YNOV AIX-EN-PROVENCE

TP ICMP

Sécurité des SI

RIGONNAX Mickael – MONIER Sebastien

25/04/2018

Table des matières

1. Objectif

L'objectif de ce TP est de comprendre le fonctionnement du protocole ICMP, connaitre ses faiblesses et savoir comment les exploiter.

Ce TP se déroule en deux parties :

- Attaque ICMP Smurf : cette attaque consiste à flooder la connexion d'un utilisateur ou d'un service avec des requêtes ICMP afin de réduire sa connexion voire la saturer et empêcher l'accès au réseau.
- Attaque ICMP Cover channel : cette attaque consiste à faire passer des informations sur le réseau en les dissimulant grâce au protocole ICMP.

2. Analyse

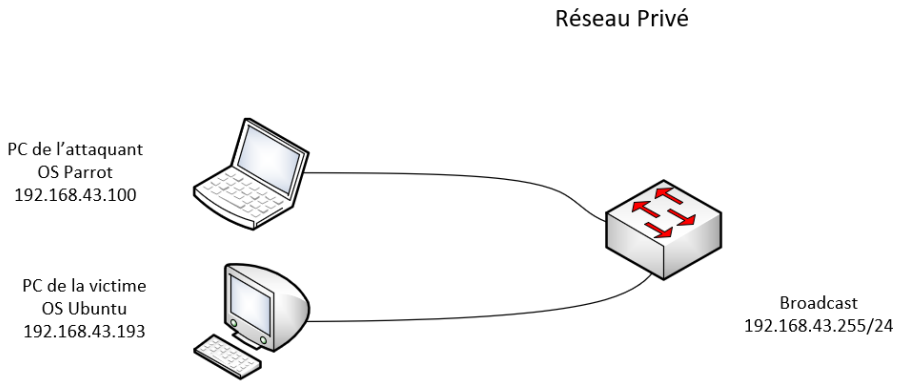
a. Attaque ICMP Smurf

Contexte

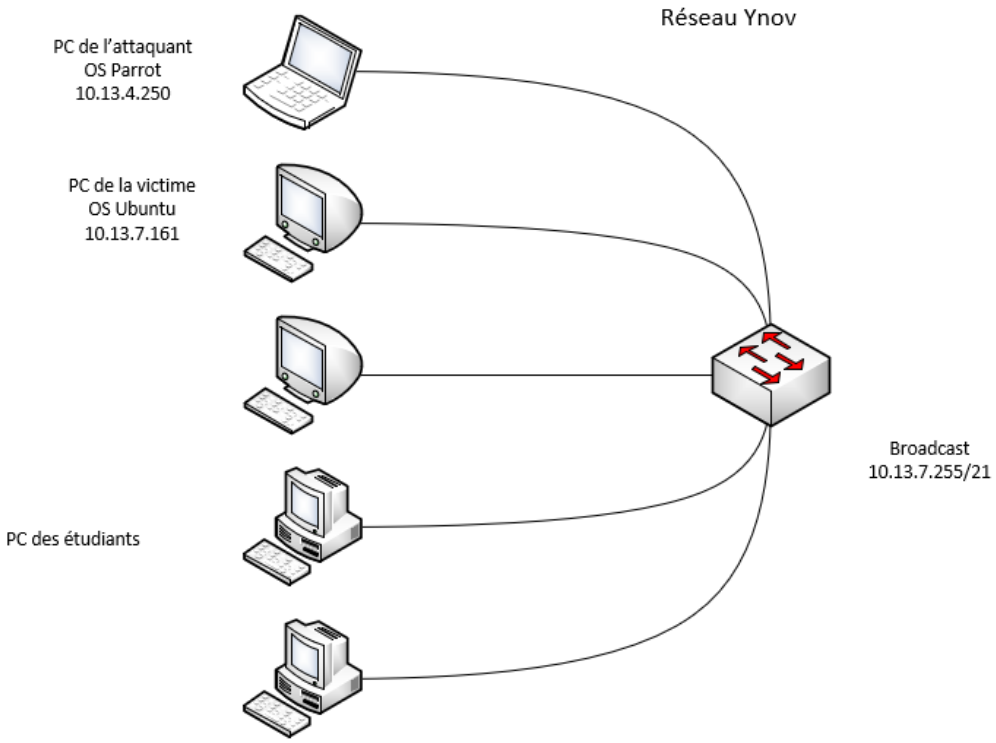
Ce TP a été réalisé en deux temps, une première fois sur un réseau privé avec un partage de connexion d'un smartphone en phase de test. Et une seconde fois en condition réelle sur le réseau de l'école Ynov d'Aix-en-Provence.

Voici les schémas correspondant aux deux infrastructures utilisées.

Réseau privé :



Réseau Ynov :



Dans chacun des TP deux machines ont été utilisée, une machine attaquante sous Parrot OS, et une seconde victime sous Ubuntu.

Problématique

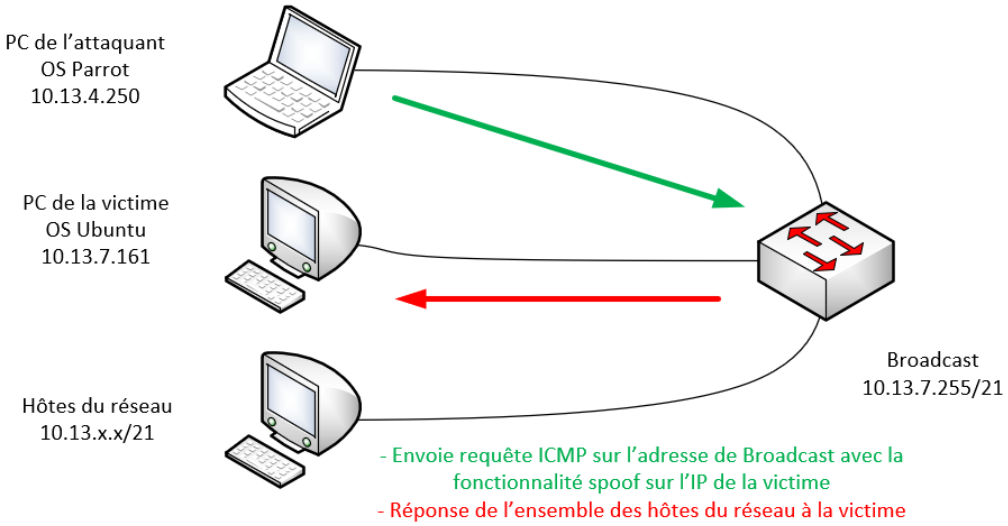
Pour ces deux réalisations, la problématique est la suivante : Comment mettre une machine en déni de service avec ICMP ?

Réalisation

Le but de ce TP est de créer un déni de service sur le PC de la victime, pour ce faire nous utilisons le protocole ICMP. Le fonctionnement est très simple, il suffit d'envoyer des requêtes ICMP sur l'adresse de broadcast du réseau en se faisant passer pour la victime. L'envoie d'une requête sur le broadcast va entrainer une réponse de l'ensemble des utilisateurs du réseau vers la machine victime, qui va être surchargée de paquet.

En plus de ce mécanisme nous avons utilisé l'outil Hping3 qui nous permet d'envoyer un grand nombre de requête, de spoofer l'adresse de la victime et d'envoyer des paquets lourds et difficiles à traiter.

Fonctionnement schématisé :



Nous avons découpé cette attaque en deux parties. Dans la première partie un poste attaquant envoi des requêtes ICMP directement à un poste cible, cela n'empêchait pas l'accès au réseau mais en ralentissait grandement l'utilisation.

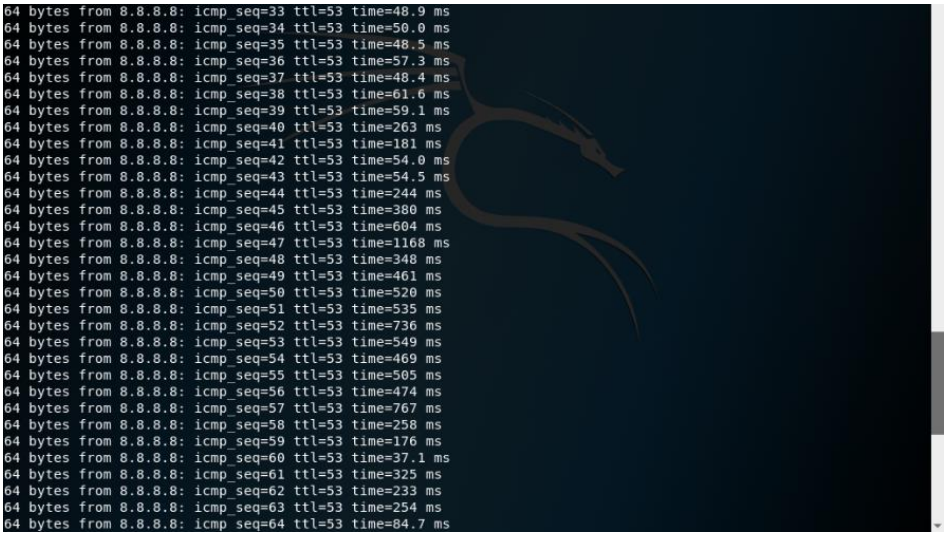
Pour cette partie nous utilisons un téléphone mobile afin d'être dans un réseau isolé. Le plan d'adressage de ce réseau est en 192.168.43.0/24 et l'IP du poste cible est 192.168.43.193

L'attaquant utilise l'utilitaire hping3 et exécute la commande suivante :

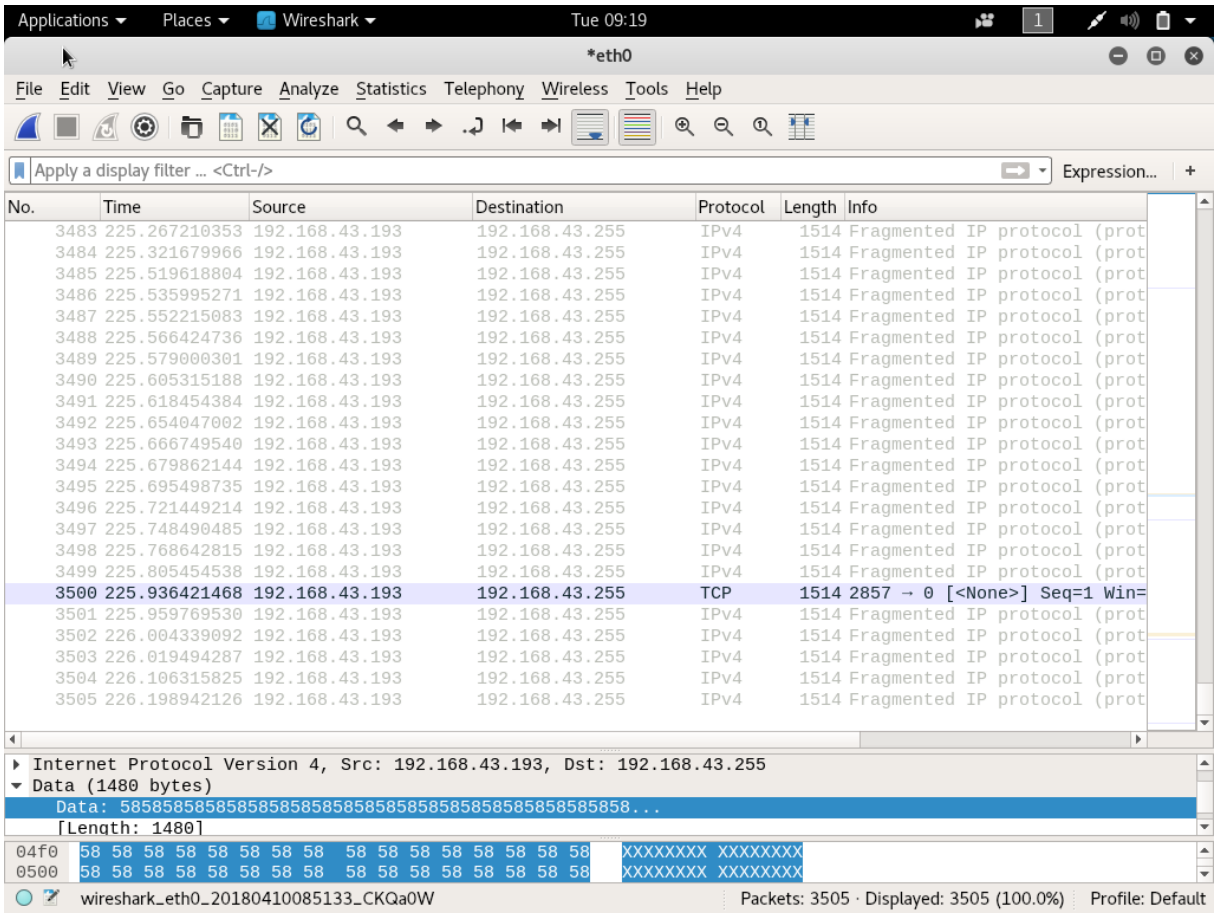
```
┌──[root@parrot]─[/home/mrigonnaux]
└─── #hping3 192.168.43.255 -a 192.168.43.193 -d 100000000
HPING 192.168.43.255 (wlan0 192.168.43.255): NO FLAGS are set, 40 headers + 5760
0 data bytes
```

Dans cette commande nous pouvons voir la destination de l'adresse qui est celle de broadcast du réseau du smartphone. Le -a suivi de l'IP correspond à l'option spoof, dans notre cas, celle du PC de la victime. Pour finir le -d indique la taille des paquets à envoyer en bytes.

Après avoir lancé l'attaque, la victime a toujours accès à internet mais son temps de réponse augmente fortement.



Sur Wireshark nous avons aussi remarqué que les trames partent en continu de la victime vers le broadcast :



L'attaque ne permet pas de bloquer totalement l'accès du poste de la victime, car il y a seulement deux hôtes dans ce réseau.

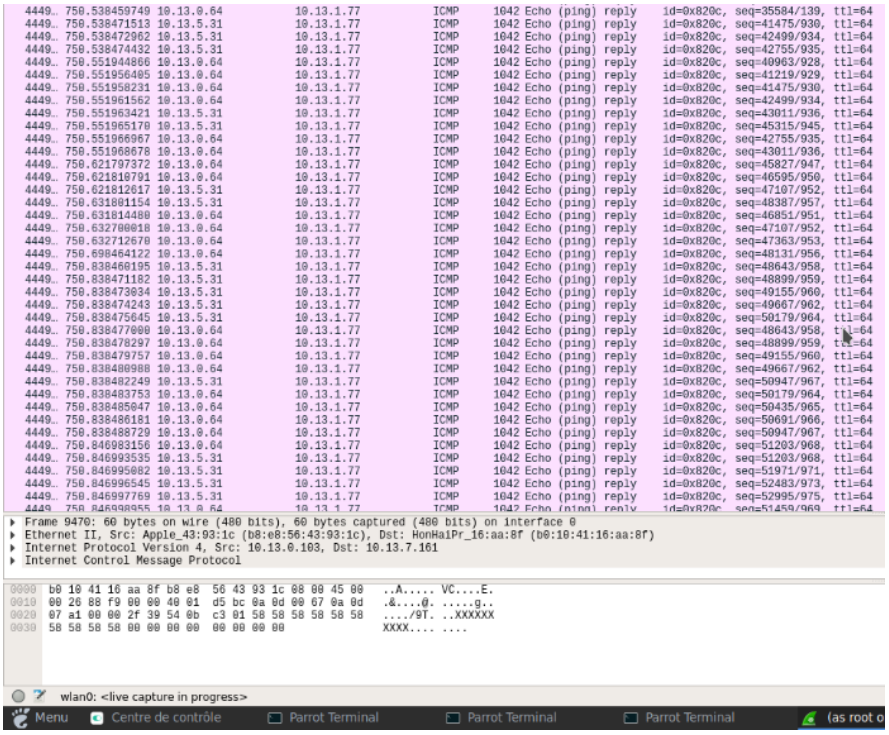
Afin de bloquer totalement l'accès d'un poste nous avons reproduit exactement la même attaque mais cette fois-ci sur le réseau de l'école Ynov. Le réseau comporte au moins un cinquantaines d'étudiants.

Le même utilitaire est utilisé dans cette partie, dans ce cas la victime est le poste qui a pour IP 10.13.7.161

```
┌─[✗]─[root@parrot]─[/home/mrigonnaux]
└──╼ #hping3 10.13.7.255 --faster --icmp -a 10.13.7.161 -d 10000
HPING 10.13.7.255 (wlan0 10.13.7.255): icmp mode set, 28 headers + 10000 data bytes
```

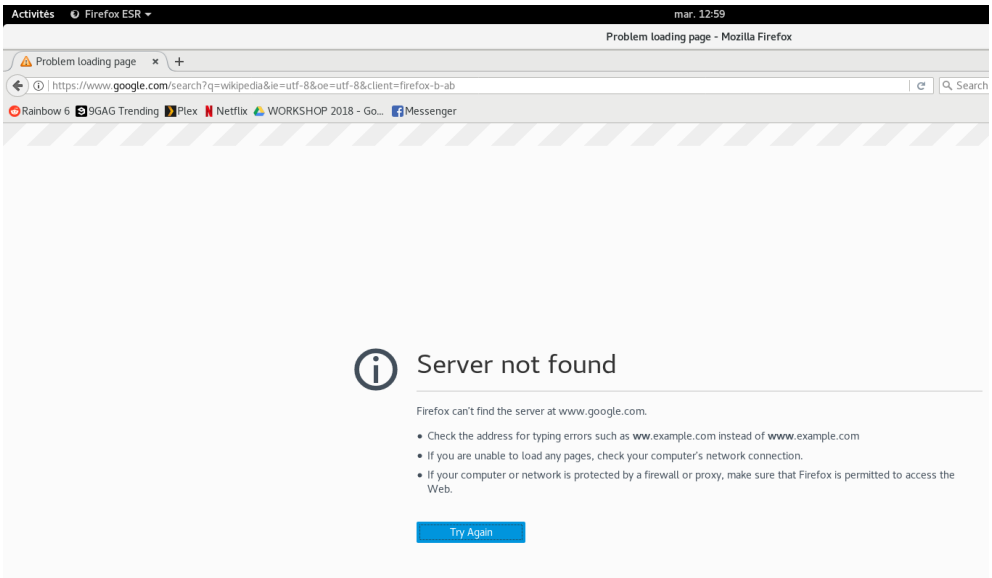
Après avoir lancé l'attaque nous observons l'impact dans WireShark, et nous découvrons que la victime reçoit des paquets ICMP de l'ensemble des machines du réseau.

Capture Wireshark :



Afin de vérifier si l'attaque a été un succès nous tentons de rejoindre le site google.fr depuis le poste de la victime, et effectivement l'accès au réseau est impossible. Le nombre de paquets envoyés à la victime est tellement important que la carte réseau ne peut plus les gérer et met donc le poste en déni de service.

Recherche internet :



b. Attaque ICMP Cover Channel Ptunnel

Contexte

Ce TP a été réalisé sur le réseau de l'école Ynov d'Aix-en-Provence. Plusieurs machines ont été manipulées, comme le poste de l'attaquant qui est une machine sous Kali Linux, le firewall de l'école et une machine qui sert de serveur Proxy hébergée dans le cloud.

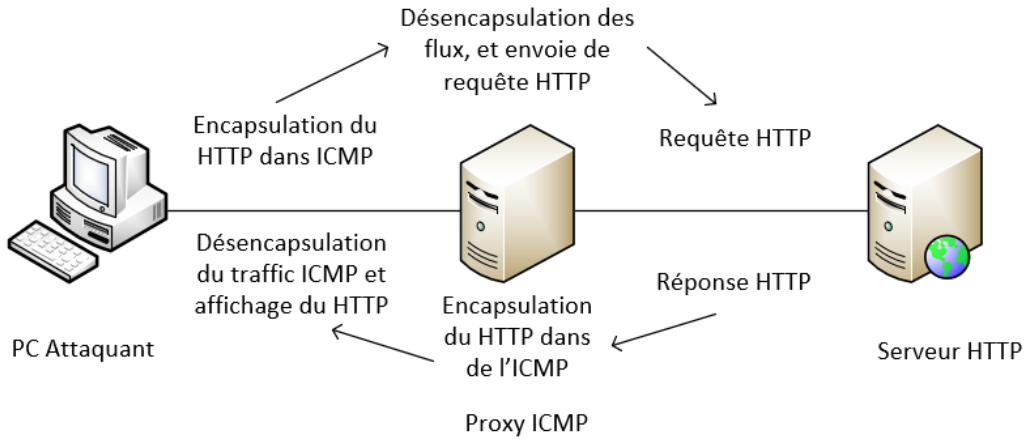
Problématique

La problématique est la suivante, comment réussir à communiquer avec un serveur bloqué par le Proxy/Firewall ?

Pour répondre à cette question, nous avons installé et utilisé PTunnel qui est un proxy ICMP et qui permet d'encapsuler le trafic HTTP dans des paquets ICMP et donc d'être invisible pour les éléments de sécurité.

Fonctionnement

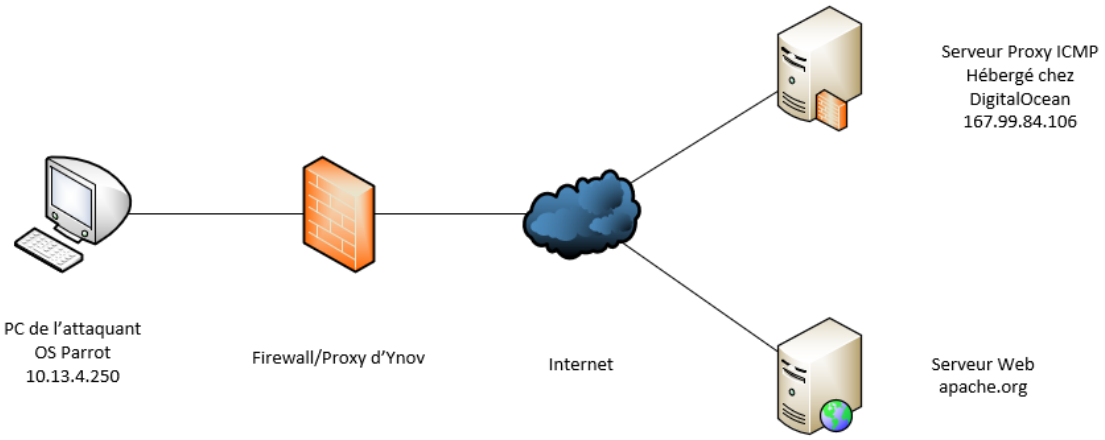
L'outil PTunnel fonctionne de la manière suivante :



Dans le cadre de ce TP nous avons voulu accéder à des sites bloqués par le pare-feu d'Ynov, pour cela nous avons travaillé en collaboration avec le BSI (bureau des systèmes d'information).

Pour réaliser cette action nous avons utilisé l'outil Ptunnel qui permet d'encapsuler les requêtes http dans de l'ICMP. Il est basé sur un modèle client-serveur, dans notre cas le client est dans le réseau de l'école Ynov et le serveur est situé dans le cloud chez Digital Océan.

L'infrastructure est la suivante :



Le but de ce TP est donc d'accéder à un site qui est bloqué par le pare-feu, nous tenterons d'abord un accès sur le site http://apache.org qui a préalablement été bloqué par le BSI d'Ynov.

Nous avons d'abord installé et lancé PTunnel sur un serveur Debian 9 herbergé chez dans le cloud :

```
root@debian-s-1vcpu-2gb-lon1-01:~# ptunnel -c eth0 -x Ynov2018!
[inf]: Starting ptunnel v 0.72.
[inf]: (c) 2004-2011 Daniel Stoedle, <daniels@cs.uit.no>
[inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>
[inf]: Forwarding incoming ping packets over TCP.
[inf]: Initializing pcap.
[inf]: Ping proxy is listening in privileged mode.
[inf]: Incoming tunnel request from 195.200.178.226.
[inf]: Starting new session to 52.41.96.17:80 with ID 44690
[err]: Dropping duplicate proxy session request.
[err]: Dropping duplicate proxy session request.
[err]: Dropping duplicate proxy session request.
[err]: Dropping duplicate proxy session request.
[err]: Dropping duplicate proxy session request.
[inf]: Received session close from remote peer.
[inf]: Received session close from remote peer.
[inf]: Received session close from remote peer.
[inf]: Received session close from remote peer.
```

L'outil se lance tout simplement avec la commande avec la commande « ptunnel », le -c correspond à l'interface à utiliser et le -x correspond au mot de passe à entrer pour s'authentifier côté client.

Il reste maintenant à lancer le client PTunnel côté client, dans notre cas sur une machine Kali Linux :

La même installation de PTunnel est nécessaire sur le client, il est cependant installé automatiquement sur Kali Linux. Les différentes options utilisées sont -p pour indiquer l'adresse IP de notre serveur distant, -lp pour indiquer le port d'écoute de notre client. C'est sur ce port où nous allons nous connecter via le navigateur Web. L'option -da indique l'URL à accéder, -dp pour le port distant et enfin -x pour le mot de passe nécessaire à l'authentification.

Une fois la commande lancée, le tunnel ICMP est créé et nous pouvons accéder à notre site Web bloqué sur localhost:4444 dans un navigateur.

Le site apache.org est donc accessible via le tunnel ICMP, même si ce dernier est bloqué par le pare-feu, les paquets http pour apache.org sont transparents pour le pare-feu car ils sont à l'intérieur des paquets ICMP.

Par contre l'utilitaire PTunnel devient vieillissant et ne fonctionne pas avec l'ensemble des sites, dans notre cas nous n'avons pas réussi à le faire fonctionner avec des sites en https et/ou qui utilisent flash player.

Mais les flux traversent bien le firewall même pour des sites bloqués comme Netflix, qui nous renvoie un message d'erreur :

La même commande a été utilisée pour utiliser le site Netflix, avec seulement le port qui a changé, le 443 à la place du port 80.

c. Attaque ICMP Covert Channels Dissimulation

Contexte

Ce TP a été réalisé sur le réseau de l'école Ynov d'Aix-en-Provence. 2 machines sous Kali Linux ont été nécessaires à sa réalisation.

Problématique

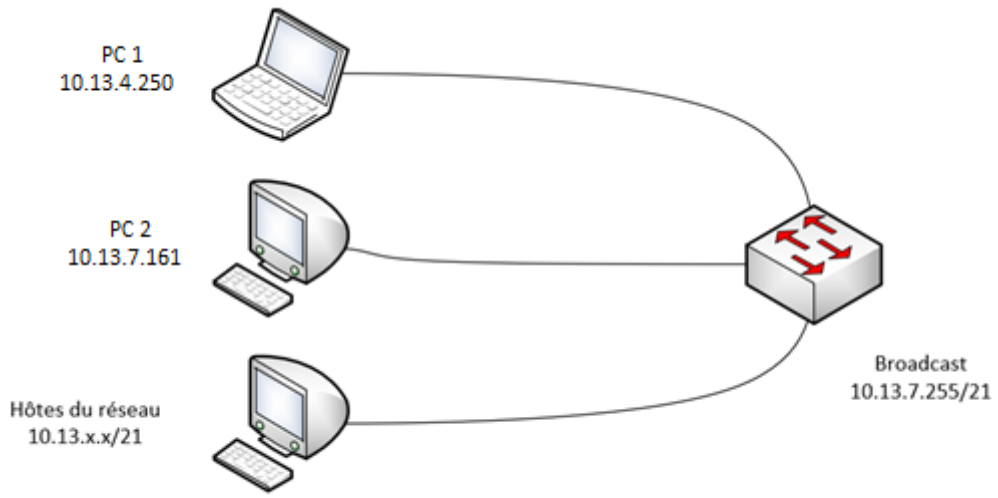
Comment administrer un outil à distance, ou communiquer de façon cachée avec ICMP ?

Problématique nous allons utiliser l'outil Hping qui permet de manipuler les trames ICMP.

Fonctionnement

Le principe de l'attaque est le suivant, nous avons plusieurs machines sur un réseau et leurs communications sont bloquées par un pare-feu ou autres. La communication peut avoir lieu pour plusieurs répons, communication entre 2 personnes, administration d'un malware, etc.

Voici l'architecture dans laquelle nous avons effectué cette attaque :



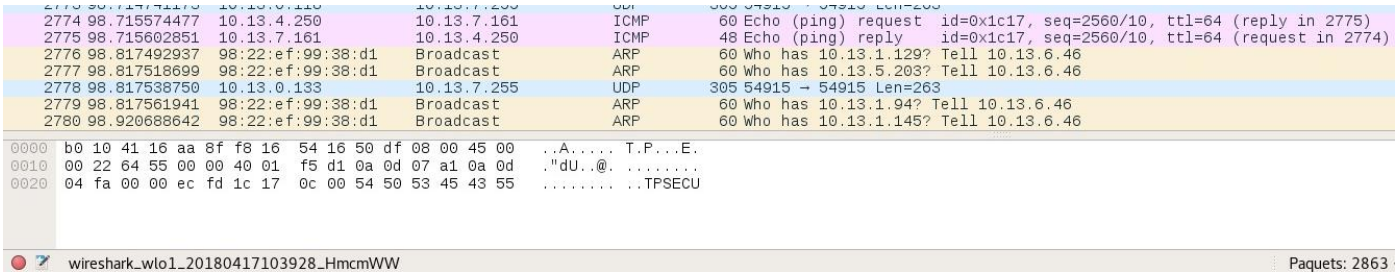
Deux machines sont présentes dans le réseau de l'école doivent pouvoir communiquer en étant transparente sur le réseau de l'école, dans notre cas deux machines Kali Linux.

La communication sera faite via de simples requêtes ICMP sur le réseau car l'ICMP n'est que très rarement bloqué sur un réseau. Pour ce faire nous avons utilisé encore une fois l'outil hping3 qui permet de stocker des données dans des trames ICMP (commande, message, etc.).

Son utilisation s'est faite de la manière suivante :

- Depuis le PC 1 nous avons envoyé la requête suivante :
  - Hping3 -1 –c 1 10.13.4.250 –e « TPSECU »

Cette commande va donc envoyer une seule requête grâce à l'argument –c 1 qui correspond au « count », en ICMP avec le -1 qui est le choix du protocole, et le plus important elle va placer notre chaîne de caractères « TPSECU » dans l'emplacement data de la requête ICMP. L'IP rentrée est l'IP de la machine qui recevra cette requête, voici le résultat avec une capture de trame sur cette même machine :



Nous voyons très bien ici l'arrivée du paquet ICMP qui contient bien dans son paquet notre chaîne de caractères. Cette méthode peut être utilisée également pour l'administration d'outil, de l'injection de commande, etc.

3. Conclusion

Pour conclure sur ce TP, nous allons d'abord faire une analyse rapide du protocole ICMP et pour finir les protections que nous pouvons mettre en œuvre.

Ce protocole est indispensable au bon fonctionnement du réseau car il gère à lui seul une grande partie des remontées d'erreur (problème de réseau, configuration, etc.). Il est donc impossible de bloquer totalement ce protocole sous peine d'avoir des gros ralentissements et problème sur le réseau.

Comme vu tout au long de ce TP, ICMP présente de multiples failles, elles permettent de réaliser des communications cachées et des dénis de service par exemple. Pour sécuriser l'utilisation de ce protocole ils existent plusieurs moyens et outils comme :

- Le blocage des requêtes ICMP sur l'IP de broadcast
- Utiliser un HIDS (Wazuh par exemple) qui permet de déceler un nombre de requête anormale et de les bloquer automatiquement
- Le même HIDS peut également intégrer une interface pour suivre en temps réel les alertes liées à ce protocole
- La mise en place d'un NIDS (Snort, Suricata) permet aussi de protéger une partie de son réseau des attaques ICMP