Projet de Sécurité
Groupe Analyse

Réalisé par :

Sébastien ARNAUD
Paul BIZOUARD
Yoann DELOMIER
Jérôme ESQUIE
Anne FALGUERA
Sébastien LAUDIC
Lydie SALVAN

SOMMAIRE

1 Introduction

1.1 Présentation du projet[1]

Le projet consiste à étudier et déployer une maquette d'infrastructure d'entreprise suivant un scénario type[2].

Le projet est divisé en trois groupes de travail :

Un groupe Défense qui est chargé de mettre en place l'infrastructure des services du scénario d'entreprise. Il doit rechercher les moyens les plus simples possibles pour se défendre contre les tentatives d'intrusion et de compromission entreprises par le groupe «Attaque».

Un groupe Analyse qui est chargé de collecter un maximum d'informations et de les analyser pour identifier les actions entreprises aussi bien en défense qu'en attaque.

Un groupe Attaque qui lui, est chargé de rechercher toutes les possibilités d'intrusion et de compromission les plus efficaces et les plus faciles à mettre en oeuvre.

Chaque groupe dispose d'une liste de diffusion de courrier électronique à laquelle est attachée un espace documentaire privatif. Ces listes sont l'outil principal de communication et surtout de coordination du groupe.

1.2 Les objectifs du groupe Analyse

Appréhender le travail en équipe.
Appréhender les difficultés de la sécurité informatique.
Appréhender les métiers concernant l'audit d'un réseau informatique.
Appréhender la communication entre les groupes ayant des rôles et des objectifs différents.

1.3 Buts à atteindre

Le but du groupe Analyse est essentiellement de récupérer les fichiers log et de repérer les attaques qui ont réussi et celles qui ont échoué en raison des précautions prises par l'équipe Défense.

1.4 Présentation du plan

Dans la suite de ce document, nous allons vous présenter tout d'abord la répartition des tâches parmi les membres du groupe ainsi que leur position dans le planning.

Après avoir défini quelque peu notre échéancier ainsi que les tâches que nous avions prévus de faire, nous vous présenterons le travail effectivement réalisé au cours des différentes phases du projet, mais aussi les difficultés rencontrées et les bilans de chaque phase.

---

[1] http://www.linux-france.org/prj/inetdoc/formations/m2-stri/

[2] http://www.linux-france.org/prj/inetdoc/formations/m2-stri/scenario.html

Enfin, avant de conclure sur le projet, nous vous détaillerons notre politique de sécurité et les perspectives pour améliorer l'audit d'une telle infrastructure.

2 Distribution des taches en fonction des objectifs

2.1 Répartition des tâches parmi les membres du groupe

N'ayant aucune information concernant le déroulement des attaques, et l'évolution de l'architecture de l'équipe défense, nous n'avons pas pu définir des rôles précis à chaque personne de l'équipe.
Néanmoins, les rôles ont été attribués de la sorte :

Yoann : Coordinateur, veille technologique.
Jérôme : Chargé des relations Inter-groupes, Analyste.
Sébastien A : Analyste.
Paul : Ingénieur Réseau.
Lydie : Veille technologique, Rédactrice.
Anne : Veille technologie, Gestion de la machine Windows.
Sébastien L : Veille technologique.

2.2 Positionnement des tâches dans le planning

Voici la répartition des tâches durant la durée du projet :

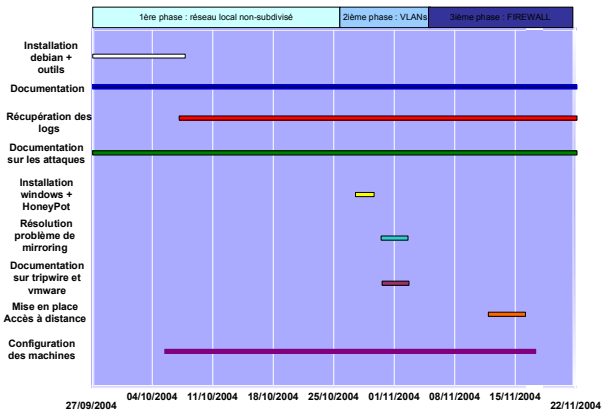


Figure 1 : Planning des tâches

3 Echéancier temporel et ce qui a été prévu

3.1 Synthèse des échéanciers

Nous étions conscients de l'importance, sur le plan professionnel, de fournir des informations sur les tâches à effectuer aussi bien en terme de dates que de coûts. Dans notre cas, seul le problème de date nous importait, cependant la constitution de l'échéancier a représenté une des étapes les plus difficiles à gérer dans notre position et cela en raison d'un grand nombre de facteurs :

Tous d'abord nous avons été freinés par le côté obscur de notre tâche d'analyse. Nous avons débuté ce projet avec très peu d'informations concernant les autres équipes, et il nous a donc fallu attendre la première architecture proposée par la Défense pour commencer à établir notre champ d'action. Néanmoins nous en avons profité pour nous documenter sur les différents outils susceptibles de correspondre à notre tâche. Encore une fois, nous n'avions que très peu d'informations pour nous guider et cette recherche s'est donc effectuée dans le flou.

De plus nous avons été freinés par le peu de connaissances que nous disposions concernant la gestion d'un projet d'Analyse. En effet, n'ayant aucune idée de la quantité d'information que nous allions être amenés à collecter, il a été très difficile de prévoir le temps d'analyse. C'est donc tout au long du projet que nous avons découvert la durée et l'ampleur des tâches à réaliser.

Les attaques étant pour nous imprévisibles et relativement irrégulières, nous avons donc été soumis à des quantités de travail plus ou moins importantes. Il était donc compliqué de déterminer avec précision les tâches correspondantes.

Toutes ces raisons nous ont donc entravés dans la réalisation d'un échéancier prévisionnel, c'est pourquoi nous avons choisi de réaliser cet échéancier au fur et à mesure de l'avancement du projet. Ce projet étant à but pédagogique, il a bien rempli son rôle, car nous sommes aujourd'hui capable de prévoir l'organisation et la durée du travail avec beaucoup plus de précision, et ceci afin de satisfaire au mieux le client.

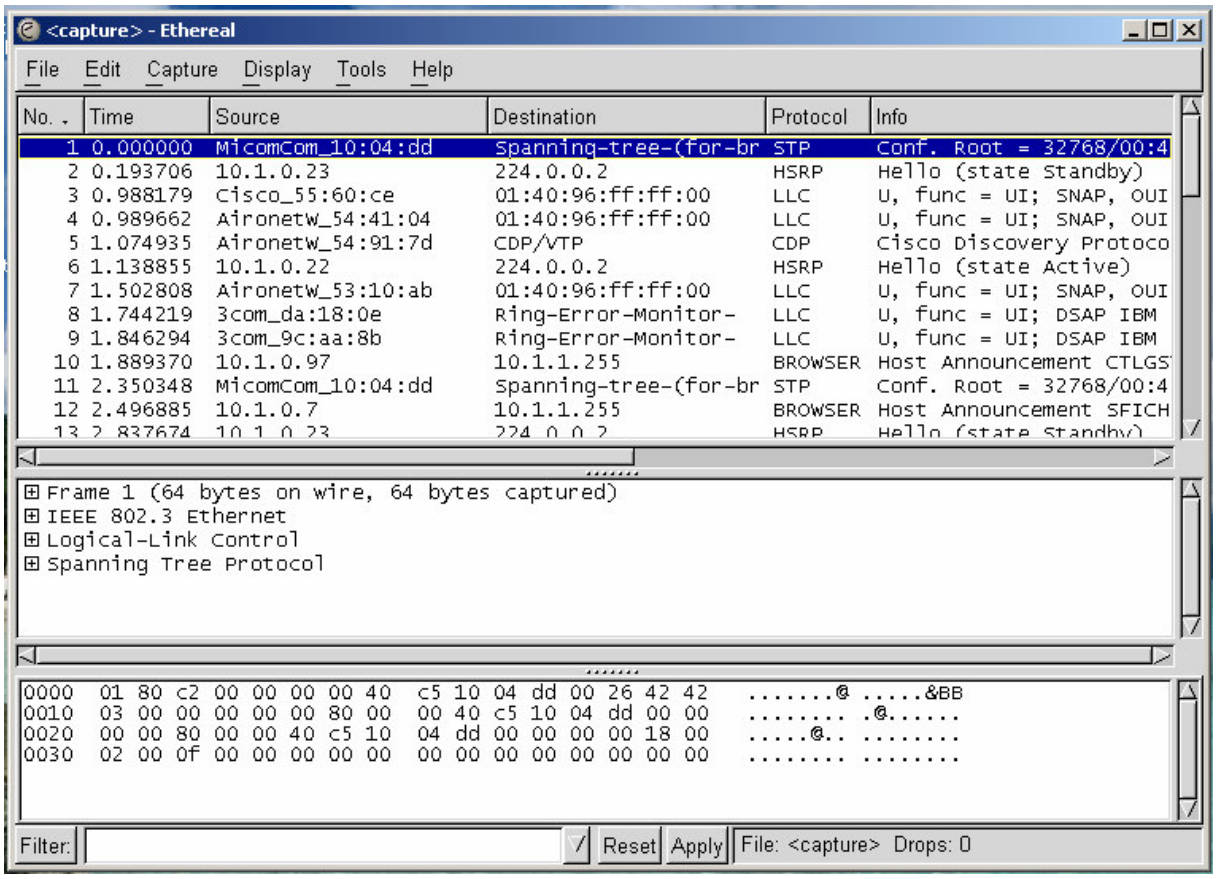
3.2 Ce qui a été prévu

Malgré l'absence d'échéancier, nous savions que quelques tâches relatives au métier d'analyste étaient incontournables, à savoir :

- Positionnement d'une machine d'analyse dans le réseau à des endroits stratégiques.
- Collecte et analyse du trafic réseau des machines susceptibles d'être attaquées.
- Etablissement de rapports hebdomadaires relatifs au trafic réseau analysé au cours de la semaine.
- Coordination avec l'équipe défense pour optimiser leur architecture réseau.

4 Outils utilisés

4.1 Ethereal et Tethereal



Les sniffers sont des outils permettant de récupérer les paquets qui passent physiquement sur un réseau (quelque soit la destination de ces paquets). Ethereal est un de ces outils sous licence GNU et qui permet d'interpréter la structure des paquets de façon graphique ou en ligne de commande (tethereal).

4.1.1 Le programme :

Ethereal permet de capturer les paquets en direct sur le réseau, ou de lire un fichier de capture (fichier pcap). Cet outil permet de disséquer plus de 602 protocoles.

Il est nécessaire de connecter la machine ayant Ethereal avec une carte réseau sur le réseau à analyser. Nous ne pouvons voir que les paquets qui passent physiquement sur le réseau auquel est connectée la carte réseau. Ce qui veut dire que si nous voulons analyser des paquets qui ne sont pas à destination de notre machine, il faut être sur un réseau à collisions (tout le monde utilise le même câble réseau à tour de rôle) ce qui est le cas des réseaux 10 base 2 (coax) ou 10/100 base T avec l'utilisation d'un concentrateur Ethernet ou HUB.

4.1.2 Les filtres

Il existe deux sortes de filtres : ceux à la capture et ceux à l'affichage. Ces filtres n'ont pas la même syntaxe, en effet, celle des filtres à la capture est la même que pour la commande

tcpdump (sous linux). Quant aux filtres à l'affichage, la syntaxe est propre à **ethereal**. La section suivante donne des exemples pour ces deux types de filtres.

Les filtres de capture :

Ne seront gardés que les paquets pour lesquels le filtre est vrai. Les filtres se décomposent en 3 parties :

- le *protocole* qui peut être **ether**, **fddi**, **ip**, **arp**, **rarp**, **decnet**, **lat**, **sca**, **moprc**, **mopdl**, **tcp** ou **udp,**
- la *direction* qui peut être **src** ou **dst**,
- un *champ* qui peut être **host**, **net** ou **port** suivi d'une valeur.

Les opérateurs **and**, **or** et **not** peuvent être utilisés pour combiner des filtres.

Filtre	Fonction
host 172.16.0.1 and tcp	Ne conserve que les paquets TCP à destination ou en provenance de la machine 172.16.0.1
udp port 53	Ne conserve que les paquets UDP en provenance ou en destination du port 53
udp port 53 and dst host 172.16.0.1	Ne conserve que les paquets UDP en provenance ou en destination du port 53 à destination de la machine 172.16.0.1
tcp dst port 80 and dst host 172.16.0.1 and src net 172.16.0.0 mask 255.255.255.0	Ne conserve que les paquets TCP en destination de la machine 172.16.0.1 sur le port 80 et en provenance des machines du réseau 172.16.0/24

Les filtres d'affichage :

Les filtres d'affichage sont un peu plus fins que ceux à la capture. Seuls les paquets pour lesquels l'expression du filtre est vraie seront gardés. Les expressions sont basées sur les champs disponibles dans un paquet. Le simple ajout d'un champ veut dire que l'on garde le paquet si ce champ est disponible. Maintenant, nous pouvons aussi utiliser les opérateurs **==**, **!=**, **>**, **<**, **>=** et **<=** pour comparer les champs avec des valeurs. Les expressions ainsi fabriquées peuvent être combinées avec les opérateurs **&&** (pour un et logique), || (pour un ou logique), **^^** (pour le ou exclusif) et **!** pour la négation. L'usage des parenthèses est possible.

Voici quelques exemples de champs disponibles :

Champ	Type	Signification
ip.addr	Adresse IPv4	Adresse IP source ou destination
ip.dst	Adresse IPv4	Adresse IP destination
ip.flags.df	Booléen	Drapeau IP, ne pas fragmenter
ip.flags.mf	Booléen	Drapeau IP, fragments à venir
ip.ttl	Entier non signé sur 8 bits	Time to live
nbdgm.src.ip	Adresse IPv4	Adresse IP source d'un paquet Netbios Datagram
nbdgm.src.port	Entier non signé sur 16 bits	Port IP source d'un paquet Netbios Datagram
http.request	Booléen	Requête http
http.response	Booléen	Réponse http
icmp.code	Entier non signé sur 8 bits	Numéro du code d'une commande ICMP
icmp.type	Entier non signé sur 8 bits	Numéro du type d'une commande ICMP
ftp.request	Booléen	Requête FTP
ftp.request.command	Chaîne de caractères	Commande FTP
ftp.reponse.data	Chaîne de caractères	Donnée de transfert FTP
dns.query	Booléen	Requête DNS
dns.response	Booléen	Réponse d'une requête DNS

Voici quelques exemples de filtres :

Filtre	Signification
ip.addr == 172.16.0.100	Tous les paquets IP en provenance ou à destination de la machine 172.16.0.100
(ip.addr == 172.16.0.100) && (dns.response	Tous les paquets IP en provenance ou à destination de la machine 172.16.0.100 qui sont des réponses à des requêtes DNS
(ip.addr >= 172.16.0.100) && (ip.addr <= 172.16.0.123)	Tous les paquets IP en provenance ou à destination des machines comprises entre l'adresse IP 172.16.0.100 et l'adresse IP 172.16.0.123 (comprises)

4.2 SNORT

4.2.1 Présentation

Snort est un système de détection d'intrusion réseau en sources ouvertes, capable d'effectuer l'analyse du trafic en temps réel et de la journalisation de paquets sur des réseaux IP.

Il peut effectuer de l'analyse de protocoles, de la recherche / correspondance de contenu afin de détecter une variété d'attaques et de scans, tels que :

- des débordements de tampons,
- des scans de ports furtifs,
- des attaques CGI,
- des scans SMB,
- des tentatives d'identification d'OS, etc.

Snort utilise un langage de règles flexible pour décrire le trafic qu'il devrait collecter ou laisser passer, ainsi qu'un moteur de détection qui utilise une architecture modulaire de plug-in.

Snort possède également des capacités modulaires d'alertes temps réel, incorporant des mécanismes d'alerte pour SYSLOG, des fichiers spécifiés par l'utilisateur, une socket UNIX, ou des messages WinPopup à des clients Windows en utilisant smbclient de Samba.

Snort a trois modes de fonctionnement principaux. Il peut être utilisé :

- comme un simple renifleur de paquets comme tcpdump,
- un enregistreur de paquets (utile pour déboguer le trafic réseau, etc),
- ou comme un système complet de détection d'intrusion réseau.

Snort journalise dans beaucoup de formats, dont le format binaire tcpdump (utilisable par la suite par un sniffer réseau tel que Ethereal) et le format ASCII décodé de Snort vers un ensemble hiérarchique de répertoires qui sont nommés d'après l'adresse IP du système "étranger".

Les plugins permettent aux sous-systèmes de détection et de rapports d'être étendus.
Les plugins disponibles incluent :

- la journalisation dans une base de données (MySQL) ou en XML,
- la détection de petits fragments,
- la détection de ports de scans,
- la normalisation des URL HTTP,
- la défragmentation IP,
- le réassemblage de flux TCP,

- la détection statistique d'anomalies.

Snort possède trois modes d'exécution principaux : sniffer (renifleur réseau), enregistreur de paquets, et système de détection d'intrusion réseau.

4.2.1.1 Mode renifleur

Dans ce mode, Snort lit et décode tous les paquets du réseau et les affiche sur la sortie standard. On peut voir les entêtes ainsi que les charges des paquets en spécifiant les options "-v" et "-d", afficher le contenu brut des octets de l'intégralité du paquet, en spécifiant l'option "-X".

4.2.1.2 Mode enregistreur de paquets

Ce mode enregistre les paquets sur le disque dans leur format ASCII décodé. Ceci enregistrera les paquets dans le répertoire de journalisation spécifié dans une hiérarchie de répertoires basés sur l'adresse IP des paquets sur le réseau.

Pour journaliser les paquets du point de vue du réseau surveillé (les répertoires créés dans celui d'enregistrement sont les adresses IP des systèmes distants / non locaux).

Les paquets peuvent être journaliser dans leur format binaire brut sur le disque. Le fait de journaliser les paquets dans ce format leur permettront d'être traités dans d'autres outils comme Ethereal, tcpdump, etc.

Le mode enregistreur de paquets peut être combiné aux options du mode renifleur sans mauvais effet, cependant les performances de journalisation peuvent être impactées par les possibilités de la machine.

4.2.1.3 Mode de détection d'intrusion

Snort entre en mode IDS quand un fichier de configuration est spécifié avec l'option "-c". Il s'agit du fichier snort.conf.

Les formats de sortie, les règles, la configuration des préprocesseurs, etc sont spécifiés dans ce fichier de configuration. Il est possible de spécifier les paquets que l'on souhaite voir journalisés.

Quand une alerte est déclenchée par une règle, les données sont journalisées vers le mécanisme d'alerte (par défaut un fichier appelé "alert" dans le répertoire d'enregistrement) en plus d'être journalisées vers le mécanisme d'enregistrement.

Le répertoire d'enregistrement par défaut est /var/log/snort, il est modifiable.

Les alertes peuvent également être envoyées à Syslog ou elles peuvent être envoyées comme des messages WinPopup avec smbclient ou encore par mail.

4.2.2 Avantages et inconvénients

Snort est un logiciel libre qui concurrence habilement les outils de détection d'intrusion commerciaux.

Le format ouvert de ses signatures est donc rapidement devenu un standard que les IDS commerciaux commencent à intégrer. Cela représente un avantage important, il s'agit là d'une des principales raisons du succès de Snort. En effet, les IDS commerciaux ont tendance

à cacher le contenu de leurs signatures (la séquence précise que recherche le logiciel pour identifier une attaque), cela oblige les clients à dépendre d'eux pour bénéficier de nouvelles règles de détections d'intrusions.

Snort, lui, offre un format ouvert et documenté qui permet d'écrire les règles de son choix.

Un autre avantage est la gratuité au niveau des licences qui permet d'installer des sondes Snort en tout point du réseau. La prolifération de ces sondes peut tout de même rapidement devenir un handicap si nous travaillons sur un réseau plus conséquent que celui de notre étude pour ce qui est de l'étude des rapports.

En effet, l'un des problèmes rencontrés avec Snort est l'obtention de ces rapports même si des interfaces existent bien tel que Acid vu en cours.

Acid permet de générer des rapports à partir des alertes de Snort et de stocker les évènements dans une base MySQL. Acid en lui-même, est néanmoins assez difficile à installer. Nous ne nous y sommes pas risqués et avons préféré générer des rapports au format HTML à l'aide de SnortSnarf.

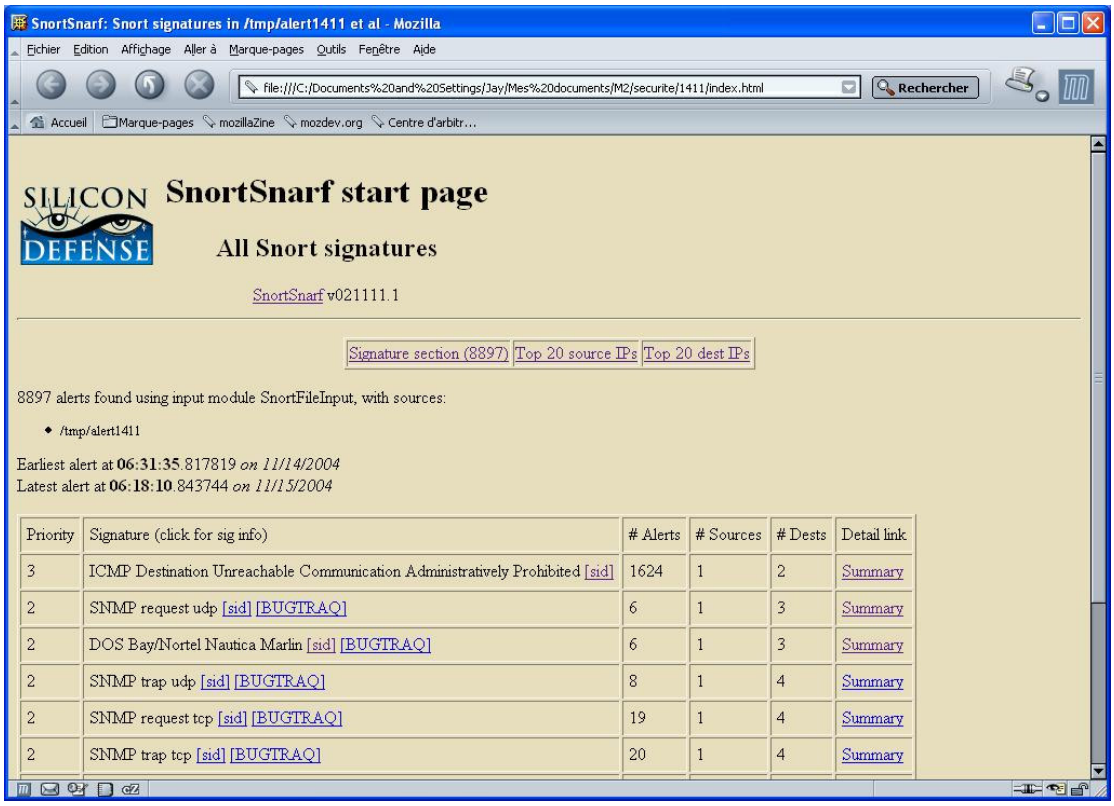
4.2.3 SNORTSNARF

SnortSnarf est un programme Perl mis au point par Silicon Defense pour traiter les fichiers d'alertes générés par Snort, et produire des pages HTML permettant de diagnostiquer et de dépister d'éventuels problèmes.

Il est possible de mettre en place SnortSnarf avec un cron afin d'obtenir un rendu à intervalles journaliers, par heure des alertes de Snort.

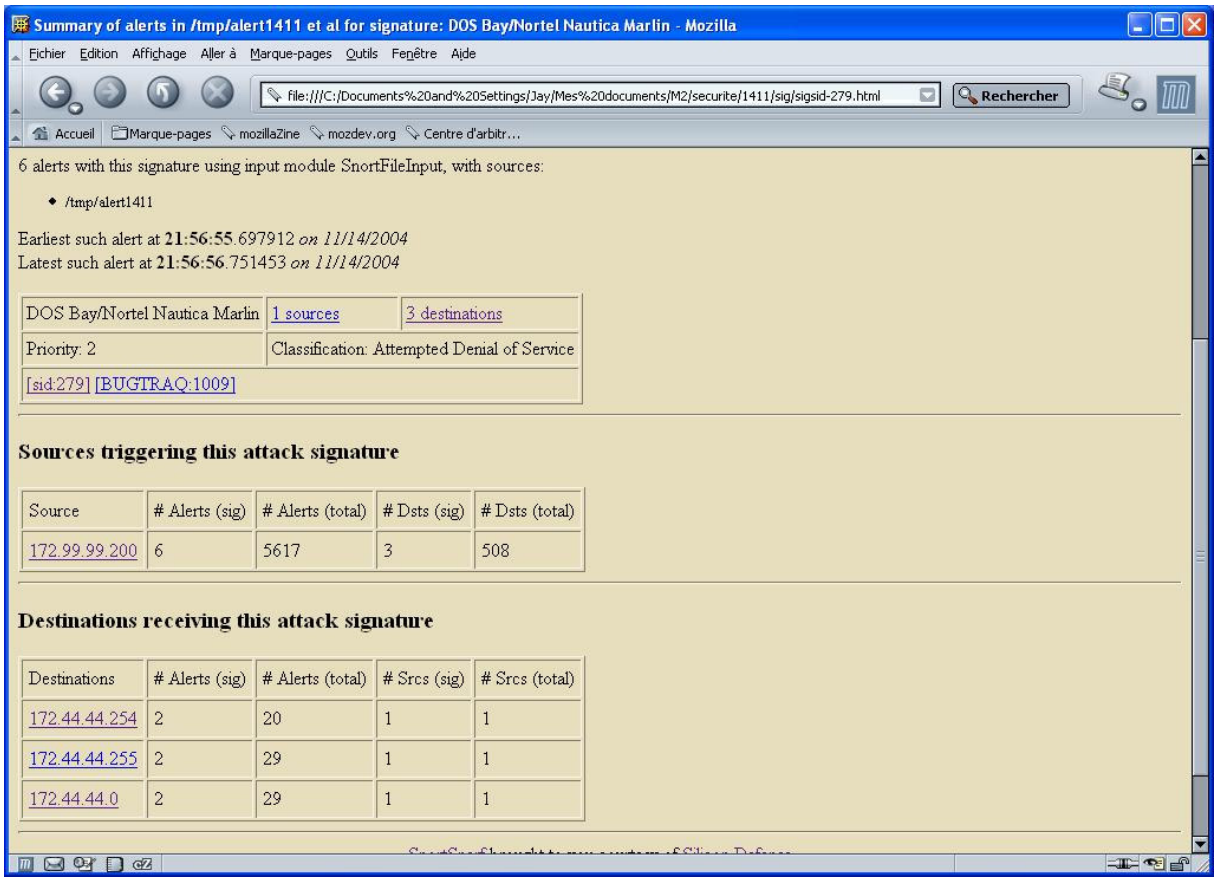
Ce logiciel permet de visualiser des résultats d'alerte de Snort en utilisant un browser HTML, sa mise en place est simple et rapide.

Voici quelques captures d'écrans, illustrant le fonctionnement de SnortSnarf.



Nous pouvons donc obtenir les informations suivantes :

- la priorité de l'alerte,
- la Signature repérée par Snort,
- le nombre de fois où l'alerte est repérée pendant l'étude du log,
- l'adresse source,
- l'adresse destination.



En affinant sur l'alerte DOS, l'adresse IP de la source ("l'attaquant"), nous obtenons les 3 adresses de destinations ("les victimes").

Le sid:279 pointe sur un lien correspondant à la règle utilisé par Snort pour repérer l'alerte.

Le BUGTRAQ:1009, pointe sur la page Securityfocus référençant l'attaque.

4.3 LOGWATCH

Il s'agit d'un outil standard sur Linux qui permet de monitorer l'état du système.

Logwatch étudie les différents logs afin de fournir des informations essentielles sur l'état du noyau du serveur, l'ouverture et la fermeture des sessions, le lancement des tâches en batch, la connexion par ssh, les changements d'utilisateurs (su), l'état de l'espace disque.

4.4 SYSLOG

Syslog permet à certains éléments actifs du réseau d'envoyer des messages vers un serveur faisant lui-même tourner un démon Syslog. Nous avons donc dans la phase 2 pu récupérer les logs des routeurs du réseau, malheureusement le niveau d'information reçu n'était pas suffisamment important pour nous apporter des résultats.

Cette méthode reste tout de même importante et utile dans le cas d'un incident survenant sur le réseau afin de faciliter son dépannage.

4.5 BackOfficer

L'objectif premier d'un pot de miel est d'attirer l'attention de l'attaquant. Pour cela, nous plaçons une machine identique aux autres machines du réseau que nous devons protéger. Le but est de faire croire à l'attaquant que la machine pot de miel est plus vulnérable que les autres, de manière à s'assurer que l'attaquant portera son attention dessus. Une fois les attaques commises, le pot de miel garde les traces des actions effectuées (date et heure de l'attaque, adresse IP source, numéro de port sur lequel l'attaque s'est portée et plus encore en fonction du logiciel). Ces traces ont deux objectifs.

Le premier est de connaître l'identité de l'attaquant : grâce à l'adresse IP source, nous pouvons savoir si l'attaquant est placé dans notre réseau ou s'il attaque de l'extérieur. Dans tous les cas, nous avons des informations sur cette personne mal intentionnée et nous possédons des indices pour la démasquer.

Le deuxième objectif est tout autre : il s'agit de déterminer quelles ont été les attaques et si elles ont réussi ou échoué.

Dans les deux cas, cela nous permet de savoir quels sont les points faibles des machines du réseau et de pouvoir en renforcer la défense.

Après quelques recherches sur le pot de miel que nous allions utiliser, nous avons choisi d'installer Back Officer (produit par NFR Security). Les raisons sont simples : il est facile de prise en main (interface simple et claire) et il est gratuit. A l'origine, Back Officer a été créé dans le but de détecter toute tentative de scannage de ports avec Back Orifice. Il a ensuite évolué pour détecter les tentatives de connexion sur d'autres services tels que telnet, FTP, SMTP, POP3 et IMAP2. Lorsque Back Officer reçoit une connexion sur l'un de ses services (il émule les services que nous lui demandons de surveiller), il va simuler des réponses aux attaquants. Le but étant d'occuper les attaquants en leur faisant perdre leur temps et de nous donner le temps de prendre nos dispositions pour protéger les machines.

5 Tâches et réalisations

5.1 Phase 1

5.1.1 Architecture

Pour pouvoir analyser le trafic dans une architecture réseau, il nous faut insérer dans cette architecture une machine qui puisse récupérer les trames qui nous intéressent. C'est à dire une machine qui puisse capturer le trafic indispensable à une bonne analyse sans pour autant capturer le trafic qui ne nous sert à rien. C'est pourquoi, il faut placer judicieusement cette machine dans l'architecture réseau à analyser.

L'architecture réseau de l'équipe Défense pour la phase 1 se présente comme suit :

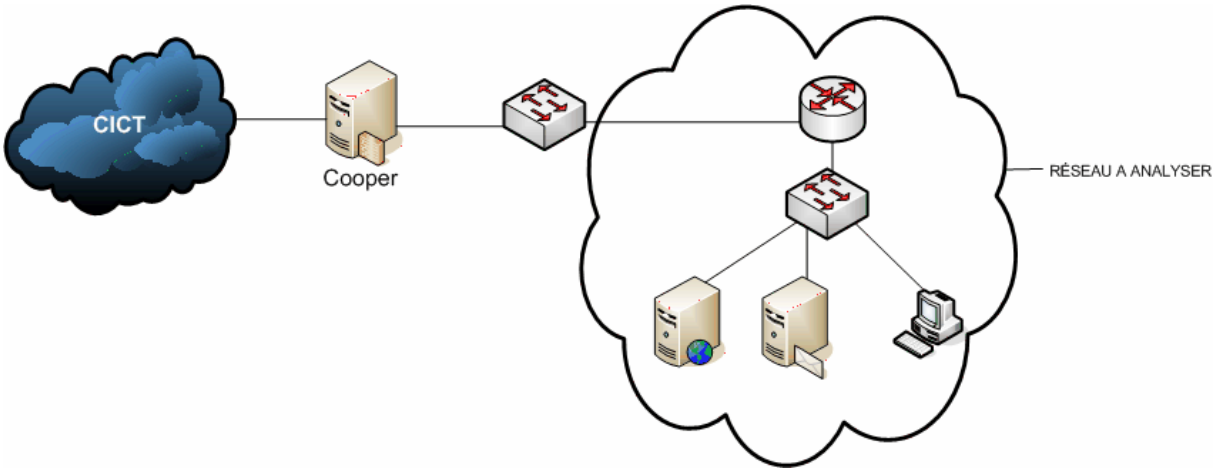


Figure 2 : Architecture du réseau en phase 1

Nous avons donc trois machines qui sont susceptibles d'être attaquées : un serveur web, un serveur mail et un poste client. Ces machines sont toutes les trois reliées physiquement au même commutateur.

Pour pouvoir analyser finement le trafic réseau, l'idéal serait de pouvoir capturer le trafic qui va des machines au commutateur et du commutateur aux machines. Il nous faudrait donc trois sondes qui puissent être insérées entre les machines et le commutateur. Malheureusement, ceci est impossible en l'état et il va donc falloir trouver une autre solution.

Sachant que le trafic d'attaque ne peut venir que de l' « extérieur » du réseau (avant le routeur), l'une des solutions est de mettre en place un sniffer qui capture les trames entre le routeur et le commutateur qui relie les machines. Ainsi nous aurons le trafic à destination des machines ainsi que le trafic en provenance des machines.

Pour cela, nous avons demandé à l'équipe Défense l'autorisation de mettre une machine sur le commutateur et de copier toutes les trames en provenance ou à destination de l' « extérieur » et de les renvoyer vers notre machine (principe du « mirroring »). Ainsi, nous obtenons l'architecture réseau suivante :

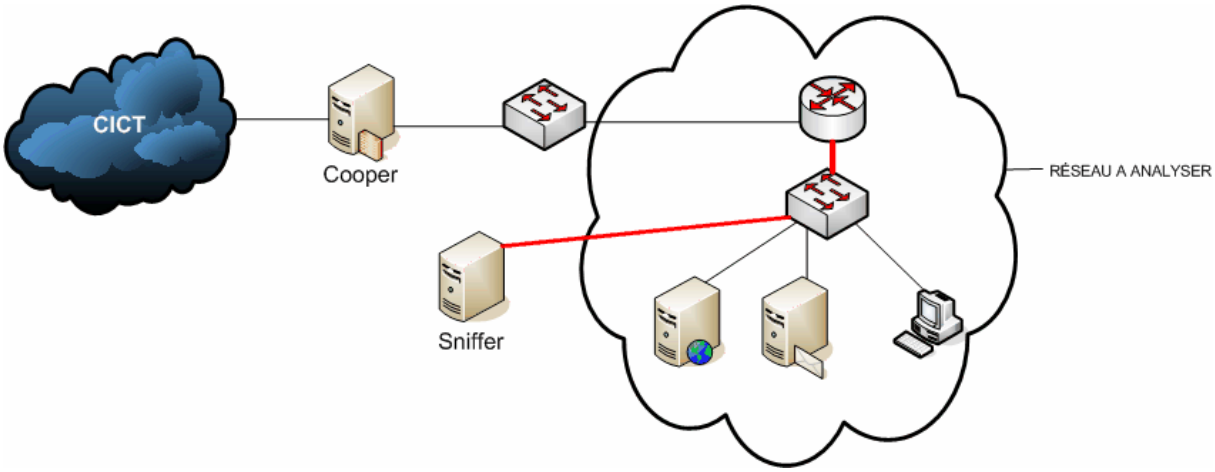


Figure 3 : Architecture du réseau en phase 1 avec sniffer

Toutes les trames qui passent entre le routeur et le commutateur sont donc copiées et renvoyées vers notre machine d'analyse (sniffer).

5.1.2 Déroulement de cette phase

	1ère phase : réseau local non subdivisé					
Tâches	Installation de la première machine sous DEBIAN	Installation des outils de capture	Documentation sur les outils d'analyse et sur les types d'attaques possibles	Mise en place de l'accès à distance sur la machine DEBIAN	Récupération des logs	Documentation sur les types des trames repérées lors de la capture
Objectifs	Insérer une machine sous debian dans le groupe défense	Pouvoir analyser le traffic dans le réseau pendant les attaques	Déterminer les différentes failles potentielles. Orienter nos écoutes du réseau	Pouvoir récupérer les fichiers logs à distance	Analyser les logs afin de repérer le type et la nature des attaques	Repérer les faux positifs ainsi que les attaques tentées
Participants	Jerôme, Sébastien A	Sébastien A, Paul	Jérôme, Sébastien A, Yoann, Lydie		Sébastien A, Jérôme	Sébastien L, Yoann, Lydie
Outils utilisés	DEBIAN préconfigurée	Ethereal, Tethereal, Snort, tcpdump, cron	Documentations sur Internet dont les sites www.cert.org, www.ossir.org, www.isc.sans.org ainsi que divers liens.	SSH	WinSCP, Putty, ethereal, tethereal sur les machines personnelles	Documentations sur Internet dont le site de snort.
Résultats obtenus	Machine correctement "insérée" dans le réseau	Capture des trames à horaires fixes et compressions des fichiers logs	Familiarisation avec les logiciels utilisés, prévisions des attaques	Accès à distance sécurisé	Statistiques des ports les plus souvent attaqués, listes des trames circulant dans le réseau, obtention des attaques potentielles	Repérage des faux positifs

Lors de la première phase (cf figure 1), le premier travail effectué par l'équipe Analyse a été d'installer une machine sous DEBIAN et de mettre en place les outils nécessaires à la capture des trames circulant sur le réseau. En attendant que l'équipe Défense mette la totalité de leur réseau en route, notre équipe a pu se documenter sur les différents logiciels Windows qui permettaient de lire les fichiers de logs récupérés mais aussi sur les « offensives » potentielles que l'équipe Attaque pourrait effectuer.

Afin de nous faciliter la capture des logs, nous nous sommes mis d'accord avec les deux autres équipes pour programmer les attaques qu'à des tranches horaires spécifiques, à savoir de 21h à minuit tous les soirs de la semaine hormis les week-ends. De plus, nous avons mis en place l'accès à distance afin que toute notre équipe Analyse puisse récupérer et analyser les fichiers trace depuis nos machines personnelles.

Après les premières collectes de trames, nous avons fait quelques analyses pour se faire une idée de ce qui circulait sur le réseau (attaque ou trafic normal du réseaux/faux positifs).

5.1.3 Bilan de cette phase

Cette phase nous a donc permis d'appréhender les difficultés à analyser les logs. En effet, il faut acquérir une certaine connaissance du bruit de fond présent sur le réseau avant de pouvoir détecter une intrusion ou une attaque.

Nous avons pu constater dès la première analyse des logs, qu'il y avait ce que nous appelons des faux positifs que nous allons maintenant détailler.

Définition :

Message légitime considéré à tort comme une attaque, ce qui entraîne l'émission d'une alerte (on la retrouve en quantité importante dans nos logs).

Méthode d'identification des faux positifs :

Suite à la capture des paquets, nous effectuons une recherche manuelle des adresses source et des adresses de destination. Connaissant l'adressage de la partie défense (172.66.66.X) et celui de la partie attaque (172.16.8.250 et 172.16.8.251), nous pouvons déterminer les attaques Snort ne venant pas de ces machines. A partir de cela, nous vérifions sur le site www.snort.org que ces alertes possèdent ou non des faux positifs connus. Une fois ces recherches effectuées, nous pouvons distinguer alors les faux positifs et agir en conséquence.

Nous avons pu également remarquer que toutes les alertes web snort ne sont pas des attaques. En effet, Snort les classe en alerte même s'il s'agit d'une simple erreur de frappe d'url ou qu'une option inconnue est présente sur la trace (comme cela peut arriver avec IE).

Moyens mis en œuvre :

Ces faux positifs sont un réel problème pour les analystes car ils sont émis en quantité extrêmement importante (comme nous le verrons dans les exemples qui suivent). En effet nos analyses de logs se sont toujours faites manuellement. Ainsi la présence de ces fausses alertes entraînent une augmentation du temps d'analyse. Nous pouvons comparer cela à de la pollution de logs.

Une configuration adaptée de l'outil Snort permettra de limiter la présence et la classification de ces paquets inoffensifs dans les logs d'alertes.

Exemples de faux positifs obtenus :

```
[**] [1:1781:1] PORN dildo [**]
[Classification: SCORE! Get the lotion!] [Priority: 1]
10/14-21:39:54.334357 66.35.250.210:80 -> 172.66.66.100:1692
TCP TTL:41 TOS:0x0 ID:24909 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x9C3F9608  Ack: 0xB71AE5A7  Win: 0x1920  TcpLen: 20
```

Obtenus une dizaine de fois le 14 octobre (dus à une consultation sur un site web par la défense)

```
[**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3]
10/18-21:01:12.481010 172.16.80.1 -> 172.66.66.81
ICMP TTL:63 TOS:0xC0 ID:4007 IpLen:20 DgmLen:239
```

```
Type:11  Code:0  TTL EXCEEDED IN TRANSIT

[**] [1:396:6] ICMP Destination Unreachable Fragmentation Needed and DF bit
was set [**]
[Classification: Misc activity] [Priority: 3]
10/18-21:09:16.749609 172.16.80.1 -> 172.66.66.6
ICMP TTL:63 TOS:0xC0 ID:55314 IpLen:20 DgmLen:576
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED, DF SET
NEXT LINK MTU: 1256
** ORIGINAL DATAGRAM DUMP:
172.66.66.6:65006 -> 172.16.240.1:3887
TCP TTL:126 TOS:0x0 ID:4597 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xB51FE1D4  Ack: 0x56AD4EDA  Win: 0xF6F1  TcpLen: 20
** END OF DUMP
```

Il s'agit de requêtes ICMP qui sont considérées comme des alertes alors que ces requêtes sont inoffensives (toutes les requêtes icmp ne sont pas inoffensives). Cependant nous avons ce type d'alerte, le 18 octobre, environ 115 fois sur 153 alertes déclarées.

5.2 Phase 2

5.2.1 Architecture

L'architecture réseau de l'équipe Défense pour la phase 2 se présente comme suit :

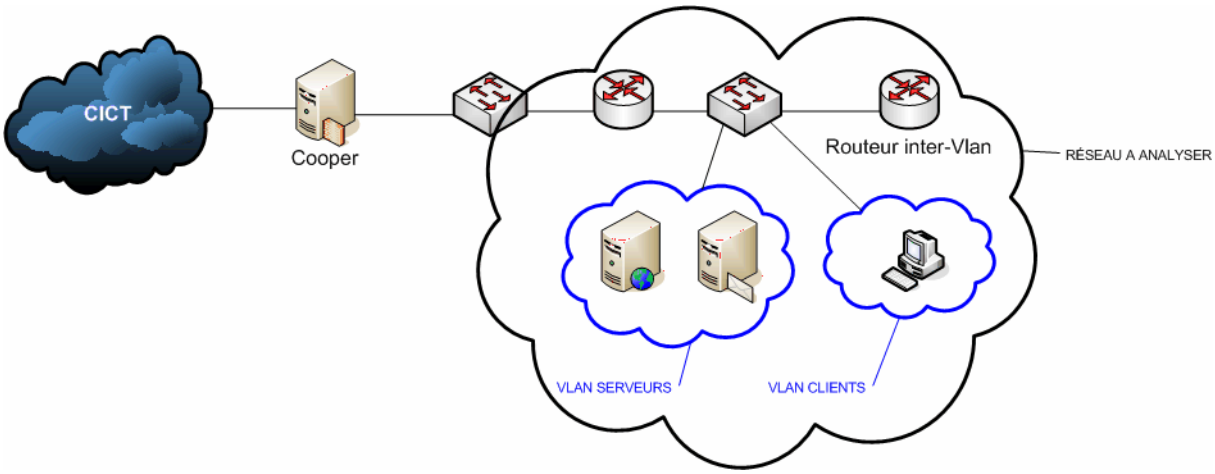


Figure 4 : Architecture du réseau en phase 2

Nous avons toujours les mêmes trois machines qui sont susceptibles d'être attaquées.

Cependant l'architecture réseau logique n'est plus la même. En effet, l'équipe Défense a mis en place des réseaux locaux virtuels (VLAN). Ce qui veut dire qu'il va être mis en place un routage inter-vlan pour faire transiter le trafic réseau comme il se doit.

Pour la mise en place de notre machine d'analyse, la solution précédente est toujours valable. Cependant, pour cette phase, l'équipe Défense a autorisé l'équipe Attaque à insérer une machine dans le réseau à partir de laquelle elle pourra lancer ses offensives sur les

machines de l'équipe Défense. La solution précédente n'est donc plus valable, puisque le trafic interne n'est pas capturé.

La solution serait donc de pouvoir capturer le trafic inter-vlan passant entre le commutateur et le routeur inter-vlan. Le trafic venant de l'extérieur passant obligatoirement par ce lien, les attaques venant de l'extérieur seraient toujours détectées.

Le principe de « mirroring » pourrait donc ici encore fonctionner, c'est à dire que nous demanderions à l'équipe Défense de copier toutes les trames à destination et en provenance du routeur inter-vlan et de les renvoyer vers notre machine d'analyse.

Nous aurions donc l'architecture réseau suivante :

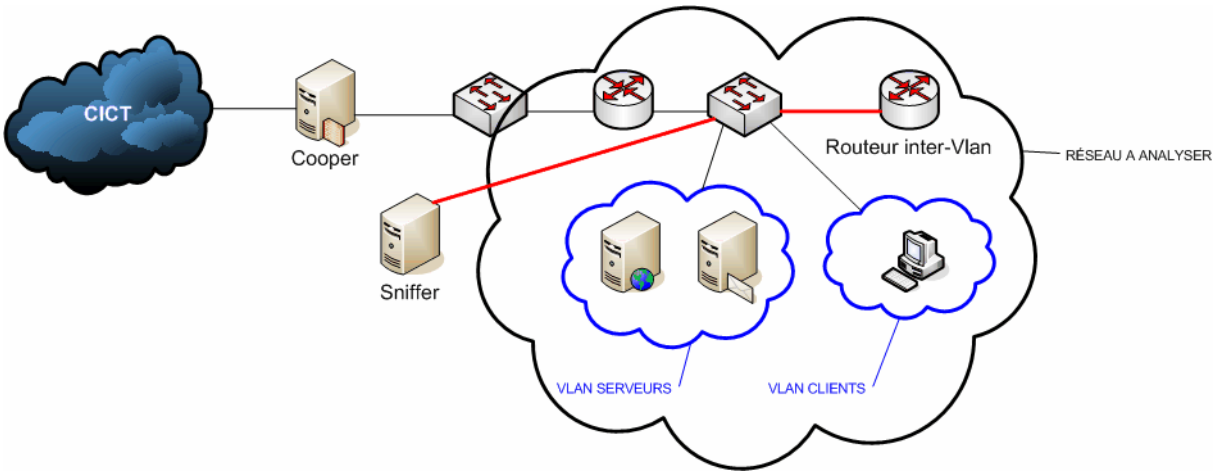


Figure 5 : Architecture du réseau en phase 2 avec sniffer

Malheureusement, cette solution ne fonctionne pas. En effet, le port du commutateur qui est relié au routeur inter-vlan est en mode « trunk » (le trafic passant par ce lien peut-être de provenance de n'importe lequel des VLANs). Avec le commutateur mis en place (et le niveau d'IOS présent sur ce commutateur), il était donc impossible de faire du « mirroring » d'un port « trunk ».

La solution a donc été de mettre en place (avec l'accord de l'équipe Défense) un concentrateur entre le commutateur et le routeur inter-vlan où nous pourrions mettre notre machine d'analyse.

L'architecture réseau est alors la suivante :

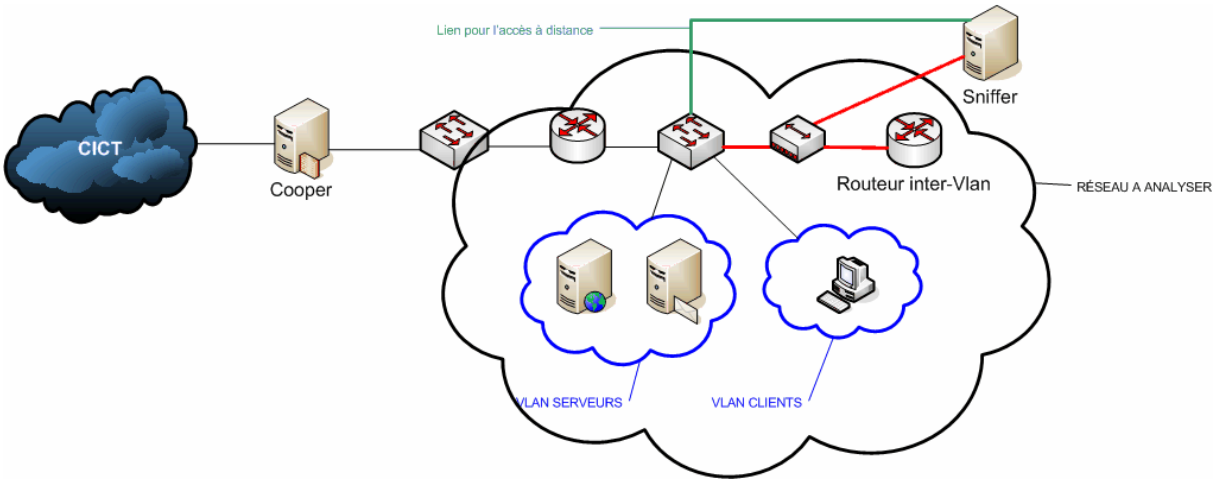


Figure 6 : Architecture du réseau en phase 2 avec sniffer version finale

Nous avons donc le trafic inter-vlan, mais avec cette architecture l'accès à distance à notre machine devient impossible (à cause de l'encapsulation 802.1Q). La solution est de relier notre machine au commutateur via une deuxième carte réseau à travers laquelle nous pourrons y accéder.

5.2.2 Déroulement de cette phase

	2ième phase : découpage du réseau en VLANs				
Tâches	Installation de la machine Windows avec le Honney Pot	Tests pour résoudre le problème de miroring	Documentation sur les autres pots de miel existants, les logiciels permettant de vérifier l'intégrité des données (tripwire par exemple) et les perspectives d'utilisation de pots de miel (utilisation de Vmware)	Récupération et analyse des logs	Mise en place de l'accès à distance sur la machine Windows
Objectifs	Mise en place d'un pot de miel	Faire fonctionner le mirroring pour voir les flux inter VLANs.		Détecter d'autres attaques	Pouvoir administrer le pot de miel
Participants	Anne, Yoann	Paul	Jérôme	Sébastien A., Jérôme	Anne, Lydie
Outils utilisés	Windows 2000 Pro, Back Officer			Snort, Ethereal	VNC, stunnel
Résultats obtenus	Machine correctement "insérée" dans le réseau - Pot de miel mis en place	Résultats des tests : Mauvaise version de l'IOS du Switch, utiliser un HUB pour pouvoir écouter tout le réseau et une carte réseau pour conserver l'accès à distance.	Tripwire : version d'évaluation Vmware : solution trop compliquée à mettre en œuvre		

Après avoir essuyé quelques problèmes de connexion à distance sur notre sniffer, nous avons pu rapatrier les fichiers logs sur les postes d'analyse. A notre grande surprise, les traces ne contenaient aucune trame 802.1Q ce qui prouvait que nous ne capturions pas le trafic inter-vlan. Après plusieurs jours de consultation avec l'équipe Défense, nous avons finalement trouvé une solution mais malheureusement nous n'avons pas enregistré de traces lors de cette deuxième étape du projet.

5.2.3 Bilan de cette phase

Malgré cela nous avons tout de même retiré des enseignements importants de cette phase dans le sens où nous avons ressenti au plus près les difficultés rencontrées par les équipes d'audit sur les sites de leurs clients. En effet, mis à part le côté technique de la tâche qui est fastidieux et qui demande une expertise des analyseurs, le côté relationnel du métier nous est apparu comme une évidence. Il est nécessaire de garder un contact permanent avec le client pour pouvoir anticiper toutes modifications de l'infrastructure aussi minime soit-elle. La communication est une qualité essentielle et c'est pourquoi à la suite de cette expérience nous avons fortement renforcé le contact que nous avions avec l'équipe Défense de manière à éviter par la suite une nouvelle mésaventure. Ce rapprochement nous aura en outre permis de partager nos connaissances et nos solutions avec les administrateurs de la maquette.

5.3 Phase 3

5.3.1 Architecture

L'architecture réseau de l'équipe Défense pour la phase 3 se présente comme suit :

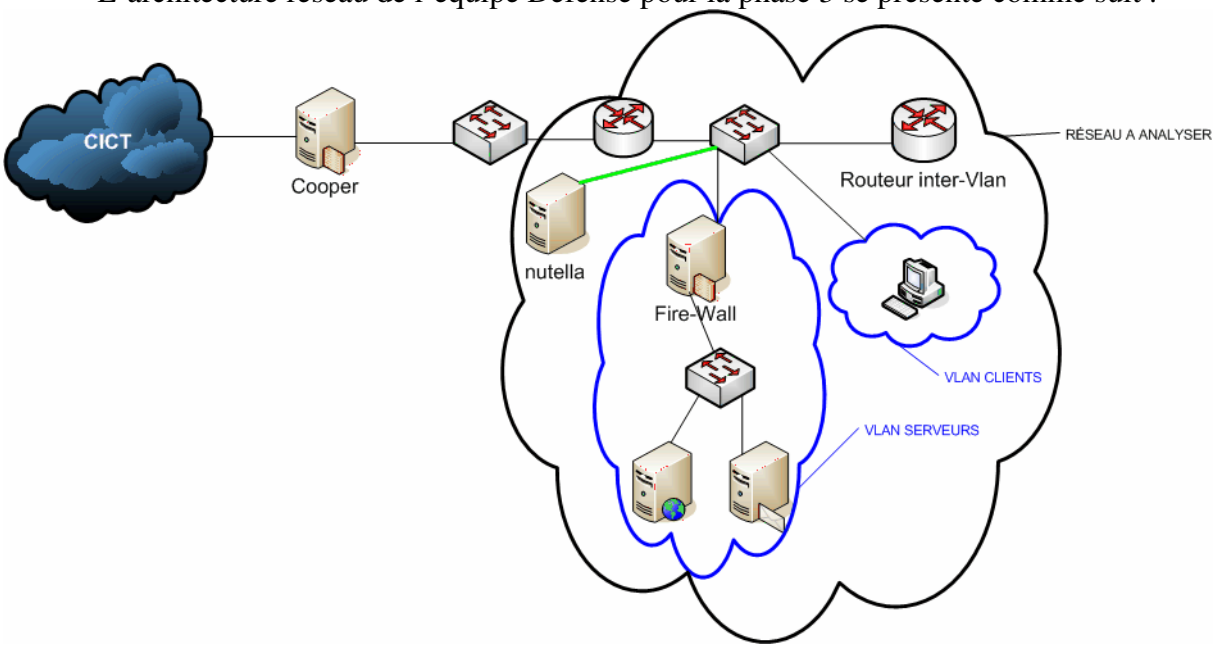


Figure 7 : Architecture du réseau en phase 3

Malgré le pare-feu qui protège le périmètre des serveurs, l'architecture réseau de l'équipe défense est identique à l'architecture réseau trouvée en phase 2. De notre côté, nous avons aussi inséré une machine ayant le logiciel Back Officer. Cette machine nous servira de pot de miel (nutella).

La solution trouvée lors de la phase 2 pour capturer le trafic sera donc ici identique :

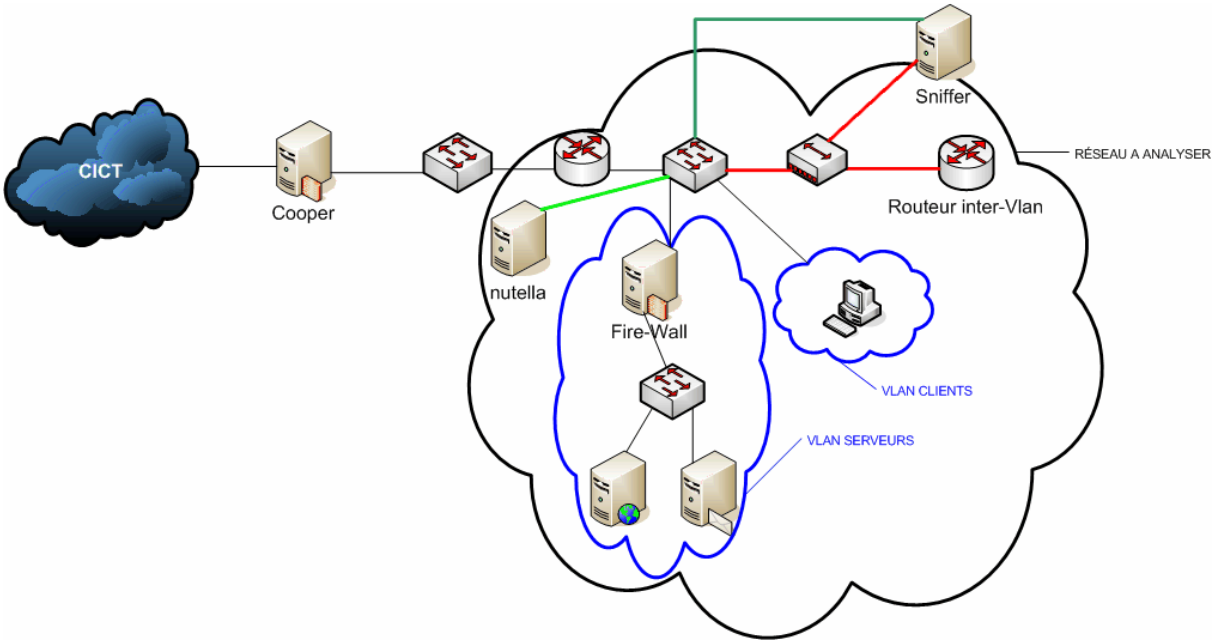


Figure 8 : Architecture du réseau en phase 3 avec sniffer

5.3.2 Déroulement de cette phase

	3ième phase : Rajout du firewall		
Tâches	Accès à distance pour le pot de miel	Installation de SnortSnarf sur la machine d'analyse	Récupération et analyse des logs
Objectifs	Assurer l'accès distant	Mettre en page en HTML les résultats de Snort	Détecter d'autres attaques
Participants	Anne et Yoann	Jérôme	Sébastien A., Jérôme
Outils utilisés	Stunnel, VNC	SnortSnarf	Snort, Ethereal
Résultats obtenus	ok	ok	ok

5.3.3 Bilan de cette phase

Pour l'équipe Analyse, cette phase est la continuité de la phase deux.

Le seul changement visible pour nous a été que les adresses des serveurs n'apparaissent plus dans les traces, mais sont remplacées par l'adresse du FireWall qui sera sollicité suivant le service donné.

Techniquement, cette phase ne nous a pas beaucoup apporté. Mais l'analyse des logs se fait plus rapidement du fait de l'amélioration de notre méthode d'analyse.

6 Politique de sécurité - Audit

6.1 Rapport d'audit sur les échanges

Tout au long de notre projet, nous avons pu capturer des trames et nous les avons donc analysées. Pour chaque alerte que nous avons détectée, nous avons rédigé un compte rendu. Ces différents rapports d'audits sont fournis en annexes.

6.2 Préconisations pour plus de sécurité

6.2.1 Les pots de miel

Lors de notre période d'audit nous avons globalement passé notre temps à capturer et analyser des paquets sur l'ensemble de la maquette.

Sur une courte période d'audit cela reste une méthode valable mais lors d'un fonctionnement habituel et un réseau conséquent cela ne semble pas être une solution très réaliste.

Une alternative, que nous avons esquissée rapidement avec le serveur Windows 2000 et son pot de "nutella", réside dans la mise en place d'une machine ou d'un sous réseau dédié à l'observation et positionné à l'écart du reste du réseau : les paquets parvenant à cet endroit étant forcément à considérer comme suspects.

En mettant en évidences certains services vulnérables (Mail, DNS…) nous disposons d'un appât à destination de l'attaquant.

Nous pouvons alors décider de réunir un jeu de machines physiques sous différents systèmes d'exploitation et un routeur, ce qui était valable pour notre maquette réduite mais deviendrait vite coûteux et ingérable en situation réelle.

Il existe pour pallier à cela, la possibilité de "virtualiser" les ressources. Ainsi les «honeypots» de génération I (filtrage basique de niveau 3) ont fait place à ceux de génération II (interaction avancée avec le niveau 2) parmi lesquels nous distinguons ceux à basse interaction (émulation de services réseaux) et ceux à haute interaction (émulation de systèmes complets).

Le type de pots de miel que nous pouvons mettre en production dépend des résultats que nous désirons obtenir, mais aussi de l'investissement qu'il est possible de consacrer au projet : un pot de miel proposant nombre de services nécessitera une attention plus soutenue en fonction des résultats souhaités. Les résultats que nous pouvons recueillir varient des statistiques, à la capture de paquets ou d'outils.

Quelques outils pour pot de miel :

- **Honeyd**

Honeyd est un petit daemon permettant de créer des hôtes virtuels sur un réseau, de leur assigner une « personnalité » de façon qu'ils apparaissent comme fonctionnant sous un système d'exploitation donné et de leur faire exécuter des services réseaux fictifs.

- **VMware**

Ce logiciel commercial permet d'émuler des architectures matérielles (sur la base d'un processeur x86), il fait cohabiter sur un même ordinateur plusieurs systèmes d'exploitation : Windows, Linux, *BSD ou encore Solaris.

L'interactivité est maximale puisque le piratage peut aboutir à la prise de contrôle totale de la machine virtuelle.

L'idée étant de pouvoir monter une couche linux avec une couche VMWARE émulant un serveur 2000 par-dessus et ainsi de pouvoir traiter les informations résultant de l'attaque sur le serveur 2000 à partir de la couche Linux.

- **Honeywall CDROM**

Distribué par le Honeynet Project, ce CDROM bootable permet le déploiement d'une passerelle destinée à la mise en place d'un « honeynet » de génération II et implémentant des fonctionnalités de contrôle et de capture des données. Nous n'avons malheureusement pas eu le temps de mettre en place cette solution.

6.2.2 Les H-IDS

Quand il s'agit de Network IDS, l'IDS est placé à côté du réseau (hors ligne) et son rôle est de "renifler" (*sniffing*) le trafic sur le réseau puis de fournir alertes et comptes-rendus sur ce qui a été analysé comme tentative ou réussite d'intrusion.

Dans le cas des H-IDS (Host IDS), l'analyse se porte sur les journaux du système et des applications mais aussi sur les logs d'accès aux fichiers.

Le H-IDS se comporte comme un démon ou un service standard sur un système hôte. Traditionnellement, le H-IDS analyse des informations particulières dans les journaux de logs (syslogs, messages, lastlog, wtmp…) et aussi capture les paquets réseaux entrants/sortants de l'hôte pour y déceler des signaux d'intrusions (Déni de Services, Backdoors, chevaux de troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par débordement de buffers…).

Nous avons un temps envisagé de mettre en test la solution proposé par TripWire, nous avons rapidement abandonné cette idée. L'administration qu'elle impose étant très lourde : il faut par exemple refaire la base de données concernant les fichiers sains lors de chaque mise à jour du serveur, voire du poste client.

6.2.3 Les IPS

Les éditeurs et la presse spécialisée parlent de plus en plus d'**IPS** (*Intrusion Prevention System*) en remplacement des IDS « traditionnels » ou pour s'en distinguer.

L'IPS est un Système de Prévention/Protection contre les intrusions et non plus seulement de reconnaissance et de signalisation des intrusions comme la plupart des IDS le sont. La principale différence entre un IDS (réseau) et un IPS (réseau) tient principalement en 2 caractéristiques :

- Le positionnement en coupure sur le réseau de l'IPS et non plus seulement en écoute sur le réseau pour l'IDS (traditionnellement positionné comme un sniffer sur le réseau).

- La possibilité de bloquer immédiatement les intrusions et ce, quel que soit le type de protocole de transport utilisé et sans reconfiguration d'un équipement tierce. En effet, l'IPS est constitué en natif d'une technique de filtrage de paquets et de moyens de blocages (*drop connection*, *drop offending packets*, *block intruder*, …).

Le dispositif de détection d'intrusion est placé "en ligne". Cette différence permet, non plus de rendre compte après coup (constat des dégâts) mais de réagir en temps réel en limitant ou stoppant le trafic douteux par diverses techniques.

Le point sensible de ce genre de dispositif de prévention est qu'en cas de faux positif, c'est le trafic - en temps réel - du système qui est directement affecté. La marge d'erreur se doit donc d'être la plus réduite possible. Certains analystes pensent que le passage de l'IDS à l'IPS n'est qu'un tour de passe-passe pour redynamiser le marché.

7 Bilan

Le bilan de se projet est évidemment conséquent, aussi bien de part son originalité que par les notions professionnelles qu'il intègre. En effet, l'approche métier étant omniprésente, nous avons pu découvrir un ensemble de notions très proches de la réalité telles que la communication ou la mise en place d'un échéancier. L'ensemble des outils utilisés au cours de ce projet, nous a permis d'acquérir non seulement de nouvelles connaissances, mais aussi une certaine « culture » de la sécurité. C'est donc l'ensemble de ces apports qui font de ce projet un sujet intéressant, innovant et complexe à la fois.

C'est à travers toutes ces notions que nous avons pu cerner les différents métiers qui se rattachent à cette activité. Ainsi, chaque approche différente du projet s'est avérée nécessaire et indispensable au bon fonctionnement de l'activité.

Il apparaît très clairement que la réalisation d'un audit de sécurité implique des personnes qui soient capables d'analyser des logs et réaliser des maquettes réseau. En revanche, c'est l'expérience acquise au cours de ce projet qui permet de mettre en avant l'importance du coordinateur ou bien la nécessité d'une personne dédiée à la rédaction des informations.

Le rôle apporté par la personne chargée des relations avec les autres équipes (Défense et Attaque) est primordial, car c'est sur lui que repose l'efficacité de réaction de l'équipe. C'est pourquoi la communication avec le client est essentielle, et nous nous sommes très vite rendus compte qu'il était impératif d'entretenir ces relations afin de travailler ensemble de manière coordonnée.

Afin de mettre en œuvre les outils les plus appropriés à l'architecture de l'équipe défense, l'activité de « veille technologique » s'est avérée capitale. De nouvelles attaques apparaissant tous les jours, l'activité d'audit nécessite une documentation permanente. Nous avons donc commencé par acquérir l'ensemble des notions indispensables de sécurité tels que les sites de référence, puis petit à petit nous avons développé certains réflexes propres au domaine.

Malgré les difficultés que nous avons pu rencontrer lors de la réalisation de l'échéancier prévisionnel, l'équipe a su tirer parti de cet échec, en découvrant petit à petit le temps et la charge de travail nécessaire à chaque tâche. Partant de ce constat, nous sommes aujourd'hui plus à même d'appréhender la gestion du temps dans un projet comme celui-ci.