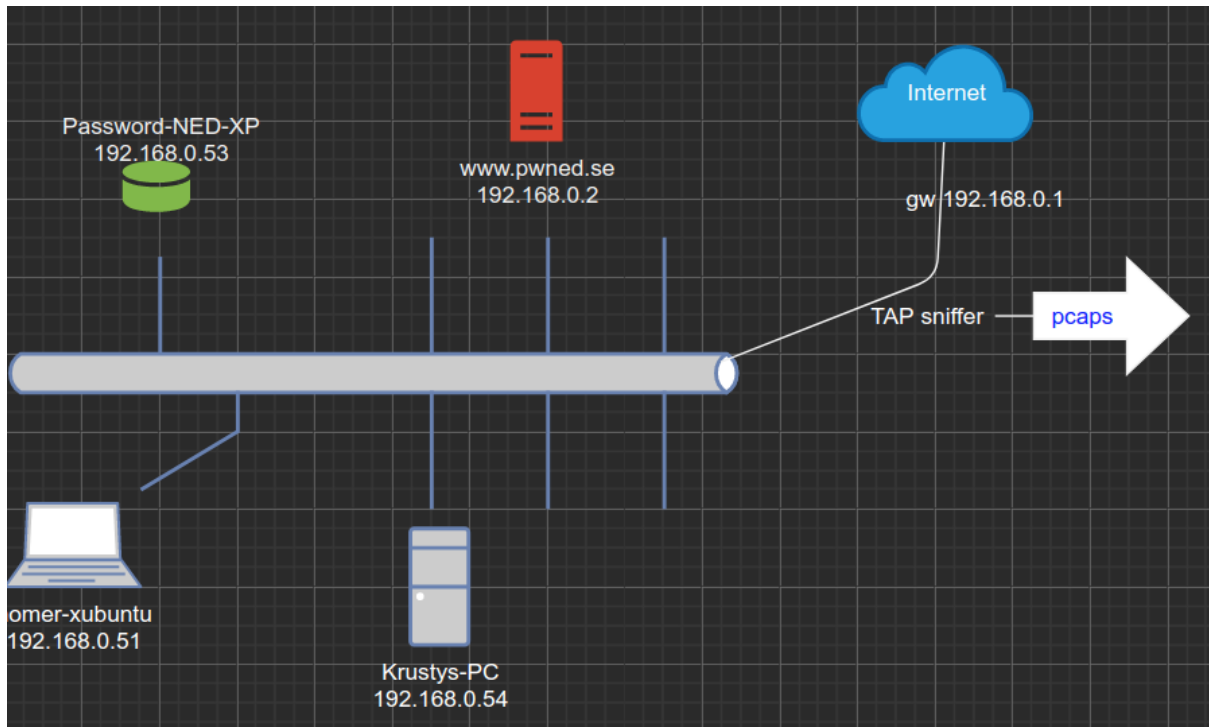


FIGURE 1 – Schéma de l'infrastructure "défacée".

A l'aide de Zui répondez aux questions suivantes:

1. Donnez la commande `zql` (langage de Zui) permettant de trouver l'IP de l'attaquant.
2. Quelle est l'IP du second attaquant et quel outil a-t-il utilisé ? a-t-il réussi ?
3. Comment l'attaquant a-t-il procédé ?
4. Quelle est la CVE utilisée par l'attaquant ?
5. Quel type d'alerte Suricata permet de visualiser les outils de l'attaquant ? Suivez les flux.
6. Sur quel port le "phpshell" de l'attaquant est-il accessible ?
7. Quel est le nom du phpshell de l'attaquant ?
8. Ned et Homer ont-ils un compte sur la machine compromise ?

4 Utilisation de Suricata pour analyser un pcap

Selks est une distribution basée sur Debian qui permet de déployer rapidement un IDS/IPS basé sur Suricata. Elle est basée sur docker et permet de déployer rapidement un IDS/IPS basé sur Suricata.

4.1 Installation de Suricata Selks

```
# création d'une interface virtuelle sur laquelle on va faire passer le traffic  
sudo ip link add veth0 type veth peer name veth1
```

```

sudo ip link set veth0 up
sudo ip link set veth1 up

git clone https://github.com/StamusNetworks/SELKS.git
cd Selks/docker
# utiliser veth1 lors des saisies du script d'installation
./easy-setup.sh --scirius-version selks --iA
sudo -E docker compose up -d
# Interface portainer accessible sur https://your_IP:9443 mot de passe admin à changer
# Interface Selks accessible sur https://your_IP user :selks-user password: selks-user

```

Modifiez selks6-addin.yaml afin de pouvoir retrouver les adresses mac:

```

- eve-log:
  enabled: yes

```

puis

```

sudo docker compose restart suricata

```

time for first test...

4.2 Utilisation de Suricata

1. Chargez le pcap snort.log.1426118407

```

# Le -c nettoie des précédents pcap
scripts/readpcap.sh -c .../snort.log.1426118407

```

2. Dans l'interface web retrouvez les alertes de sécurité dans "hunting" puis dans les dashboards kibana. N'oubliez pas de sélectionner la date des logs ou un intervalle de dates large.
3. Parsez le fichier eve.json avec jq afin d'extraire uniquement les alertes (il existe un chapitre dédié à ce sujet dans la documentation Suricata³).
4. Donnez le top 10 des ports de destination.

time for second test...

3. <https://docs.suricata.io/en/suricata-7.0.0/output/eve/eve-json-examplesjq.html>