Workshop détection des intrusions réseaux

Jean-Marc Pouchoulon

Mars 2022

1 Avant de commencer

1.1 Mises en situation professionnelle lors du TP et compétences à valider

- Utiliser jq afin de parser des logs au format json.
- Détecter une tentative d'intrusion en temps réel à l'aide d'un IDS.
- Analyser une intrusion post-mortem (via Brim , Zeek, Suricata).
- Utiliser un SIEM afin de corréler les évènements de sécurité.
- Détecter une tentative d'intrusion en temps réel à l'aide d'un honeypot.

1.2 Pré-requis, recommandations et notation du TP.

Vous travaillerez individuellement. Un QCM permettra de vous évaluer. Il est pour partie en rapport direct avec les exercices de ce workshop.

1.3 Logiciels à installer pour commencer

- "brim" (installation via `https://github.com/brimdata/brim/wiki/Installation`). Brim est un logiciel permettant de lire plus rapidement et d'analyser plus facilement des "dumps" réseaux au format "pcap".
- "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.
- installation de "nrich" via `https://gitlab.com/shodan-public/nrich/-/tree/master`. C'est un utilitaire permettant d'interroger l'API de "Shodan" afin de récupérer les résultats du scan d'un IP. Il apporte des informations précieuses lors de la détection et de l'analyse des incidents de sécurité réseau.

2 Analyse des traces réseaux du botnet Emotet

Emotet est un botnet encore très actif et en constante mutation encore à la date d'écriture de ce workshop. Il est donc intéressant d'analyser les traces réseaux afin de connaître son "modus operandi".

2.1 Analyse des traces réseaux d'Emotet à l'aide de WireShark

[1]

2.2 Analyse post mortem de traces au format json avec emotet

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 json [2] issu du pcap et générés à l'aide de zeek ,retrouvez les éléments d'analyse de l'article. [3]

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", "padreescapes.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 "nDUrg8uFD5hl$\dot{d}$ll" (files) depuis "files.log".
4. Sélectionnez les requêtes de type "POST" sur les ports de destination 80 ou 8080.

2.3 Analyser le premier pcap à l'aide de brim et de nrich

1. Vérifiez a "virus total" que les sites "hangarlastik.com", "padreescapes.com", "sarture.com" et "seo.udaipurkart.com" sont bien des sites malicieux.
2. A l'aide de "nrich" trouvez les ports ouverts et les vulnérabilités depuis les sites "fournisseurs" d'Emotet et qui ont probablement conduits à la compromission de ces hôtes.
3. Retrouver le "GET" sur "seo.udaipurkat.com" et le nom de la librairie "dll" téléchargée en réponse au post. Regardez la corrélation que trouve Brim avec "files" et les alertes suricata. Ouvrez la sélection dans wireshark depuis Brim.
4. Depuis brim lancez WireShark pour en extraire la dll.
5. Retrouvez les requêtes relatives au traffic "C2" (Control & Command). En extraire les réseaux qui hébergent les "C2".

3 Défacement d'un site web par une vilaine grenouille

[4]

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 :

1. le fichier pcap se trouve sur Moodle
2. fichiers sur moodle
3. voir `https://www.sans.org/blog/parsing-zeek-json-logs-with-jq/`
4. source du challenge first 2015 par Erik Hjelmvik, Swedish Armed Forces CERT

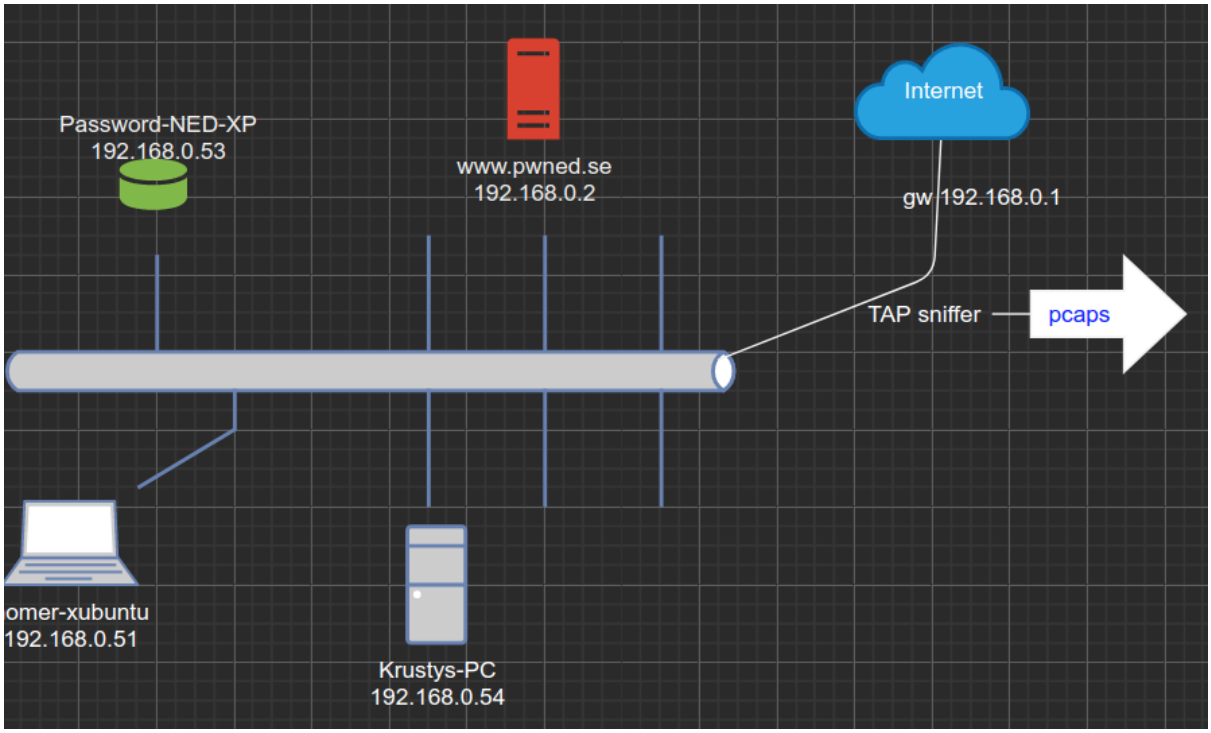

FIGURE 1 – Schéma de l'infrastructure "défacée".

A l'aide de brim répondez aux questions suivantes :

1. Donnez la commande zql (langage de Brim) permettant de trouvez 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 Installation d'un Honeypot

Vous installerez le multi-honeypot "TPOT" [5]. Vous lancerez des attaques sur les ports 22, 23, 25, 21, 80 et vous en vérifierez l'impact sur les tableaux de bord du honeypot. Vous ferez constater à l'enseignant le résultat.

5 Utilisation d'un SIEM

Utilisez le projet suivant `https://github.com/pushou/siem.git` afin d'installer "elastic SIEM", l'IDS "Suricata", Evebox, et Zeek. La configuration nécessaire est musclée et une machine avec 16go de Ram est un minimum. Vous obtiendrez de l'aide en lançant la commande "make help". Modifiez le fichier /etc/sysctl

```
vm.max_map_count=262144
```

Puis

5. voir `https://github.com/telekom-security/tpotce`

```
systctl -p
```

```
make es
make siem
```

5.1 Utilisation de Suricata afin de lire des pcaps

Suricata permet de détecter des intrusions mais il peut aussi relire des "dumps" réseaux. Le script suri-ingest-pcap.sh permet d'afficher les alertes générées dans suricata. Les fichiers sur l'hôte dans ./logs correspondent dans les containers à la directory /var/log/suricata.

1. Retrouvez les alertes générées dans 'Elastic Search', puis dans Kibana.

```
docker exec -it suricata bash -c '/var/log/suricata/suri-ingest-pcap.sh \
        /var/log/suricata/pcaps/défacement/2015-03-12/snort.log.1426118407'
```

2. Parsez le fichier eve.json avec jq afin d'extraire uniquement les alertes.
3. Donnez le top 10 des ports de destination.
4. Quel filtre permet d'afficher les enregistrements relatif au défacement ? en déduire l'agent responsable de la remontée des logs suricata vers Elastic Siem.
5. Retrouvez le fichier de configuration de l'agent et faite un schéma sommaire des flux de données.
6. Vérifiez que les alertes sont bien remontées dans Evebox.
7. Quel est l'outil responsable de la remontée des informations dans EveBox. Expliquez en le fonctionnement.
8. Explorer les tableaux de bords de sécurité. Suricata monitore-til en temps réel ?
9. Charger des règles proposée
10. Installer auditbeat et filebeat sur votre Linux via apt. Configurez auditbeat.yaml en reprenant ces éléments :

```
setup.kibana:
  host: "localhost:5601"


output.elasticsearch:
        hosts: ["https://localhost:9200"]
        username: "elastic"
        password: "le password de votre instance"
        ssl:
          enabled: true
          certificate_authorities: ["/home/pouchou/siem/temp/ca.crt"]
```

Verifier la bonne tenue de l'installation via :

```
sudo /usr/bin/auditbeat test config -v
journalctl -u -f auditbeat
```

11. Générer des erreurs d'authentification et vérifier qu'elastic SIEM les voit. Activez le "dashboard" relatif à auditbeat.
12. Installer auditbeat sur un windows du cloudlab. Activez des règles pour windows.
13. Installer elastic-agent sur votre machine comme EDR. (utilisez le mode manuel)

```
sudo ./elastic-agent install --insecure  --url=http://localhost:8220 \
  --fleet-server-es=https://localhost:9200 \
  --fleet-server-service-token=AAEAAWVsYXN0aWMvZmxlZXQtc2VydmVyL3Rva2VuLTE2NDU1Njc5NTkwMTE6eXpZU2tqYW5RLS1kdEd4
  --fleet-server-policy= 499b5aa7-d214-5b5d-838b-3cd76469844e \
  --fleet-server-insecure-http \
  --fleet-server-es-ca=/var/lib/suricata/temp/ca.crt
Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you want to continue? [Y/n]:Y
{"log.level":"info","@timestamp":"2022-02-22T23:22:35.038+0100","log.origin":{"file.name":"cmd/enroll_cmd.go","file.line":778},"messag
```

14. Utilisez 'atomic redteam" [6] afin de génerer des alertes conforment à la matrice MITRE.

6. https://github.com/redcanaryco/atomic-red-team