Master 2 SeCReTS
Surveillance et gestion d'incidents

TP : Capture et analyse réseau

Février 2020

Première partie
Généralités

1 Objectifs du TP

L'objectif de ce TP est de se familiariser avec la capture réseau et de prendre conscience des données qui peuvent transiter en clair. On manipulera notamment des outils basiques qui permettent d'extraire des informations sur la base de flux réseau.

La deuxième partie du TP est consacré à l'IDS Suricata. On utilisera en particulier cet outil pour lever des alertes sur la base de signatures que l'on construira.

Une troisième partie s'intéresse à l'utilisation de l'outil argus, un analyseur de flux réseau.

Enfin, à partir des captures réseau que l'on aura faites, nous utiliserons l'outil Bro pour extraire des fichiers de différents types.

2 Pré-requis

Vous devez avoir monté l'architecture vue dans les TPs précédents, avec à minima :

- La machine virtuelle Metasploitable connectée au réseau interne ;
- La machine virtuelle Kali connectée au réseau externe ;
- La machine virtuelle vmcollecte en tant que routeur au centre de l'architecture.

Les règles de routages doivent permettre d'accéder depuis la Kali, à l'ensemble des services présents sur la machine virtuelle Metasploitable.

Pendant tout le TP nous aurons besoin de fichiers de capture. Relancer le script de capture que vous avez créé dans le TP de lundi afin de capturer le trafic qui transite par la VM de collecte.

Vous pouvez modifier votre script afin de faire tourner les fichiers de capture dès qu'ils dépassent une certaine taille (par exemple 100Mo). Vous pouvez obtenir ce comportement avec une option de la commande `tcpdump`.

3 Surveillance ARP et suite Dsniff

A l'aide de tcpdump, faites ressortir le trafic ARP qui transite sur l'interface externe.

Installez `arpwatch` sur la VM de collecte et lancez le service associé. Les logs sont par défaut envoyé dans `/var/log/syslog`. Surveiller les logs émis par arpwatch. En même temps, depuis la Kali, utilisez `arpspoof` pour vous faire passer successivement pour les IPs 10.0.0.10-15.

Que permet ce type de surveillance ? Quelles sont les limites de celle-ci ?

La suite Dsniff est installée sur la Kali. Quels sont les différents exécutables installés par dsniff (utilisez la commande dpkg -L) ?

Depuis la Kali, naviguez sur les différentes pages présentes sur le service web de la machine metasploitable à l'aide de firefox. Après avoir recopié le pcap de votre session de surf, lancer `urlsnarf` sur votre fichier de capture. Que permet-il de faire ressortir ? Avec quel autre outil, auriez vous pu faire ressortir ces informations ?

Utilisez tshark pour faire ressortir des informations similaires.

```
tshark -nr web.cap -T pdml
tshark -nr web.cap -T fields -e http.host -e http.user_agent -e http.request.uri
```

Installez un serveur pop/imap sur la VM De collecte (dovecot-imapd et dovecot-pop3d) ainsi que le client mutt sur la kali.

Une configuration mutt permettant de s'authentifier à un serveur pop ou imap est la suivante (/root/.muttrc) :

```
# IMAP:
set spoolfile='imap://10.0.0.254/INBOX'
set folder='imap://10.0.0.254/INBOX'
# POP:
#set spoolfile="pop://10.0.0.254/INBOX"
#set folder="pop://10.0.0.254/INBOX"
```

Après avoir tenté de vous connecter successivement au serveur pop et imap, utiliser dsniff sur vos fichiers de capture pour faire ressortir les identifiants.

A l'aide de la commande suivante, faite ressortir ces mêmes identifiants. Comment sont ils encodés ?

```
tshark -nr pop.cap -T fields -e pop.request.command
```

4 Suricata

L'outil Suricata est pré-installé sur la VM de collecte. Afin de le configurer, vous devez modifier plusieurs fichiers.

Tout d'avord, ouvrez le fichier `/etc/default/suricata` et modifiez les lignes appropriées pour que Suricata soit en mode de capture pcap, et qu'il capture sur l'interface connectée au réseau virtuel `reseau_interne`.

Ensuite ouvrez le fichier `/etc/suricata/suricata-debian.yaml`. En particulier vous ajouterez le chargement d'un fichier qui contiendra vos propres règles (repérez la section correspondante dans le fichier) :

```
[...]
rule-files:
  - local.rules
  - botcc.rules
[...]
```

Le fichier dans lequel vous allez ajouter des règles personnalisées est `/etc/suricata/rules/local.rules`. Voici une première règle :

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:1000001;rev:1; classtype:icmp-
    event;)
```

Quel type de trafic va déclencher cette alerte ? Après avoir lancé Suricata (systemctl restart suricata), vérifiez que des alertes sont bien levées lorsque ce type de trafic ce présente.

Recopiez et adaptez la règle pour lever une alerte lorsqu'un client se connecte au port 22 d'une machine avec un client se présentant en tant que "OpenSSH 3vil" (bannière ssh). On pourra simuler cette tentative avec l'outil `nc`.

La règle suivante permet de lever une alerte en cas de scan nmap :

```
alert tcp any any -> $HOME_NET any (msg":TCP Port "Scanning; detection_filter:track by_src,
     count 10, seconds 10; sid:1000003; rev:2;)
```

Mettez en place cette règle et validez la levée d'alerte. Comment est-il possible de scanner sans se faire détecter ? Adaptez votre ligne de commande nmap afin de ne pas lever d'alertes.

La règle précédente lève une alerte à chaque fois qu'un flux réseau correspondant est détecté. Dans le cas d'un scan de port agressif d'une grande quantité de port, cela va générer de nombreuses alertes. Le fichier `/etc/suricata/threshold.config` permet d'éviter d'avoir plusieurs fois la même alerte :

```
event_filter gen_id 1, sig_id 1000003, type limit, track by_src, count 1, seconds 60
```

Depuis la Kali, utilisez `nikto` pour scanner l'arborescence du serveur Web de la machine Metsploitable. Après avoir analysé la capture du scan, quelles règles pouvez-vous mettre en place pour détecter ce type d'activité ?

5 Argus

Argus est préinstallé sur la machine virtuelle `vmcollecte`. Comme les autres outils, on va le configurer pour qu'il capture le trafic sur l'interface relié au réseau virtuel interne. On pourrait également l'utiliser en lecture des fichiers `pcap`.

Commencez par modifier le fichier `/etc/argus.conf`, pour spécifier la bonne interface de capture. Ensuite utilisez la commande `systemctl` pour démarrer ou redémarrer le service `argus-server`.

Argus enregistre ses résultats dans le fichier `/var/log/argus/argus.log` par défaut. Ce fichier est dans un format binaire et ne peut pas être lu avec des commandes pour lire du texte. Les commandes de lecture du fichier `argus.log` sont installées par le package `argus-client`, elles sont toutes préfixées par `ra`. C'est aussi le nom de la commande de base pour lire ce fichier. Pour lire le fichier produit par argus, vous pouvez donc faire la commande suivante :

```
ra -nr /var/log/argus/argus.log
```

Pouvez-vous observer la différence fondamentale de ces résultats par rapport à ce qui est enregistré par la commande `tcpdump` ?

Vous pouvez également ajouter un filtre BPF sur la ligne de commande, par exemple :

```
ra -nr /var/log/argus/argus.log - port 80
```

6 Bro

Lancer bro sur un fichier de capture et observez les informations remontées. Relancer bro mais cette fois en spécifiant le script `/usr/share/bro/policy/frameworks/files/extract-all-files.bro`.

```
bro -r <fichier_de_capture> /usr/share/bro/policy/frameworks/files/extract-all-files.bro
```

Variante avec un filtre BPF appliqué sur la capture :

```
bro -r <fichier_de_capture> -f 'port 80' /usr/share/bro/policy/frameworks/files/extract-all
    -files.bro
```