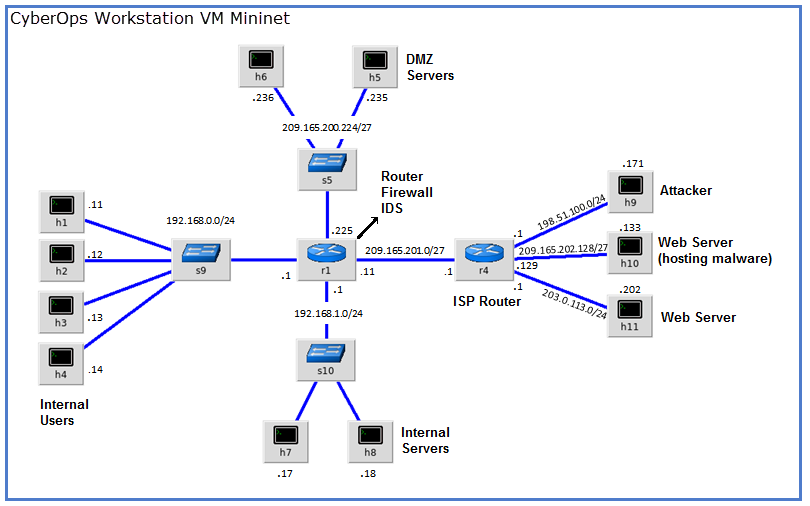
Travaux pratiques – Règles de Snort et de pare-feu

# Topologie



# Objectifs

Partie 1 : préparer l'environnement virtuel

Partie 2 : pare-feu et journaux IDS

Partie 3 : arrêter et supprimer le processus Mininet

# Contexte/scénario

Dans un réseau de production sécurisé, les alertes réseau sont générées par divers types d'appareils tels que des appliances de sécurité, des pare-feu, des systèmes de protection contre les intrusions, des routeurs, des commutateurs, des serveurs et autres. Le problème est que toutes les alertes ne sont pas similaires. Par exemple, les alertes générées par un serveur et les alertes générées par un pare-feu seront différentes sur le plan de leur contenu et de leur format.

Au cours de ces travaux pratiques, vous allez vous familiariser avec les règles de pare-feu et les signatures IDS.

# Ressources requises

* Poste de travail virtuel CyberOps
* Connexion Internet

**Remarque** : dans ces travaux pratiques, le poste de travail virtuel CyberOps contient l'environnement Mininet montré dans la topologie. Si une erreur mémoire est reçue lors d'une tentative d'exécution de commande, arrêtez tout, accédez aux paramètres de la machine virtuelle et augmentez la capacité de la mémoire. La taille par défaut est de 1 Go. Essayez avec 2 Go.

# Instructions

## Préparer l'environnement virtuel

* + - 1. Lancez **Oracle VirtualBox** et changez de **poste de travail virtuel CyberOps** pour le mode ponté, si nécessaire. Sélectionnez **Machine > Settings > Network**. Sous **Attached To**, sélectionnez **Bridged Adapter** (ou si vous utilisez le Wi-Fi avec un proxy, choisissez **NAT adapter**) et cliquez sur **OK**.
      2. Lancez le **poste de travail virtuel CyberOps**, ouvrez un terminal et configurez son réseau en exécutant le script **configure\_as\_dhcp.sh**.

Comme le script nécessite des privilèges de super-utilisateur, saisissez le mot de passe correspondant à l'utilisateur **analyst**.

[analyst@secOps ~]$ **sudo ./lab.support.files/scripts/configure\_as\_dhcp.sh**

[sudo] password for analyst:

[analyst@secOps ~]$

* + - 1. Utilisez la commande **ifconfig** pour vérifier que le **poste de travail virtuel CyberOps** a maintenant une adresse IP sur le réseau local. Vous pouvez également tester la connectivité à un serveur web public en envoyant une requête ping à www.cisco.com. Appuyez sur **Ctrl+ C** pour arrêter l'envoi de pings.

[analyst@secOps ~]$ **ping www.cisco.com**

PING e2867.dsca.akamaiedge.net (23.204.15.199) 56(84) bytes of data.

64 bytes from a23-204-15-199.deploy.static.akamaitechnologies.com (23.204.15.199): icmp\_seq=1 ttl=54 time=28.4 ms

64 bytes from a23-204-15-199.deploy.static.akamaitechnologies.com (23.204.15.199): icmp\_seq=2 ttl=54 time=35.5 ms

^C

--- e2867.dsca.akamaiedge.net ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1002ms

rtt min/avg/max/mdev = 28.446/32.020/35.595/3.578 ms

## Pare-feu et journaux IDS

Les pare-feu et les systèmes de détection des intrusions (IDS) sont souvent déployés pour automatiser partiellement la tâche de surveillance du trafic. Les pare-feu et les IDS appliquent des règles administratives au trafic entrant. Les pare-feu comparent généralement l'en-tête du paquet à une règle définie tandis que les IDS comparent souvent la charge utile du paquet à une règle définie. Étant donné que les pare-feu et les IDS appliquent des règles prédéfinies pour différentes parties du paquet IP, les structures des règles d'IDS et de pare-feu sont différentes.

Malgré ces différences, il existe certaines similitudes entre les composants des règles. Par exemple, les règles de pare-feu et d'IDS contiennent des composants correspondants et des composants d'action. Des mesures sont prises lorsqu'une correspondance est trouvée.

* **Composant correspondant** : spécifie les éléments de paquet intéressants tels que la source du paquet, la destination du paquet, les protocoles et les ports de la couche transport et les données incluses dans la charge utile du paquet.
* **Composant d'action** : indique ce qu'il faut faire avec ce paquet qui correspond à un composant tel que : accepter et transférer le paquet, rejeter le paquet ou envoyer le paquet à un ensemble de règles secondaires pour une inspection supplémentaire.

En général, les pare-feu rejettent les paquets par défaut. Il est ensuite nécessaire de spécifier manuellement quel trafic est autorisé. Cette conception a l'avantage de protéger le réseau contre les attaques et les protocoles inconnus. Sous cette conception, il arrive souvent de devoir consigner les événements relatifs aux paquets rejetés étant donné qu'ils n'ont pas été explicitement autorisés et qu'ils enfreignent les politiques de l'entreprise. Ces événements doivent être enregistrés pour une analyse ultérieure.

### Surveillance en temps réel des journaux de l'IDS

* + - 1. Depuis le **poste de travail virtuel CyberOps**, exécutez le script de démarrage de **mininet**.

[analyst@secOps ~]$ **sudo ./lab.support.files/scripts/cyberops\_extended\_topo\_no\_fw.py**

[sudo] password for analyst:

\*\*\* Adding controller

\*\*\* Add switches

\*\*\* Add hosts

\*\*\* Add links

\*\*\* Starting network

\*\*\* Configuring hosts

R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11

\*\*\* Starting controllers

\*\*\* Starting switches

\*\*\* Add routes

\*\*\* Post configure switches and hosts

\*\*\* À partir de l'interface de ligne de commande :

mininet>

L'invite **mininet** doit s'afficher ce qui indique que **mininet** est prêt à recevoir des commandes.

* + - 1. Depuis l'invite de **mininet**, ouvrez un interpréteur de commandes sur **R1** à l'aide de la commande suivante :

mininet> **xterm R1**

mininet>

#### Question :

L'interpréteur de commandes de **R1** s'ouvre dans une fenêtre de terminal avec le texte en noir et un fond blanc. Quel utilisateur est connecté à cet interpréteur de commandes ? Qu'est-ce que cela indique ?

* + - 1. Depuis l'interpréteur de commandes de **R1**, démarrez l'IDS Linux, Snort.

[root@secOps analyst]# **./lab.support.files/scripts/start\_snort.sh**

Running in IDS mode

--== Initializing Snort ==--

Initializing Output Plugins!

Initializing Preprocessors!

Initializing Plug-ins!

Parsing Rules file "/etc/snort/snort.conf"

<output omitted>

**Remarque :** vous ne verrez pas d'invite, car Snort est exécuté maintenant dans cette fenêtre. Si pour une raison quelconque, Snort cesse de fonctionner et si l'invite **[root@secOps analysts]#** s'affiche, exécutez de nouveau le script pour lancer Snort. Snort doit fonctionner afin de capturer des alertes plus tard dans ces travaux pratiques.

* + - 1. Depuis l'invite **mininet du poste de travail virtuel CyberOps**, ouvrez les interpréteurs de commandes pour les hôtes **H5** et **H10**.

mininet> **xterm H5**

mininet> **xterm H10**

mininet>

* + - 1. **H10** simule un serveur sur Internet qui héberge des malwares. Sur **H10**, exécutez le script **mal\_server\_start.sh** pour démarrer le serveur.

[root@secOps analyst]# .**/lab.support.files/scripts/mal\_server\_start.sh**

[root@secOps analyst]#

* + - 1. Sur **H10**, utilisez **netstat** avec les options **- tunpa** pour vérifier que le serveur web s'exécute. Lorsqu'elle est utilisée comme indiqué ci-dessous, la commande **netstat** répertorie tous les ports actuellement affectés aux services :

[root@secOps analyst]# **netstat -tunpa**

Connexions Internet actives (serveurs et connexions établies)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:6666 0.0.0.0:\* LISTEN 1839/nginx: master

[root@secOps analyst]#

Comme le montre la sortie ci-dessus, le serveur web léger **nginx** fonctionne et écoute les connexions sur le port TCP 6666.

* + - 1. La fenêtre du terminal **R1** montre une instance de Snort en cours d'exécution. Pour saisir plus de commandes sur **R1**, ouvrez un autre terminal **R1** en tapant de nouveau **xterm R1** dans la fenêtre de terminal du **poste de travail virtuel CyberOps**. Vous pouvez également réorganiser les fenêtres des terminaux pour voir chaque appareil et interagir avec chacun d'entre eux.
      2. Dans le nouvel onglet du terminal **R1**, exécutez la commande **tail** avec l'option **-f** pour surveiller le fichier **/var/log/snort/alert** en temps réel. Ce fichier contient la configuration de Snort pour enregistrer les alertes.

[root@sec0ps analyst]# **tail -f /var/log/snort/alert**

Étant donné qu'aucune alerte n'a été enregistrée, le journal doit être vide. Toutefois, si vous avez précédemment effectué ces travaux pratiques, d'anciennes alertes peuvent être indiquées. Dans les deux cas, vous ne recevrez pas d'invite après avoir tapé cette commande. Cette fenêtre affiche les alertes à mesure qu'elles se produisent.

* + - 1. Depuis **H5**, utilisez la commande **wget** pour télécharger un fichier nommé **W32.Nimda.Amm.exe**. Conçue pour télécharger du contenu via HTTP, la commande **wget** est un excellent outil pour télécharger des fichiers depuis des serveurs web directement à partir de la ligne de commande.

[root@secOps analyst]# **wget 209.165.202.133:6666/W32.Nimda.Amm.exe**

--2017-04-28 17:00:04-- http://209.165.202.133:6666/W32.Nimda.Amm.exe

Connecting to 209.165.202.133:6666… connected.

HTTP request sent, awaiting response… 200 OK

Length: 345088 (337K) [application/octet-stream]

Saving to: 'W32.Nimda.Amm.exe'

W32.Nimda.Amm.exe 100%[==========================================>] 337.00K --.-KB/s in 0.02s

2017-04-28 17:00:04 (16.4 MB/s) - 'W32.Nimda.Amm.exe' saved [345088/345088]

[root@secOps analyst]#

#### Question :

Quel port est utilisé pour communiquer avec le serveur web hébergeant des malwares ? Qu'est-ce que cela indique ?

Le fichier a-t-il été entièrement téléchargé ?

L'IDS a-t-il généré des alertes concernant le téléchargement du fichier ?

* + - 1. Comme le fichier malveillant transitait par **R1**, l'IDS Snort a inspecté sa charge utile. La charge utile correspondait à au moins une des signatures configurées dans Snort et a déclenché une alerte sur la deuxième fenêtre du terminal **R1** (l'onglet où la commande **tail -f** est exécutée). L'entrée de l'alerte est présentée ci-dessous. Votre horodatage sera différent :

04/28-17:00:04.092153 [\*\*] [1:1000003:0] Malicious Server Hit! [\*\*] [Priority: 0] {TCP} 209.165.200.235:34484 -> 209.165.202.133:6666

#### Questions :

En étudiant l'alerte présentée ci-dessus, trouvez quelles étaient les adresses IPv4 source et de destination utilisées dans le cadre de la transaction.

En étudiant l'alerte présentée ci-dessus, trouvez quels étaient les ports source et de destination utilisés dans le cadre de la transaction.

En étudiant l'alerte présentée ci-dessus, trouvez quand a eu lieu le téléchargement.

En étudiant l'alerte présentée ci-dessus, trouvez quel était le message enregistré par la signature de l'IDS.

Sur **H5**, utilisez la commande **tcpdump** pour capturer l'événement et télécharger de nouveau le fichier contenant le malware afin de capturer la transaction. Lancez la commande suivante pour démarrer la capture des paquets :

[root@secOps analyst]# **tcpdump –i H5-eth0 –w nimda.download.pcap &**

[1] 5633

[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

La commande ci-dessus indique à tcpdump de capturer les paquets sur l'interface **H5-eth0** et d'enregistrer la capture dans un fichier nommé **nimda.download.pcap**.

Le symbole **&** à la fin indique à l'interpréteur de commandes d'exécuter **tcpdump** en arrière-plan. Sans ce symbole, la commande **tcpdump** rendrait le terminal inutilisable pendant son fonctionnement. Notez **[1] 5633**, qui indique qu'un processus, dont l'ID (PID) est 5366, a été envoyé à l'arrière-plan. Votre PID sera très probablement différent.

* + - 1. Appuyez sur **ENTRÉE** à plusieurs reprises pour reprendre le contrôle de l'interpréteur de commandes tandis que **tcpdump** s'exécute en arrière-plan.
      2. Maintenant que la commande **tcpdump** capture les paquets, téléchargez de nouveau le malware. Sur **H5**, réexécutez la commande ou utilisez la flèche vers le haut pour la rappeler à partir de l'utilitaire d'historique des commandes.

[root@secOps analyst]# **wget 209.165.202.133:6666/W32.Nimda.Amm.exe**

--2017-05-02 10:26:50-- http://209.165.202.133:6666/W32.Nimda.Amm.exe

Connecting to 209.165.202.133:6666… connected.

HTTP request sent, awaiting response… 200 OK

Length: 345088 (337K) [application/octet-stream]

Saving to: 'W32.Nimda.Amm.exe'

W32.Nimda.Amm.exe 100%[===================>] 337.00K --.-KB/s in 0.003s

02/05/2017 10:26:50 (105 MB/s) - 'W32.Nimda.Amm.exe' saved [345088/345088]

* + - 1. Arrêtez la capture en exécutant **tcpdump** au premier plan avec la commande **fg**. Étant donné que **tcpdump** est le seul processus à avoir été envoyé en arrière-plan, il n'est pas nécessaire de spécifier son PID. Arrêtez le processus **tcpdump** avec **Ctrl+C**. Le processus **tcpdump** s'arrête et affiche un résumé de la capture. Le nombre de paquets peut être différent pour votre capture.

[root@secOps analyst]# **fg**

tcpdump -i h5-eth0 -w nimda.download.pcap

^C316 packets captured

316 packets received by filter

0 packets dropped by kernel

[root@secOps analyst]#

* + - 1. Sur **H5**, utilisez la commande **ls** pour vérifier que le fichier pcap a été en fait enregistré sur disque et que sa taille est supérieure à zéro :

[root@secOps analyst]# **ls -l**

total 1400

drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop

drwx------ 3 analyst analyst 4096 Jul 14 11:28 Downloads

drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files

-rw-r--r-- 1 root root 371784 Aug 17 14:48 nimda.download.pcap

drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second\_drive

-rw-r--r-- 1 root root 345088 Apr 14 15:17 W32.Nimda.Amm.exe

-rw-r--r-- 1 root root 345088 Apr 14 15:17 W32.Nimda.Amm.exe.1

[root@secOps analyst]#

**Remarque** : votre liste d'annuaire peut être composée de fichiers différents, mais le fichier **nimda.download.pcap** doit toujours y figurer.

#### Question :

Comment ce fichier PCAP peut-il être utile à l'analyste en sécurité ?

**Remarque** : l'analyse du fichier PCAP sera effectuée au cours d'un autre TP.

### Régler les règles de pare-feu à partir des alertes IDS

Au cours de l'étape 1, vous avez démarré un serveur malveillant sur Internet. Pour empêcher d'autres utilisateurs d'accéder à ce serveur, il est recommandé de le bloquer dans le pare-feu de périmètre.

Dans la topologie utilisée ici, **R1** exécute non seulement un IDS, mais aussi un pare-feu basé sur Linux très populaire appelé **iptables**. Au cours de cette étape, vous allez bloquer le trafic vers le serveur malveillant identifié à l'étape 1 en modifiant les règles de pare-feu actuellement activées sur **R1**.

**Remarque** : même si **iptables** n'est pas abordé en détail dans ce cours, la structure de base des règles et de la logique de **iptables** est assez simple.

Le pare-feu **iptables** utilise les concepts de *chaînes* et de *règles* pour filtrer le trafic.

Le trafic entrant dans le pare-feu et destiné au pare-feu lui-même est géré par la chaîne **INPUT**. Il peut s'agit par exemple de paquets ping provenant d'appareils sur un réseau et envoyés à l'une des interfaces du pare-feu.

Le trafic provenant du pare-feu lui-même et destiné à d'autres appareils est géré par la chaîne **OUTPUT**. Il peut s'agir de réponses ping générées par le pare-feu lui-même.

Le trafic provenant d'autres appareils et passant par le pare-feu est géré par la chaîne **FORWARD**. Il peut s'agir par exemple de paquets acheminés par le pare-feu.

Chaque chaîne peut être composée de son propre ensemble indépendant de règles spécifiant comment le trafic doit être filtré. Le nombre de règles par chaîne n'est pas limité. Il est même possible qu'une chaîne ne contienne aucune règle du tout.

Les règles sont créées pour vérifier les caractéristiques spécifiques des paquets, ce qui permet aux administrateurs de créer des filtres très complets. Si un paquet ne correspond pas à une règle, le pare-feu passe à la règle suivante et refait une vérification. Si une correspondance est trouvée, le pare-feu applique l'action définie dans la règle correspondante. Si après avoir vérifié toutes les règles dans une chaîne, aucune correspondance n'est trouvée, le pare-feu effectue l'action spécifiée par la politique de la chaîne, en général arrêter le paquet ou le laisser poursuivre sa route.

* + - 1. Sur le **poste de travail virtuel CyberOps**, lancez une troisième fenêtre du terminal R1.

mininet > **xterm R1**

* + - 1. Sur la nouvelle fenêtre du terminal **R1**, utilisez la commande **iptables** pour répertorier les chaînes et les règles en cours d'utilisation :

[root@secOps ~]# **iptables -L -v**

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 6 packets, 504 bytes)

pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

[root@secOps ~]#

#### Question :

Quelles chaînes sont actuellement utilisées par **R1** ?

* + - 1. Les connexions au serveur malveillant génèrent des paquets qui doivent transiter par le pare-feu **iptables** sur **R1**. Les paquets qui transitent par le pare-feu sont traités par la règle FORWARD. C'est donc la chaîne qui recevra la règle de blocage. Pour empêcher les ordinateurs de se connecter au serveur malveillant identifié à l'étape 1, ajoutez la règle suivante à la chaîne FORWARD sur **R1** :

[root@secOps ~]# **iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -j DROP**

[root@secOps ~]#

Où :

* **-I FORWARD** : insère une nouvelle règle dans la chaîne FORWARD.
* **tcp -p** : spécifie le protocole TCP.
* **-d 209.165.202.133** : spécifie la destination du paquet
* **--dport 6666** : spécifie le port de destination
* **-j DROP** : définit l'action « rejeter ».
  + - 1. Utilisez de nouveau la commande **iptables** pour que la règle soit ajoutée à la chaîne FORWARD. Il est possible que plusieurs minutes soient nécessaires au poste de travail virtuel CyberOps pour générer la sortie :

[root@secOps analyst]# **iptables -L -v**

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

0 0 DROP tcp -- any any anywhere 209.165.202.133 tcp dpt:6666

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

[root@secOps analyst]#

* + - 1. Sur **H5**, essayez de télécharger de nouveau le fichier :

[root@secOps analyst]# **wget 209.165.202.133:6666/W32.Nimda.Amm.exe**

--2017-05-01 14:42:37-- http://209.165.202.133:6666/W32.Nimda.Amm.exe

Connecting to 209.165.202.133:6666… failed: Connection timed out.

Retrying.

--2017-05-01 14:44:47-- (try: 2) http://209.165.202.133:6666/W32.Nimda.Amm.exe

Connecting to 209.165.202.133:6666… failed: Connection timed out.

Retrying.

Saisissez **Ctrl+C** pour annuler le téléchargement, si nécessaire.

#### Questions :

Le téléchargement a-t-il réussi cette fois ? Expliquez votre réponse.

Saisissez vos réponses ici

Quelle approche plus agressive, mais également plus appropriée pourrait être utilisée pour bloquer le serveur malveillant ?

## Arrêter et supprimer le processus Mininet

* + - 1. Accédez au terminal utilisé pour démarrer Mininet. Pour arrêter le processus Mininet, saisissez **quit** dans la fenêtre principale du poste de travail virtuel CyberOps.
      2. Une fois Mininet arrêté, supprimez les processus démarrés par Mininet. Saisissez le mot de passe **cyberops** lorsque vous y êtes invité.

[analyst@secOps scripts]$ **sudo mn –c**

[sudo] password for analyst:

Fin du document