

École Publique d'Ingénieurs en 3 ans

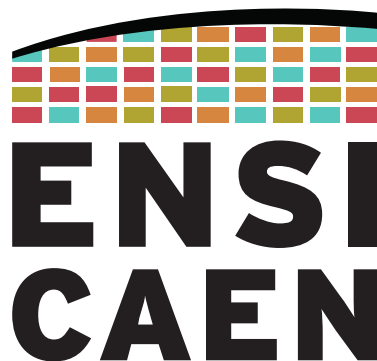
Rapport

MANUEL UTILISATEUR : PROJET ATTAQUES SUR MACHINES VIRTUELLES

le 14 avril 2021,

Boris ANODEAU
Flavien HANICOTTE
Marine HERVET

Tuteur Projet : Lyes Khoukhi



ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

www.ensicaen.fr

TABLE DES MATIÈRES

1. APPLICATION PÉDAGOGIQUE AVEC INTERFACE GRAPHIQUE.....	4
1.1. Version de Python.....	4
1.2. Installation des Modules.....	4
1.3. Fonctionnalités de l'Application.....	4
1.3.1. Fenêtre de Lancement.....	4
1.3.2. Information.....	5
1.3.3. Extraction d'images à partir d'un fichier de capture.....	6
1.3.4. Lecture d'un fichier audio sur la plateforme.....	7
2. MACHINES VIRTUELLES ET CONFIGURATION RÉSEAU.....	8
2.1. Installation des logiciels GNS3 et VirtualBox.....	8
2.2. Installation et lien de la VM GNS3.....	8
2.3. Setup GNS3.....	8
2.4. Configuration des différentes machines virtuelles.....	8
2.5. Création du projet GNS3 et Importation des VMs.....	9
2.6. Setup Réseau de Base.....	10
2.7. Test de Fonctionnalité du Réseau de Base.....	11
2.8. Gestion des Fichiers et Dossiers partagés.....	11
3. ATTAQUE ARP POISONING.....	11
3.1. Mise en place des outils de l'attaque.....	11
3.2. Exécution de l'Attaque.....	12
3.3. Vérification de l'Attaque.....	12
4. ATTAQUE DHCP SPOOFING.....	12
4.1. Mise en place des outils de l'attaque.....	12
4.2. Exécution de l'Attaque.....	13
4.3. Vérification de l'Attaque.....	13
5. EXTRACTION D'AUDIO À PARTIR D'UN FICHIER PCAP.....	13
5.1. Version de Python.....	13
5.2. Installation des Modules.....	13
5.2.1. Pyshark.....	13
5.2.2. SoX - Sound eXchange.....	13
5.2.3. Modules subprocess, os et sys.....	14
5.3. Comment utiliser le script.....	14
5.3.1. Ressources disponibles.....	14

5.3.2. Commande pour extraire le fichier audio de Forensic.cap.....	14
5.3.3. Arguments Obligatoires.....	15
5.3.4. Lecture du Fichier Audio.....	15
5.3.5. Fichier sans contenu Audio.....	15
5.3.6. Commandes Incorrectes.....	15
5.3.7. Comparer l'audio extrait et celui généré par Wireshark.....	16
6. EXTENSIONS POSSIBLES ENVISAGÉES.....	18
6.1. Réseau : Lien avec le PC Host.....	18

TABLE DES FIGURES

Figure 1: Fenêtre de lancement.....	4
Figure 2: Fenêtre d'Information.....	5
Figure 3: Table de Correspondances à la suite d'un attaque ARP Poisoning.....	5
Figure 4: Trames récupérée lors de l'usurpation de l'identité du serveur DHCP.....	6
Figure 5: Extraction des images.....	6
Figure 6: Images extraites.....	7
Figure 7: Lecture d'un fichier d'écoute sur VoIP.....	7
Figure 8: Réseau utilisé dans le cadre de la plateforme.....	10
Figure 9: Version de l'interpréteur Python installé.....	13
Figure 10: Fichiers PCAP fournis.....	14
Figure 11: Commande correcte d'extraction.....	14
Figure 12: Erreur lorsque qu'aucun fichier audio n'est trouvé.....	15
Figure 13: Nombre incorrect d'arguments.....	16
Figure 14: Erreur dans le type des arguments.....	16
Figure 15: Accès au Flux RTP via Wireshark.....	16
Figure 16: Analyse des fichiers audio depuis Wireshark.....	17
Figure 17: Informations sur l'audio et lecture.....	17

MANUEL UTILISATEUR

1. Application pédagogique avec interface graphique

1.1. Version de Python

Cette application a été développée sous Python 3.8. Il est recommandé d'utiliser cette version au minimum. Si votre version est trop ancienne, vous pouvez télécharger une version plus récente [ici](#).

1.2. Installation des Modules

Pour utiliser l'application il faudra installer les modules Python suivants :

```
pip install regex
pip install scapy
pip install audioplayer
pip install openpyxl
```

Pour l'aspect graphique, l'application utilise *tkinter*, module directement intégré dans Python 3 et qui n'a donc pas besoin d'être installé.

1.3. Fonctionnalités de l'Application

1.3.1. Fenêtre de Lancement

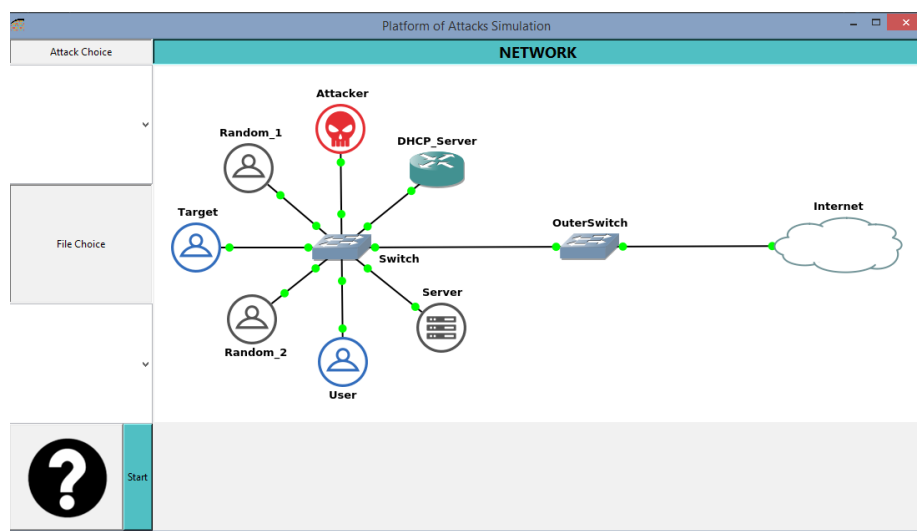


Figure 1: Fenêtre de lancement

1.3.2. Information

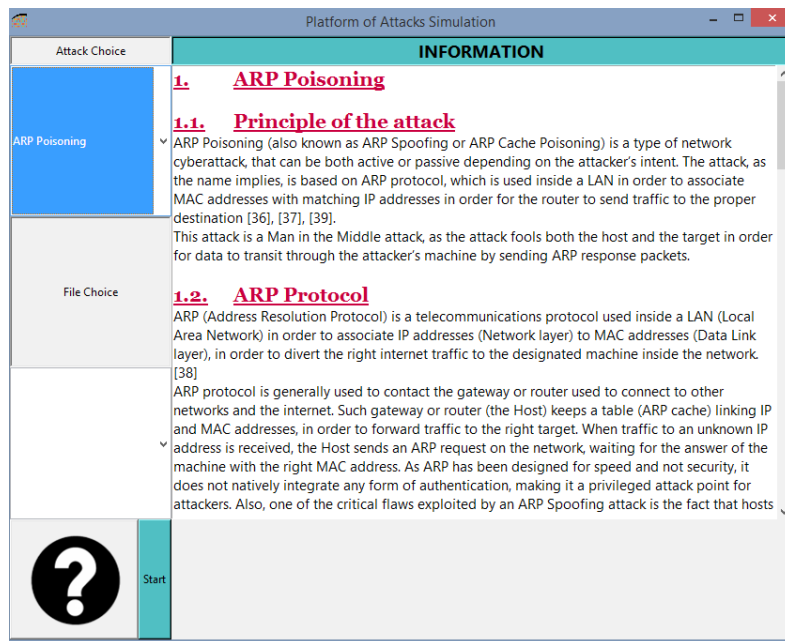


Figure 2: Fenêtre d'Information

Le bouton « Point d'interrogation » est le bouton d'information, après clic dessus, les informations sur l'attaque sélectionnée dans la liste déroulante s'affichent; et si aucune attaque n'est sélectionnée, un texte général sur les attaques et les différents attaquants est affiché.

L'illustration de l'attaque Arp Poisoning montre une table usurpée où on voit l'adresse de l'attaquant en double :

Platform of Attacks Simulation

Attack Choice

RESULTS

Internet Address	Physical Address	Type
172.16.1.64	10-40-f3-ab-71-02	dynamic
172.16.1.254	10-40-f3-ab-71-02	dynamic
172.16.1.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

File Choice

Start

Figure 3: Table de Correspondances à la suite d'un attaque ARP Poisoning

L'illustration de l'attaque DHCP Spoofing montre des trames récupérées en usurpant le serveur :

Attack Choice

DHCP Poisoning

File Choice

?

itar

Platform of Attacks Simulation

RESULTS

Number	Time	Source	Destination	Protocol	Length	Information
1	0.060875	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
2	0.061468	192.168.1.1	255.255.255.255	DHCP	342	DHCP Offer
3	0.061673	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
4	0.062448	192.168.1.1	255.255.255.255	DHCP	342	DHCP Ack

Figure 4: Trames récupérée lors de l'usurpation de l'identité du serveur DHCP

1.3.3.Extraction d'images à partir d'un fichier de capture

Deux fichiers de capture contenant des paquets avec des images sont disponibles pour montrer la possibilité d'extraire des images depuis un fichier de capture.

Les images extraites se retrouvent dans le dossier "Pictures" dans l'application :

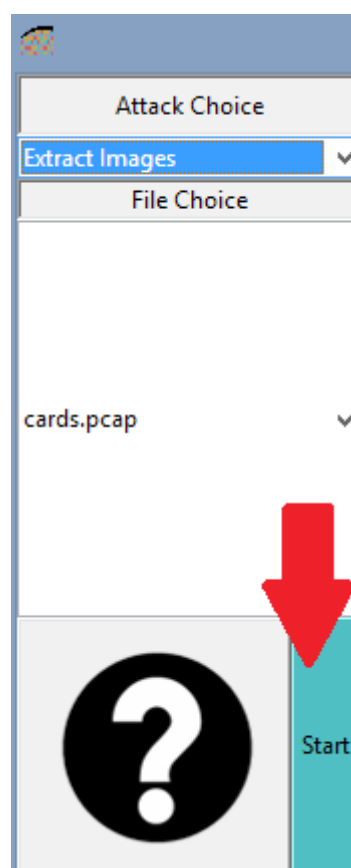


Figure 5: Extraction des images

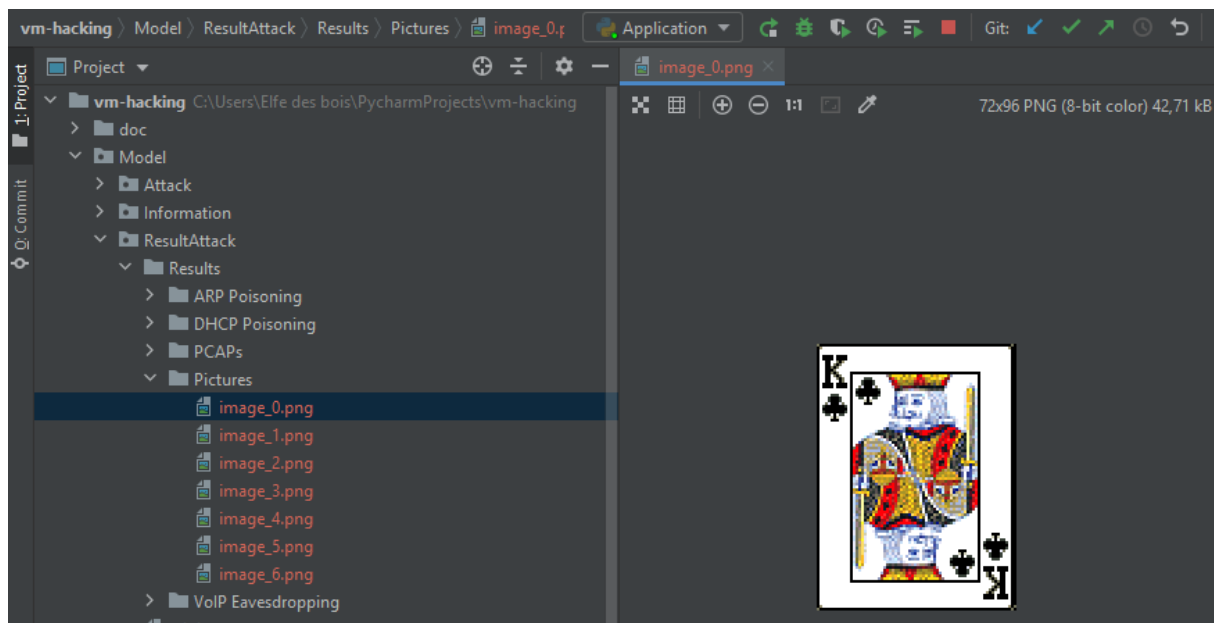


Figure 6: Images extraites

Pour obtenir ce fichier contenant des cartes, nous avons capturé le trafic en consultant l'espace personnel de l'ensicaen : <http://www.ecole.ensicaen.fr/~login/>

1.3.4. Lecture d'un fichier audio sur la plateforme

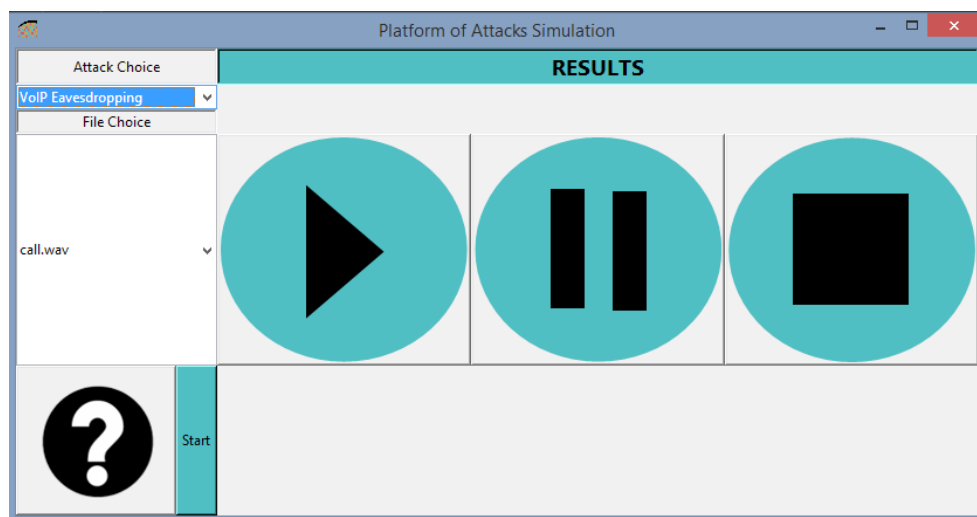


Figure 7: Lecture d'un fichier d'écoute sur VoIP

La plateforme possède un lecteur audio intégré qui permet de lire notamment les deux fichiers audios issus des fichiers extraits à partir des .pcap .

2. Machines Virtuelles et Configuration Réseau

Cette partie présente de la manière la plus exhaustive possible l'installation et la configuration sous Linux des différents outils nécessaires pour opérer le réseau utilisé dans le cadre de la plateforme. Les différentes sous-parties correspondent aux différentes phases de la configuration et doivent être suivies dans l'ordre. Ne pas suivre l'ordre des différentes étapes est à vos risques et périls.

2.1. Installation des logiciels GNS3 et VirtualBox

Les deux logiciels sont téléchargeables gratuitement en ligne et ne nécessitent pas de licence pour fonctionner. Suivez ces tutoriaux pour télécharger et installer [GNS3](#) et [VirtualBox](#) sous Linux. Les procédures d'installation et les téléchargements à effectuer pour une installation sous Windows sont également aisément trouvables sur les sites indiqués ci-dessus.

(À titre indicatif, la configuration initiale du réseau a été réalisée sur Linux Ubuntu 20.04 LTS)

2.2. Installation et lien de la VM GNS3

Suivre pour cette partie [ce tutoriel](#). Une seule modification à effectuer par rapport au tutoriel : Lors de l'étape consistant à ajouter un routeur, remplacer le routeur C3725 suggéré par un routeur C7200, routeur qui sera utilisé par la suite. Le routeur choisi est un routeur Cisco C7200, modèle très répandu qui est très utilisé en entreprise. L'image du routeur est téléchargeable [ici](#), et peut également être téléchargée depuis l'espace dédié au cours de réseau dispensé en première année dans la filière informatique à l'ENSICAEN. Lors de la configuration réseau, activez le slot 0.

2.3. Setup GNS3

Veillez dans un premier temps à retirer GNS3 des restrictions de votre pare-feu. Ensuite, ouvrez VirtualBox et ouvrez le panneau de configuration VM GNS3. Dans la section Réseau, sélectionnez "connexion par pont" pour l'adaptateur n°3 : Ce dernier permet de faire le lien entre le réseau virtuel et la machine host.

2.4. Configuration des différentes machines virtuelles

Il est nécessaire de configurer 3 machines virtuelles différentes afin de pouvoir mettre en place le réseau souhaité. Les trois machines virtuelles sont configurées sur une base Linux Ubuntu 20.04 LTS et sont respectivement nommées *User*, *Target* et *Attacker*. Les 3 configurations sont identiques : Configuration minimale lors de l'installation, *password* identique au *username* de la machine avec Log In automatique pour faciliter la phase d'installation et de développement du projet. Une fois les mises à jours initiales effectuées lors du premier lancement de chaque VM, installez le package *net-tools* ainsi que le package *virtualbox-guest-utils* :


```
sudo apt install net-tools  
sudo apt install virtualbox-guest-utils
```

Validez ensuite la bonne installation du premier package via l'utilisation de la commande *ifconfig* qui doit vous lister l'ensemble des interfaces réseau et leurs caractéristiques. La bonne installation du second groupe de package sera nécessaire pour que les VM puissent partager des fichiers avec le système host.

Shutdown l'ensemble des VM et configurez pour chacune d'entre-elles l'ensemble des *network adapter* en *non attached*, ce qui forcera la reconfiguration lors de l'utilisation des VM au sein de notre réseau sur GNS3 et le passage par l'interface choisie que nous configurons par la suite.

La configuration des machines virtuelles est alors terminée.

2.5. Création du projet GNS3 et Importation des VMs

Lancez GNS3 et créez un nouveau projet nommé *Projet_VM*. Il faut alors importer le routeur dont l'ISO a été précédemment téléchargé (*Partie b.*) qui servira de serveur DHCP lors des simulations (*Nous considérons ici le cas d'une "grande entreprise", cas dans lequel les serveurs DHCP utilisés sont des routeurs et non des ordinateurs "classiques" reconfigurés comme cela peut être le cas pour des structures plus modestes*).

Une fois le routeur importé, il vous faut importer également les 3 VMs créées et configurées lors de la phase précédente : *Edit* → *Preferences* → *VirtualBox VMs* → *New*. Il est conseillé d'avoir VirtualBox est en cours d'exécution lors de la phase d'importation, GNS3 étant dans ce cas capable de détecter les différentes VM et de les importer quasi automatiquement.

Une fois les VMs importées, il faut modifier la configuration réseau pour permettre à GNS3 d'utiliser tout adaptateur précédemment configuré par VirtualBox : *Edit* → *Preferences* → *VirtualBox VMs* → *Edit* → *Network*. Il ne reste alors plus qu'à checker la case correspondante, ce qui conclut la configuration initiale du projet.

2.6. Setup Réseau de Base

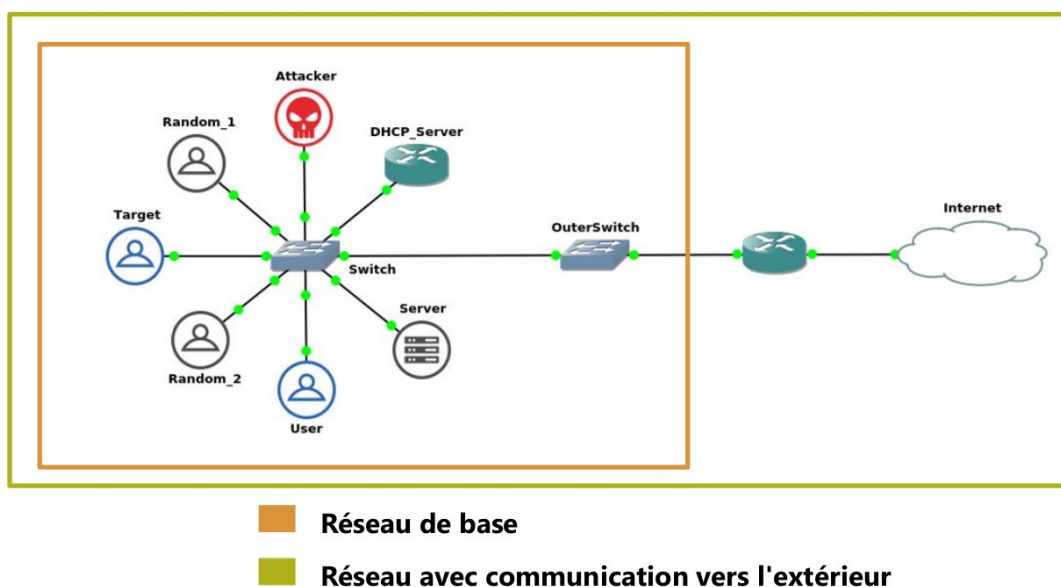


Figure 8: Réseau utilisé dans le cadre de la plateforme

Importer une VM de chaque "type" (*attacker*, *target*, *user*), 2 switch Ethernet, 2 routeurs Cisco C7200, un "cloud" NAT (ce dernier symbolise la connexion au PC host et pourra être utilisé pour communiquer entre le réseau simulé et le PC host), ainsi que 3 VPCS (ces derniers représentant des utilisateurs ou des serveurs également sur le réseau). Relier les différents composants comme indiqué sur la figure ci-dessus.

La seule contrainte à vérifier est de relier le switch central à l'interface *FastEthernet 0/0* du routeur ayant le rôle de serveur DHCP. Une fois ceci fait, il ne reste plus qu'à configurer le serveur DHCP. Après avoir lancé le réseau, ouvrez une console routeur et entrez les commandes de configuration suivantes :

<pre># conf t</pre>	Bascule en mode configuration
<pre># service dhcp</pre>	Active le routeur en mode dhcp
<pre># ip dhcp pool LAN</pre>	Déclare un nouveau pool LAN DHCP
<pre># network 192.168.1.0 255.255.255.0</pre>	Définit la range d'adresses du pool
<pre># lease 7 (Optional)</pre>	Définit la durée de validité du pool
<pre># default-router 192.168.1.1</pre>	Définit l'adresse de Gateway
<pre># exit</pre>	Sortie de la configuration DHCP
<pre># int fa0/0</pre>	Configuration de l'interface fa0/0
<pre># ip address 192.168.1.1 255.255.255.0</pre>	Configuration de l'adresse statique
<pre># no shut</pre>	Maintient l'interface active
<pre># exit</pre>	Sortie de la configuration de l'interface
<pre># copy running-config startup-config</pre>	Sauvegarde la configuration
<pre># exit</pre>	Sortie du mode de configuration

La configuration du routeur jouant le rôle de serveur DHCP conclue la configuration du réseau de base.

2.7. Test de Fonctionnalité du Réseau de Base

Lancer le réseau GNS3 dans son ensemble, ce qui lancera également les 3 VM. Sur chacune d'entre-elles, lancer la commande *ifconfig*. Cette commande doit normalement afficher, pour la première interface de chaque VM (*théoriquement eth0/0*), une adresse IP associée de la forme 192.168.1.X, X étant compris entre 2 et 254.

Si aucune adresse IP n'est associée à une des VM lors de l'exécution de la commande *ifconfig*, utilisez alors la commande *dhclient -r* pour rafraîchir la configuration du serveur DHCP local (*i.e le routeur précédemment configuré*), avant de relancer la commande *ifconfig*, qui devrait normalement vous ramener dans le premier cas.

Les différentes VMs sont alors capables d'interagir entre elles au sein du réseau, ce qui permet dès lors l'utilisation des différentes attaques implémentées.

2.8. Gestion des Fichiers et Dossiers partagés

L'exécution des différentes attaques à partir des machines virtuelles du réseau simulé nécessite l'utilisation de dossiers partagés, présents sur le PC host et à la fois accessibles par les VMs pour exécuter les différents scripts d'attaque. Les modules *virtualbox-guest-** précédemment installés permettent aux VMs d'avoir accès aux dossiers partagés précédemment configurés au sein de VirtualBox. Cette dernière partie de configuration est l'objet de cette section.

Lancez tout d'abord VirtualBox. Pour chacune des VM du projet, aller dans le panneau de configuration de la VM sélectionnée, dans la partie *shared folders*, et ajouter le dossier contenant l'ensemble des différents scripts d'attaques que vous aurez préalablement téléchargé sur le PC host. Le chemin permettant d'accéder depuis la VM aux script d'attaque est à configurer à votre convenance.

Pour vérifier que les fichiers sont bien accessibles à partir des VM, lancer chacune de ces dernières et vérifier en ouvrant Nautilus ou un terminal que vous pouvez bien lister le contenu des dossiers partagés ainsi que lire le contenu des différents scripts d'attaque. Les VM seront ainsi capables, une fois lancés avec le reste du réseau dans GNS3, d'interagir également avec les fichiers des dossiers partagés (*exécution d'attaque, sauvegarde des résultats, etc*).

3. Attaque ARP Poisoning

3.1. Mise en place des outils de l'attaque

Afin d'exécuter l'attaque ARP Poisoning, seuls un interpréteur Python 3 ainsi que le script de l'attaque sont nécessaires. Ce script a été développé avec Python 3.8, et il est conseillé d'avoir une version égale ou supérieure d'interpréteur pour exécuter l'attaque. Il convient également de vérifier que les adresses IP utilisées par le script sont bien valides au regard du réseau utilisé.

Dans le cas où l'attaque est réalisée sur la plateforme, le script d'attaque doit être placé au sein d'un des dossiers partagés avec la VM *attacker* (cf. 6.1.).

3.2. Exécution de l'Attaque

L'attaque ARP Poisoning débute une fois que le script a été exécuté sur la machine attaquante. L'exécution s'effectue à l'aide de la commande suivante :

```
sudo python3 arper.py
```

Une fois le script lancé, ce dernier va envoyer à intervalle de temps régulier (2 secondes dans notre cas, paramètre réglable) une "Gratuitous ARP Response" à la fois à la machine cible et à la gateway, le renvoi régulier étant mis en place pour s'assurer que le trafic destiné à la cible soit rerouté vers l'attaquant durant toute la durée souhaitée. Enfin, une fois que le nombre de requêtes restant à envoyer est nul, le script rétablit la communication initiale et met fin à l'attaque en sauvegardant les paquets capturés durant l'attaque dans un fichier PCAP.

3.3. Vérification de l'Attaque

La vérification du bon déroulement de l'attaque ne peut se faire que lorsque l'attaque est en cours d'exécution, le script d'attaque prenant le soin de rétablir la communication initiale et d'effacer ses traces une fois l'attaque terminée.

La manière la plus aisée de vérifier manuellement le bon déroulement de l'attaque est de comparer le contenu du cache ARP d'une machine tiers au sein du même réseau qui n'est pas impactée par l'attaque (*qui correspond sur la plateforme à la VM User*) avant et pendant l'attaque. On constatera alors que l'adresse IP de la cible sera bien devenue celle de l'attaquant. De même une fois l'attaque terminée, une consultation du contenu du cache indiquera bien que l'adresse IP de la cible est de nouveau sa véritable IP.

4. Attaque DHCP Spoofing

4.1. Mise en place des outils de l'attaque

Le script nécessite la bibliothèque "scapy", qu'il faut donc installer sur la machine susceptible d'exécuter le code. Pour ce faire, il existe deux façons. La première consiste à passer par pip et exécuter la commande suivante:

```
pip install --pre scapy[basic]
```

Si cette méthode ne fonctionne pas (*ce qui était le cas*), une autre solution consiste à télécharger la bibliothèque scapy sur [ce github](#). Déposer ensuite le script dans le dossier principal et l'exécuter depuis ce dossier. Le script sera alors en mesure de trouver la bibliothèque.

Ce code a été réalisé avec Python 3.8.5. Bien qu'il soit probablement possible d'utiliser ce script avec une version antérieure de python, cela est fortement déconseillé.

4.2. Exécution de l'Attaque

Avant de lancer le script, vérifiez quelle est l'interface réseau qui est utilisée dans le réseau. Pour ce faire, faites la commande "ifconfig" et regardez quelle interface a une adresse IP appartenant au sous réseau 10.0.0.0/255.255.255.0. Remplacez alors au sein du script la valeur de la variable "IFACE" par le nom de l'interface que vous venez de trouver.

Pour lancer l'attaque, il suffit de lancer le script grâce à Python. Il faut cependant l'exécuter en mode administrateur. Comme le script requiert un accès à certains sockets réseau qui ne sont normalement pas censés être utilisés par un utilisateur lambda, il faut lancer le script en mode admin. On peut donc le lancer via la commande :

```
sudo python3 dhcp_spoofing.py
```

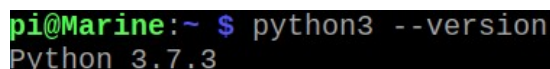
4.3. Vérification de l'Attaque

Le script affiche un message dans le terminal à chaque requête envoyée. La fin de l'attaque provoque l'arrêt de l'affichage de messages. Un fichier "request.txt" est créé à chaque lancement de l'attaque dans le dossier dans lequel le script se trouve. Ce fichier récapitule toutes les requêtes envoyées par l'attaque. On peut donc y lire toutes les adresses IP qui ont été récupérées par l'attaque.

5. Extraction d'Audio à partir d'un fichier PCAP

5.1. Version de Python

Ce script a été développé sous Python 3.7.3. Il est recommandé d'utiliser cette version au minimum. Pour connaître votre version de Python :



```
pi@Marine:~ $ python3 --version
Python 3.7.3
```

Figure 9: Version de l'interpréteur Python installé

Si votre version est trop ancienne, une version plus récente est disponible [ici](#).

5.2. Installation des Modules

5.2.1. Pyshark

Commande pour installer pyshark :

```
pip install pyshark
```

5.2.2. SoX – Sound eXchange

Installation SoX sous Linux :

```
sudo apt update
sudo apt install sox
```

Pour Windows, SoX peut être téléchargé [ici](#).

5.2.3. Modules subprocess, os et sys

Ces modules sont directement intégrés à Python. Aucune installation n'est donc requise.

5.3. Comment utiliser le script

Désormais vous avez tous les modules et la version Python permettant d'utiliser le script. Nous allons voir quelles commandes il faut réaliser afin d'extraire un fichier audio.

5.3.1. Ressources disponibles

Deux fichiers de capture, des fichiers d'extension .pcap, sont fournis dans un dossier nommé « PCAPs » afin de pouvoir utiliser le script. Il s'agit des fichiers « Forensic.pcap » et « call.pcap ».

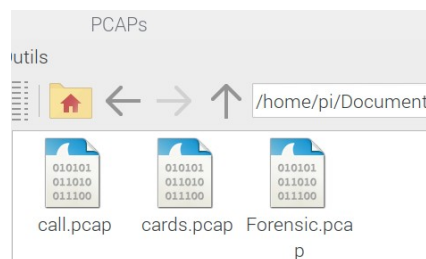


Figure 10: Fichiers PCAP fournis

Notons que dans le cadre de ce tutoriel, il y a un fichier en plus nommé cards.pcap, c'est un fichier qui ne contient pas d'audio et qui est ici uniquement là afin de présenter le comportement du script face à un fichier de capture ne contenant pas de contenu audio.

5.3.2. Commande pour extraire le fichier audio de Forensic.cap

```
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  PCAPs
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ python3 AudioPCAP.py PCAPs/Forensic.pcap forensic
Trying to find RTP payload ...
Creating raw audio file ...
Converting to .wav ...
Completed process: the file forensic.wav is in your working directory
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  forensic.wav  PCAPs
```

Figure 11: Commande correcte d'extraction

Analysons les différentes étapes :

- 1) Au début, on n'a dans le répertoire courant que le dossier avec les ressources (les trois fichiers de capture en .pcap) et le script python
- 2) Ensuite on exécute la commande :

```
python3 AudioPCAP.py PCAPs/Forensic.pcap forensic
```

- 3) On récupère notre fichier dans le répertoire courant, avec le nom précisé en second argument lors de l'appel du script

5.3.3. Arguments Obligatoires

Comme on vient de le voir, le script prend deux arguments obligatoires :

- 1) Fichier .pcap (attention toute autre extension sera refusée par le script)
- 2) Nom du fichier de sortie

Ici, on a donc choisi comme de nom de fichier « forensic » et on a bien récupéré notre .wav qui s'appelle **forensic.wav** dans le répertoire courant.

5.3.4. Lecture du Fichier Audio

Le fichier audio extrait peut être lu par un lecteur classique supportant les fichiers .wav, comme VLC par exemple.

À titre indicatif, l'audio extrait de Forensic.pcap fait 59 secondes et l'audio extrait de call.pcap fait 25 secondes. Le premier extrait audio, Forensic.pcap, est issu d'un CTF (Capture The Flag) avec le flag donné à la fin et le deuxième extrait, de call.pcap est simplement la sonnerie d'un téléphone ainsi que la voix d'un homme à la fin.

5.3.5. Fichier sans contenu Audio

Dans le cas où le fichier fourni ne contiendrait pas de contenu audio (comme le fichier cards.pcap) :

```
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ python3 AudioPCAP.py PCAPs/cards.pcap cards
Trying to find RTP payload ...
Your .pcap does not have packets using RTP, no audio found !
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs
```

Figure 12: Erreur lorsque qu'aucun fichier audio n'est trouvé

Le script signale à l'utilisateur qu'il ne s'est rien passé car le fichier n'a pas de paquets avec le protocole RTP.

5.3.6. Commandes Incorrectes

Si jamais vous ne mettez pas du tout ou pas assez d'arguments, alors le script refusera de s'exécuter et il ne se passera rien :


```

pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ python3 AudioPCAP.py
Bad number of arguments, must be 2!
First argument: a .pcap
Second argument: name of output .wav file
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs

```

Figure 13: Nombre incorrect d'arguments

De la même façon, si le fichier fourni n'est pas un .pcap, le script refusera de s'exécuter et il ne se passera rien :

```

pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ python3 AudioPCAP.py call.wav bad_extension
First argument was not .pcap file !
Your first argument has .wav for extension
pi@Marine:~/Documents/Work/Projet2A/AudioPCAP $ ls
AudioPCAP.py  call.wav  forensic.wav  PCAPs

```

Figure 14: Erreur dans le type des arguments

5.3.7. Comparer l'audio extrait et celui généré par Wireshark

Wireshark permet de recréer les fichiers audios d'un fichier de capture et de les lire. Pour cela, rendez-vous sur Wireshark et ouvrez le fichier de capture, puis dans l'onglet « Téléphonie » :

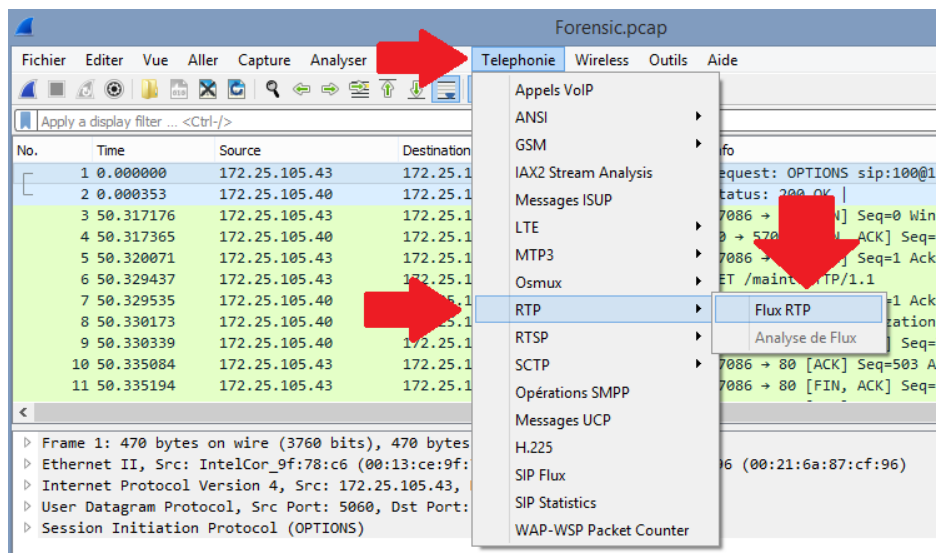


Figure 15: Accès au Flux RTP via Wireshark

On remarque qu'on a deux appels, qui avec le script sont fusionnés en un fichier audio, on peut les sélectionner pour les lire (individuellement également) :

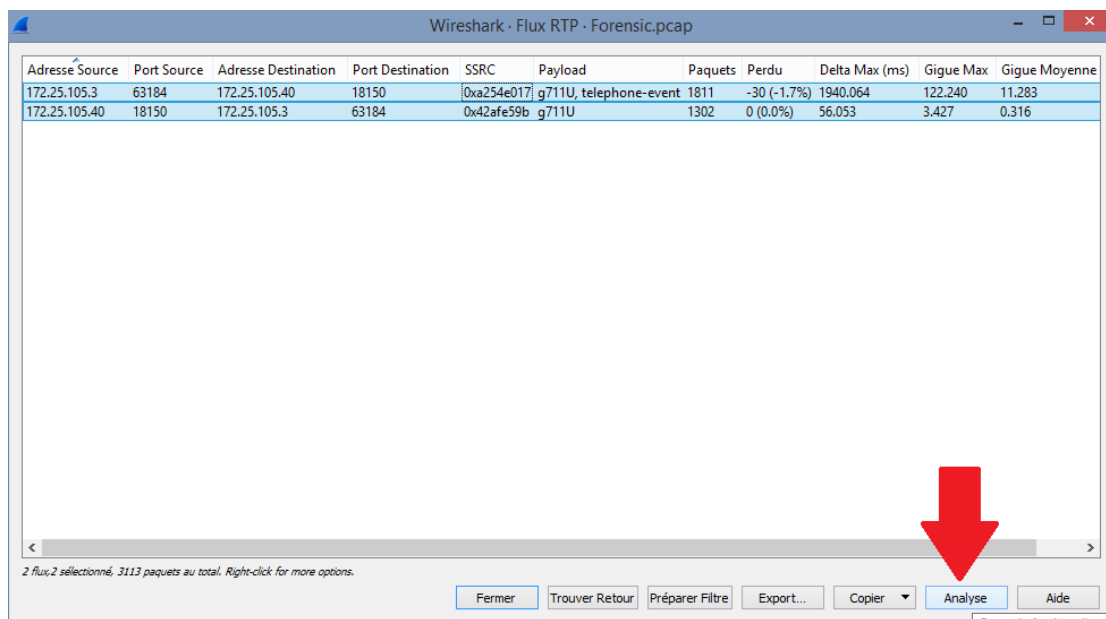


Figure 16: Analyse des fichiers audio depuis Wireshark

Cliquer ensuite sur le bouton « Jouer Flux » permet de lire/écouter les audios ainsi extraits.

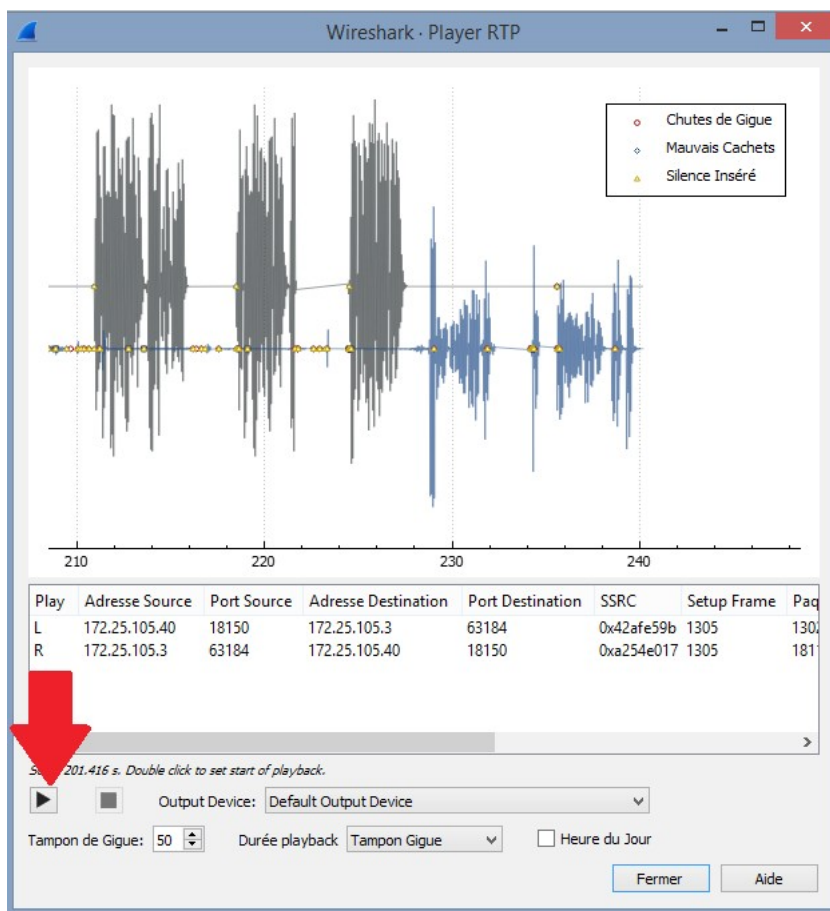


Figure 17: Informations sur l'audio et lecture

6. Extensions possibles envisagées

6.1. Réseau : Lien avec le PC Host

Cette partie est indiquée en tant qu'extension possible car nous n'avons pas pu la compléter de par les limites de délai du projet. La solution de lien avec le PC Host proposée ci-après a donc été testée dans un cas particulier mais n'a pas bénéficié de tests extensifs et peut donc nécessiter des mises à jour et ajustements.

Configurez le cloud NAT pour pouvoir utiliser l'interface *eth2*. Relier le switch à l'extrémité du réseau de base à l'interface *FA0/1* du routeur de lien, puis connecter l'interface *FA0/0* du routeur de lien à l'interface *eth2* du cloud. Ouvrez ensuite une console du routeur de lien et entrez les commandes suivantes :

# conf t	Bascule en mode configuration
# int fa0/0	Configuration de l'interface fa0/0
# ip address dhcp	Active l'attribution d'adresse IP via un serveur DHCP local
# no shutdown	Maintient l'interface active
# end	Fin de la configuration

Une fois les commandes ci-dessus entrées, le routeur doit indiquer qu'une adresse a été attribuée à l'interface reliée au *réseau de base*. Pour tester le bon fonctionnement, on peut utiliser la commande *ping* depuis une des VM et depuis le PC host pour vérifier que le routeur répond bien dans les deux cas. Il nous reste alors à configurer les différentes routes pour que les différentes VM puissent échanger avec le PC host :

# conf t	Bascule en mode configuration
# int fa0/1	Configuration de l'interface fa0/1
# nat inside	Activation du NAT
# ip route 10.1.1.0 192.168.1.0 192.168.1.xxx	Adresse fixe de la route
# exit	Sortie de configuration de l'interface
# int fa0/0	Configuration de l'interface fa0/0
# nat outside	Activation du NAT
# exit	Sortie de configuration de l'interface
# ip route 10.1.1.0 192.168.1.0 192.168.1.xxx	(adresse du routeur interface 0/0)
# ip nat inside source static "attacker IP" 10.1.1.xxx	(adresse routeur interface 0/1)
# copy running-config startup-config	Sauvegarde la configuration
# exit	Sortie du mode de configuration

Il ne reste alors plus qu'une modification à apporter à la configuration du serveur DHCP :

# conf t	Bascule en mode configuration
# int fa0/0	Configuration de l'interface fa0/0
# ip route 10.1.1.0 192.168.1.0 192.168.1.xxx	(adresse du routeur interface 0/0)

```
# copy running-config startup-config  
# exit
```

Sauvegarde la configuration
Sortie du mode de configuration

Les différentes VMs peuvent désormais envoyer du trafic à destination du PC host. Le test de la communication peut être effectué à l'aide d'un serveur Python, les détails de ce test étant à votre charge.



École Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 4505
14050 CAEN cedex 04

