HAL
open science

Base de traces d'anomalies légitimes et illégitimes

Julien Aussibal, Pierre Borgnat, Yann Labit, Guillaume Dewaele, Nicolas Larrieu, Laurent Gallon, Philippe Owezarski, Patrice Abry, K. Boudaoud

To cite this version:

Julien Aussibal, Pierre Borgnat, Yann Labit, Guillaume Dewaele, Nicolas Larrieu, et al.. Base de traces d'anomalies légitimes et illégitimes. SAR-SSI 2007 - 2nd Conference on Security in Network Architectures and Information Systems, Jun 2007, Annecy, France. 16 p. hal-00349423

HAL Id: hal-00349423

https://hal.science/hal-00349423

Submitted on 9 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Base de traces d'anomalies légitimes et illégitimes[*]

J. AUSSIBAL[α], P. BORGNAT[χ], Y. LABIT[β], G. DEWAELE[χ], N. LARRIEU[δ], L. GALLON[α], P. OWEZARSKI[β], P. ABRY[χ], K. BOUDAOUD[ε]

[α] *LIUPPA-CSYSEC, IUT des Pays de l'Adour,* [β] *LAAS, UPR CNRS,* [χ] *Laboratoire de physique, UMR CNRS-ENS Lyon,* [δ] *ENAC,* [ε] *ESSI-RAINBOW, Sophia-Antipolis*

L'objectif de ce papier consiste à décrire la base d'anomalies créée dans le cadre du projet METROSEC. Cette base contient des traces d'anomalies, légitimes (foules subites, …) et illégitimes (DDoS). Elle permet de calibrer les outils de détection d'anomalies développés dans le projet, et de mesurer leurs performances en termes de faux positifs et faux négatifs.

Mots clés : Base de traces d'anomalies, attaques DoS, Foules subites

---

1. Motivation

L'Internet tend aujourd'hui à devenir un réseau universel de communication. Les nouvelles technologies d'accès ont permis, ces dernières années, d'augmenter très sensiblement le nombre de connectés, et les débits d'accès. Et inévitablement, de nouveaux besoins sont apparus, donnant lieu à la création de nouveaux services sur le réseau. Il y a encore une dizaine d'année, Internet était le support d'applications telles que le Web, les téléchargements par FTP, … Aujourd'hui, le profil du trafic a complètement changé. Les applications se sont très largement diversifiées : web, ftp, « peer to peer », téléphonie sur IP, streaming audio et vidéo pour n'en citer que quelques unes. Toutes ces nouvelles applications ont des besoins en qualité de service (QoS) très variables. Et malheureusement, Internet n'a pas été conçu, à la base, pour répondre à tous ces besoins. Ainsi, le réseau est devenu très sensible aux dégradations de QoS, et plus particulièrement aux attaques de déni de service (DoS), qui sont en forte recrudescence ces dernières années [17]. Ces attaques sont très pénalisantes, et peuvent engendrer des pertes financières importantes. Mais la dégradation de QoS peut aussi être engendrée par des mécanismes tout à fait légitimes, tels que les mouvements de foules, c'est-à-dire la sollicitation d'un service sur l'Internet par un très grand nombre de personnes, suite à un événement extraordinaire (comme par exemple les événements du 11 septembre 2001 aux USA). La détection de ces variations (que nous appellerons par la suite anomalies), légitimes et illégitimes, constitue donc un enjeu de première importance pour les opérateurs voulant garantir la QoS de leurs services.

---

[*] Réalisé dans le cadre du projet METROSEC (ACI Sécurité Informatique 2004-2007), avec le soutien du Conseil Général des Landes.

Les NIDS (Network Intrusion Detection Systems) et NIPS (Network Intrusion Prevention Systems) actuels se basent sur deux principales techniques pour détecter les anomalies du trafic :

- L'utilisation de signatures, c'est-à-dire de formats spécifiques de paquets, ou de successions particulières de paquets, donnant lieu à l'attaque. Cette technique n'est pas bien adaptée à la détection des variations du trafic ne comportant pas de signature particulière (comme les foules subites ou des DDoS sans signature);
- L'utilisation de profils statistiques du trafic. Cette technique est plus appropriée à la détection des anomalies étudiées dans nos travaux. Mais jusqu'à présent, les statistiques utilisées sont d'ordre 1 (moyenne et écart-type). La très forte variabilité naturelle du trafic [18] produit une forte fluctuation de ces mesures, induisant ainsi de très forts taux de faux positifs (fausses alarmes) et de faux négatifs (détections manquées). C'est le reproche majeur fait aux NIDS et NIPS actuels. Des études récentes prennent en compte une forme plus riche de la structure statistique du trafic (corrélation, densité spectrale, ...) [16,19,20,21,22]. Le projet METROSEC de l'ACI Sécurité et Informatique [7] a pour principal objectif la définition de nouveaux outils de détection par profil, basé sur des analyses statistiques d'ordre 2. Ces travaux ont permis de définir des procédures basées sur la caractérisation de marginales du trafic par des modèles non gaussiens, et plus précisément par des lois Gamma-Farima [9]. L'originalité de cette approche réside dans sa nature multirésolution (plusieurs niveaux d'agrégation sont analysés conjointement), qui fournit une statistique robuste, prenant finement en compte la structure de corrélation (court-terme) présente dans le trafic agrégé.

Quel que soit le type de détection utilisé, les nouvelles techniques développées doivent être calibrées et évaluées, c'est-à-dire qu'il faut mesurer leur taux de faux positifs et de faux négatifs, et vérifier quelles anomalies peuvent être détectées. Il est alors indispensable de pouvoir utiliser des traces de trafic, comportant des anomalies bien connues, pour mener à bien ces études. A ce jour, très peu de traces publiques de réseaux opérateurs sont disponibles pour la communauté scientifique, pour des raisons évidentes de confidentialité. On peut néanmoins citer plusieurs exemples :

- La base de traces du LBNL (Lawrence Berkeley National Laboratory) [28] contient des traces de trafic réel, antérieure à l'an 2000. Elles sont maintenant obsolètes car les caractéristiques du trafic ne sont plus les mêmes aujourd'hui. Par exemple les applications P2P, ou bien la téléphonie sur IP n'existaient par encore, et ces applications ont aujourd'hui un fort impact sur la nature du trafic réseau. Il en va de même pour les attaques éventuellement capturées dans ces traces, qui ne sont plus les mêmes aujourd'hui. Notons que cette base a été mise en place pour avoir des traces permettant de modéliser le trafic Internet. La détection d'intrusion n'était pas un objectif du projet ;
- KDD99 [34] contient des traces documentées d'anomalies, mais antérieures à l'an 2000, et donc obsolètes ;
- Les traces Auckland IV [24,25] sont plus récentes (2001), et se rapprochent plus du trafic actuel. Mais ces bases sont peu ou pas documentées sur les anomalies qu'elles contiennent, et donc rendent la mise au point des outils plus difficiles. Elles peuvent par contre servir de tests une fois que les outils de détection sont calibrés ;
- CAIDA [27] a développé une base avec des traces récentes. Les données publiques couvrent plusieurs domaines, comme l'étude des Backscatters [33], de quelques vers, et de scans de réseau. On y trouve aussi des captures de trafic de différents backbones. Toutes ces traces souffrent, elles aussi, de manque de documentation sur les anomalies qu'elles contiennent ;
- Les traces DARPA98 [23,26] sont issues de simulations de trafic réseau, et datent de 1998. Elles ne reflètent plus le comportement actuel du réseau Internet [32];

Globalement, les traces disponibles sont soit obsolètes, et ne reflètent plus la nature du trafic Internet actuel, soit issues de simulation. De plus, ces traces sont trop peu documentées pour pouvoir effectuer une validation correcte des nouvelles techniques de détection d'intrusion. Il est donc

nécessaire de pouvoir créer de nouvelles bases, basées sur du trafic réel, et contenant des anomalies légitimes et illégitimes parfaitement connues et maîtrisées.

L'objectif de cet article est de présenter la base de traces de trafic réel, contenant des anomalies légitimes et illégitimes, créée dans le cadre du projet METROSEC [7], et servant de support à l'étalonnage des procédures de détection développées dans ce même projet. Le paragraphe 2 présente la plateforme utilisée pour capturer les traces, et générer les anomalies. Dans la partie 3, nous présentons les outils utilisés pour générer les différentes anomalies, légitimes et illégitimes. La partie 4 donne un aperçu des traces contenues dans la base. Enfin, la partie 5 conclut cet article, et donne quelques perspectives à ces travaux.

2. Plateforme METROSEC

Une dégradation de QoS peut être provoquée soit par une anomalie légitime, c'est-à-dire par une forte demande de service par des utilisateurs légitimes, soit par une anomalie illégitime, c'est-à-dire une attaque menée par des utilisateurs malicieux, à l'encontre d'une cible (serveur ou réseau). Un outil efficace de détection doit pouvoir faire la différence entre ces deux types d'anomalies. Il est donc absolument nécessaire de posséder des traces des deux types dans notre base, pour pouvoir tester les outils de détection. De plus, il faut avoir un éventail le plus large possible des anomalies (légitimes et illégitimes) que l'on peut rencontrer aujourd'hui. Dans le cadre de ce travail, nous nous sommes limités aux anomalies qui ont un impact sur les ressources réseaux (bande passante, …), et pas directement sur un service ciblé. De plus, notre objectif est d'étudier les anomalies ayant un impact sur la nature du trafic (profil). Nous avons donc volontairement laissé de côté les attaques comportant une signature spécifique (au sens IDS du terme), et plutôt généré des anomalies d'accumulation de paquets. Nous avons utilisé différents outils (HPING, IPERF, TRINOO et TFN2k) afin de recréer des attaques de flooding, tel que le ferait un hacker, ou bien provoqué des phénomènes de « foules subites » pour recréer des anomalies légitimes.

Pour réaliser notre base d'attaque, nous ne souhaitions pas faire de la simulation (NS-2, …), mais capturer du trafic réel. Nous avions donc besoin d'un réseau support de grande envergure, sur lequel nous espérions rencontrer le moins d'attaques non contrôlées possible. Le réseau RENATER (réseau national pour l'enseignement et la recherche) rempli ces conditions. Nous n'avons jamais observé d'attaque au niveau des points supervisés, lors de nos différentes captures, provoquant un refus de service quelconque. De plus, ce réseau est surdimensionné, et donc nos expérimentations n'ont pas d'influence sur son utilisation par les utilisateurs légitimes. En effet, les liens OC-48, qui constituent le cœur du réseau, ont une capacité de 2,4Gbits/s. Les liens vers les réseaux locaux, à la bordure de RENATER, ont des capacités variables. Par exemple, le lien vers le réseau du LAAS-CNRS de Toulouse a un débit de 100Mbits/s. Or le débit utile moyen de ce laboratoire est de 10Mbit/s. Les expérimentations à faibles débits que nous menons n'ont donc, en pratique, aucune incidence sur ce réseau. Lors d'attaques plus massives (volume de données plus important) qui pourraient affecter le comportement des utilisateurs légitimes du réseau, nous utilisons un réseau cible expérimental spécifique (laasnetexp.laas.fr [4]) que l'on peut saturer sans causer de dommage au réseau de production du laboratoire.

Nous avons donc mis en place plusieurs points de captures de trafic sur le réseau RENATER pour pouvoir effectuer nos différentes mesures. Les participants sont le LAAS-CNRS à Toulouse, l'ENS à Lyon, le LIP6 à Paris, l'IUT de Pau et des Pays de l'Adour à Mont de Marsan, l'ESSI à Nice. Le LAAS-CNRS est la cible des simulations d'attaques, les autres sites sont les sources des anomalies (voir figure 1). Les différents sites possèdent, pour la plupart, du matériel de mesure de trafic, constitué d'une station Linux équipée d'une sonde DAG [3]. Ces sondes sont spécialisées dans la capture de trafic réseau. Elles sont équipées de module GPS afin de permettre une synchronisation temporelle de grande précision. Sur chaque site source d'anomalie, plusieurs machines sont utilisées, permettant ainsi de contrôler la puissance de ces anomalies. Leur nombre peut varier entre 2 et 7 pour

les attaques en dénis de services distribués. Une centaine de machines ont été utilisées pour générer les anomalies légitimes de type « flash crowd ».

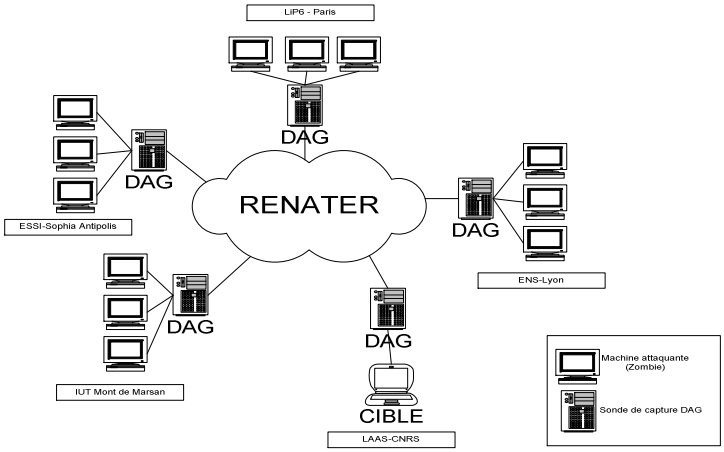


Figure 1 : plateforme METROSEC

Conformément aux lois sur la protection de la vie privée, les traces capturées sont anonymisées. L'algorithme utilisé est décrit dans [31]. Il permet de remplacer les adresses IP et les numéros de ports, dans les entêtes des paquets TCP/IP, par de fausses adresses IP et de faux numéros de ports. On peut ainsi conserver l'anonymat des machines. Cet algorithme à la particularité de garder une cohérence lors de l'anonymisation, en conservant des préfixes communs aux adresses réelles de même préfixe. Par exemple toutes les adresses de préfixe 192.168.25 auront le même préfixe une fois anonymisées (par exemple 10.25.192). Enfin, notons que les captures des paquets ne dépassent pas les entêtes TCP/IP : la partie applicative n'est pas capturée. Les outils de détection basés sur l'utilisation de statistiques d'ordre 2 (Loi Gamma-Farima) développés dans le cadre de ce projet, n'utilisent que des métriques de type nombre de paquets, d'octets, de connexions TCP, … Seule l'entête des paquets est alors nécessaire pour effectuer les différents calculs statistiques.

Lors de nos diverses expérimentations, les machines sources impliquées n'ont jamais appliqué de « spoofing », c'est-à-dire remplacé leur adresse IP par une fausse adresse dans les paquets qu'elles émettaient. En effet, le « spoofing » n'a pas de conséquence sur les métriques que nous utilisons à ce jour. Cela nous permet de repérer facilement les machines attaquantes dans le trafic capturé. Il est évident que ce choix à des conséquences plus importantes sur d'autres types de métriques, comme par exemple le nombre de flux par seconde.

3. Génération des anomalies

3.1 Mise en place des expérimentations

Différents logiciels et techniques ont été utilisés pour créer les anomalies légitimes et illégitimes. Pour les anomalies légitimes, nous avons utilisé deux types de Flash Crowd visant un service particulier : service web via un navigateur Internet et service web via des téléchargements HTTP. Pour les anomalies illégitimes, nous avons fais appel à des logiciels de test de bande passante, Iperf [11] et Hping [13], ainsi qu'à des outils de flooding en DDoS, Trinoo [14] et TFN2k [12]. Ces logiciels sont largement diffusés et accessibles sur Internet. Ils effectuent des envois massifs de paquets sur le réseau. Les paquets envoyés peuvent être de nature différente : on peut spécifier le protocole utilisé (UDP, TCP, ICMP), les différents bits de drapeau de l'entête des paquets, les adresses source et destination mais aussi la taille du contenu du paquet. Chaque combinaison de ces paramètres permet donc de créer une anomalie particulière.

Chaque trace de notre base contient une et une seule anomalie. Mais les captures de trafic réalisées pendant les différentes campagnes ne se limitent pas à la période de génération de

l'anomalie. En effet, il est nécessaire de capturer le trafic « régulier » avant l'anomalie, mais aussi après. La zone pré anomalie permet d'initialiser et de calibrer les outils de détection. La zone post anomalie permet d'observer les conséquences éventuelles de l'anomalie sur les différents services utilisés sur le réseau. Le temps de capture et la durée de l'anomalie sont précisés, pour chaque trace, dans les tableaux du paragraphe 4.

Chaque trace de notre base est documentée. On sait quel type d'anomalie a été généré et quels en sont les différents paramètres : durée de la capture, durée de l'attaque, nombre de machines impliquées dans l'attaque, débit des attaquants, intensité de l'attaque, … Ces informations sont essentielles pour pouvoir tester au mieux les outils de détection.

Pour chaque type d'attaque effectuée, nous avons répété plusieurs fois le scénario d'attaque et donc obtenu plusieurs traces pour la même anomalie (base de traces reproductibles).

L'organisation des acteurs entrant en jeu pour générer une anomalie légitime et une anomalie illégitime est différente. Nous développons ces différences dans les paragraphes 3.2 et 3.3.

3.2 Anomalies légitimes

Les anomalies légitimes que nous avons réalisées sont des « foules subites », c'est-à-dire des requêtes simultanées massives et légitimes auprès du serveur web du LAAS-CNRS (http://www.laas.fr) [22]. Pour la réalisation de ces expérimentations, nous avons fais appel à un maximum de personnes physiques, afin d'avoir l'impact le plus important possible au niveau de la machine victime, et d'avoir un profil d'anomalie le plus vraisemblable possible. Ce type d'anomalie que nous avons réalisé est difficilement contrôlable, et donc elle peut avoir un impact sur la QoS du réseau.

Deux types de foules subites ont été réalisés. Le premier correspond au comportement d'utilisateurs légitimes consultant massivement des pages web sur un serveur Internet. De tels comportements arrivent lors de l'annonce d'évènement de grandes envergures, tels que les attentats du 11 septembre 2001 ou bien l'explosion de l'usine AZF à Toulouse. Les sites Web d'information sont alors submergés par les demandes, et subissent un véritable déni de service distribué « légitime ». Le second type de foule subite est le téléchargement massif d'un ou plusieurs fichiers sur un serveur. Ce phénomène peut arriver, par exemple, lors de la mise à disposition d'une dernière version d'un système d'exploitation, ou d'un logiciel, sur un serveur de téléchargement. La différence principale entre ces deux types de Flash Crowd réside au niveau de la nature même du trafic. Lors de consultations Web, le trafic généré est de type impulsionnel. Lors d'un téléchargement, le trafic généré est continu. Ces deux types de Flash Crowd permettent donc d'obtenir deux anomalies légitimes avec du trafic aux caractéristiques sensiblement différentes.

Pour les simulations de foules subites via un browser web, les participants ont consulté, dans un période de temps donné, le plus grand nombre de pages du site web du LAAS-CNRS. Le nombre précis de participants est une donnée que nous n'avons pu contrôlée. En effet, nous avions diffusé un message de participation à la Flash Crowd dans plusieurs laboratoires, mais sans savoir exactement qui allait réellement participer à l'expérience. Cependant, fixer a priori ce nombre n'est pas, de notre point de vue, l'élément essentiel de ces expérimentations. L'objectif principal est de créer une variation significative du volume du trafic et du nombre de requêtes HTTP sur le serveur Web cible.

Pour la réalisation de foules subites de téléchargement massif d'une même ressource, un certain nombre de participants ont rapatrié le(s) fichier(s) mis à leur disposition pour l'opération. Ces téléchargements peuvent être faits au travers de diverses applications utilisant un protocole particulier (HTTP, FTP, Torrent,…). Pour notre expérience, nous avons utilisé le protocole HTTP.

La réalisation de ces expériences, réunissant un grand nombre de participants, n'est pas chose aisée. Aussi, la base de trafic METROSEC ne dispose, pour le moment, que très peu de captures de trafic contenant des anomalies légitimes.

3.3 Anomalies illégitimes

Il existe une multitude d'attaques différentes en déni de service qui ont été classées en différentes taxonomies [1,2]. On peut séparer les attaques visant un service particulier (exploit attacks), par exemple le service Web ou le service SSH, des attaques visant les ressources réseaux (network attacks), c'est-à-dire soit la bande passante de la cible, soit des ressources plus spécifiques, comme la pile de connexion semi-ouvertes de TCP. Dans le cadre du projet METROSEC, nous nous sommes intéressés aux network attacks, et ce pour deux raisons. Tout d'abord, l'objectif initial du projet était de pouvoir faire de la détection proche des sources d'attaque, i.e. chez les opérateurs par exemple, et pas sur le réseau cible de l'attaque (si l'attaque parvient à une telle proximité de sa cible, elle a déjà consommé beaucoup de ressources, dégradé la QoS, et est donc un succès). Cependant, il est très difficile et très coûteux, sur des réseaux d'opérateurs, d'analyser individuellement chaque flux de données (chaque flux correspond à l'utilisation d'un service particulier), pour savoir s'il contient ou non des attaques. Souvent, on ne peut que traiter le trafic de façon globale (agrégations des flux), sans forcément différencier ni les cibles, ni les services. Les deux grands types d'attaques cités précédemment ne sont alors pas différenciés. La seconde raison est que ces attaques ont un impact sur la QoS beaucoup plus important que des attaques ciblées : tous les services sont atteints par une consommation excessive de bande passante ! Il nous a donc semblé opportun de traiter ces attaques en premier lieu, et, dans un futur proche, de passer à des attaques plus ciblées.

Parmi les attaques les plus connues du second type, on trouve les attaques de type flooding. Les principales sont l'UDP flooding, l'ICMP flooding et le TCP SYN flooding. L'objectif de ces attaques est d'envoyer un maximum de paquets à la victime afin de réduire fortement la QoS de la victime ou du réseau de la victime.

Les premières campagnes d'expérimentations ont été effectuées en utilisant des logiciels de test de bande passante, largement connus des administrateurs réseaux : Hping [13] et Iperf [11]. Ces logiciels sont utilisés pour envoyer un grand nombre de paquets sur un lien, et vérifier la valeur du débit utile offert. Il s'agit donc de logiciels de flooding, mais à but de test. Nous les avons utilisés comme logiciels d'attaque, en envoyant le plus de paquets possibles vers la cible de nos attaques, à travers le réseau RENATER. Les campagnes suivantes ont utilisé de « vrais » logiciels de flooding. Plus exactement, nous avons utilisé des logiciels de DDoS : Trinoo [14] et TFN2k [12]. Ces logiciels nous ont permis de générer des attaques les réalistes.

Le logiciel Hping [13] fonctionne de la même manière que l'utilitaire Ping, à la différence que celui-ci ne se limite pas à l'utilisation du protocole ICMP. Il permet d'utiliser les protocoles ICMP, TCP et UDP. Le fonctionnement de l'application Iperf [11] est celui d'une architecture client/serveur. Le serveur est la machine à attaquer et les clients sont les machines corrompues qui vont surcharger la bande passante du serveur. Iperf (tout comme Hping) n'est à l'origine pas prévu pour réaliser une attaque en DDoS. Cependant si le nombre de clients est conséquent alors le serveur subira une attaque de flooding distribuée.

Pour ces deux logiciels, on peut quantifier le volume de données à envoyer en manipulant deux paramètres : le nombre de paquets envoyés par seconde et la taille des données (payload) de chacun de ces paquets. Le logiciel Iperf peut en plus modifier les flags des paquets TCP. Par conséquent on peut changer la signature de l'attaque à sa guise (nous n'avons pour l'instant pas utilisé cette possibilité dans nos travaux). L'inconvénient des ces deux logiciels est que la synchronisation des zombies distants n'est pas possible. Dans nos simulations, une personne par site effectue le lancement du logiciel à un horaire prédéfini. En conséquence, l'attaque fait apparaître une montée en charge plus ou moins progressive au niveau de la victime.

Les deux autres logiciels que nous utilisons (Trinoo [14] et TFN2k [12]) ont été développés pour générer des attaques en DDoS. Leur principe est de distribuer une application malicieuse (zombie ou bot) sur des machines corrompues, qui pourront ensuite être manipulées par l'application « maître » (master). L'ensemble des machines corrompues (« botnet ») forme une véritable armée que le hacker peut commander à sa guise (à partir d'un ou plusieurs master) pour déclencher une attaque, et ce de manière synchrone. Ces deux logiciels génèrent le maximum de trafic possible sur le réseau. Pour pouvoir contrôler les débits des différents zombies, nous avons mis en place, sur chaque machine corrompue par un zombie, une interface virtuelle, sur laquelle on peut

spécifier le débit voulu. Ainsi, dans toutes nos expérimentations, nous contrôlons l'intensité de l'attaque.

Il faut noter ici que les IDS commerciaux basés signatures sont parfois capables de détecter la présence de ces logiciels sur un réseau. En fait, les paquets recherchés sont les paquets de communication entre les masters et les zombies (l'envoi des ordres par le master aux zombies), qui comportent des signatures très spécifiques. Par contre, ces IDS ne peuvent pas détecter le trafic généré par un zombie, et à destination de la cible, qui ne comporte pas de signature particulière. C'est ce trafic qui nous intéresse dans le cadre de ce projet.

Le logiciel Trinoo est conçu pour faire uniquement de l'UDP flooding à destination de la machine cible. Les paramètres que l'on peut spécifier sont la durée de l'attaque et la taille des paquets envoyés. Le programme « master » peut aussi indiquer aux différents zombies d'attaquer plusieurs cibles simultanément.

Le logiciel TFN2k a la possibilité de générer une plus grande variété d'attaques. Contrairement à Trinoo qui est limité à de l'UDP flooding, le bot TFN2k est capable de générer des attaques en UDP, ICMP et TCP SYN flooding. Les deux premières attaques ont pour but de saturer la bande passante de la machine cible. L'attaque en TCP SYN consiste à utiliser une vulnérabilité de TCP, en ouvrant un maximum de connexion TCP afin de saturer la pile de connexions semi-ouvertes TCP. Pour ce faire, un large nombre de paquets TCP avec le flag SYN sont envoyés à destination de la victime. Le zombie TFN2k est aussi capable de générer des attaques MIX flooding et des attaques SMURF. Le MIX flooding consiste à mélanger les 3 types de flooding (UDP, ICMP et TCP SYN). Smurf est une technique d'attaque par amplification : les bots se servent d'adresses de broadcast pour multiplier artificiellement le nombre de paquets d'attaque destinés à la cible, et donc multiplier la puissance de cette attaque. De plus, contrairement au logiciel Trinoo, on peut modifier les adresses IP (spoofing) des machines corrompues, rendant plus compliqué la recherche les machines incriminées depuis la cible. Cette fonctionnalité n'a pas été utilisée dans nos traces, afin que chaque machine attaquante conserve toujours la même adresse IP, même après anonymisation.

Pour élargir un peu plus notre éventail de captures, nous avons apporté des modifications au bot TFN2k. En effet, le bot TFN2k, tout comme le bot Trinoo, ne font à l'origine que du flooding continu pendant toute la durée de l'attaque. Les modifications apportées permettent d'envoyer des rafales de paquets (bursts), générant ainsi une attaque discontinue. Plusieurs paramètres peuvent être maîtrisés : durée du pic dans le temps, nombre de pics dans le temps, durée du temps d'attente entre deux pics successifs, répétition périodique ou apériodique des pics, variation de l'intensité de deux pics successifs. Ce type d'attaque de flooding discontinu a été récemment mis à jour dans [29,30].

4. Base d'anomalies

4.1 Anomalies légitimes

Logiciel	Type d'anomalie	Durées traces	Durées anomalie	Intensités
Campagne Flash Crowd de avril 2005				
Navigateur web	HTTP GET Requests	2h	38mn	34%
Navigateur web	HTTP downloads	1h30	20mn	71%

Tableau 1 - Description des Flash Crowd dans la base

Un outil efficace de détection d'attaques doit pouvoir faire la différence entre une anomalie légitime, par exemple une foule subite («flash crowd »), et une anomalie illégitime (une attaque). Il est donc nécessaire de posséder les deux types de traces dans notre base. Pour le moment les seules traces de Flash Crowd présentes dans notre base sont des foules subites qui utilisent le protocole HTTP (voir Tableau 1).

La première trace correspond à une consultation massive du site web du LAAS (www.laas.fr). Elle a été réalisée le 14 avril 2005. Afin de conserver au maximum les caractéristiques réelles d'une telle anomalie, nous n'avons pas utilisé d'outil automatique de consultation, mais fait

appel à des utilisateurs humains. Le nombre total de personnes impliquées s'approche de la centaine. Chaque utilisateur avait reçu comme consigne de consulter le site web de 14h30 à 15h, en effectuant le maximum de requêtes possibles. Cet horaire a été choisi afin de réaliser notre expérience dans une période de trafic nominal du LAAS. La capture du trafic du LAAS a été effectuée entre 13h45 et 15h45. Le trafic à destination du serveur web représente un tiers des paquets total du LAAS au moment de l'anomalie. C'est cette puissance de l'anomalie qui est représentée par le champ « intensité » du Tableau 1. Elle correspond à la proportion de paquets d'anomalie dans le nombre total de paquets circulant sur le réseau du LAAS.

La Figure 2 (a) fait apparaître le nombre de requêtes HTTP arrivant et sortant du LAAS. Vers la seconde 2600 de la capture (soit 14h30), on voit le nombre de requêtes externes augmenter fortement. C'est le début de la foule subite. De plus on peut observer que le nombre de requêtes HTTP a vite décliné pendant la période prévue. On peut en conclure que les différents acteurs ont été très actifs au départ mais se sont essoufflés rapidement. Comme on aurait pu le deviner, sur la figure 2 (b) et 2 (c), on peut aussi observer une légère augmentation du nombre moyen de connexions et de paquets par seconde pendant la Flash Crowd. Cependant sur ces deux dernières figures ont peut distinguer deux anomalies qui ne sont pas dues à nos actions. La première apparaît avant la Flash Crowd sur la figure 2 (c), on peut distinguer deux augmentations du nombre de paquets par seconde. La seconde est une anomalie post Flash Crowd que l'on observe sur la figure 2 (b). Grâce à une analyse du trafic avec l'outil Traffic Designer de la société QoSMOS [8], on peut déterminer la nature de ces autres anomalies présentes dans la trace. L'anomalie visible sur la figure 2 (c) est due à la reprise d'activité des chercheurs sur le web après la pause déjeuner. L'autre anomalie sur la figure 2 (b) est un pic de trafic SMTP. Deux raisons peuvent être avancées. Tout d'abord, beaucoup de chercheurs du LAAS utilisent Webmail. La dégradation de la QoS du serveur Web a donc aussi atteint le service de courrier électronique. Les personnes du laboratoire ont donc utilisé Webmail en masse, dès qu'il a été de nouveau accessible avec une QoS acceptable. De plus un mécanisme de *grey listing* (utilisé pour réduire les spams) retarde certains emails et les envois lors d'horaires planifiés. La première ouverture planifiée sur le serveur correspond au pic post Flash Crowd.

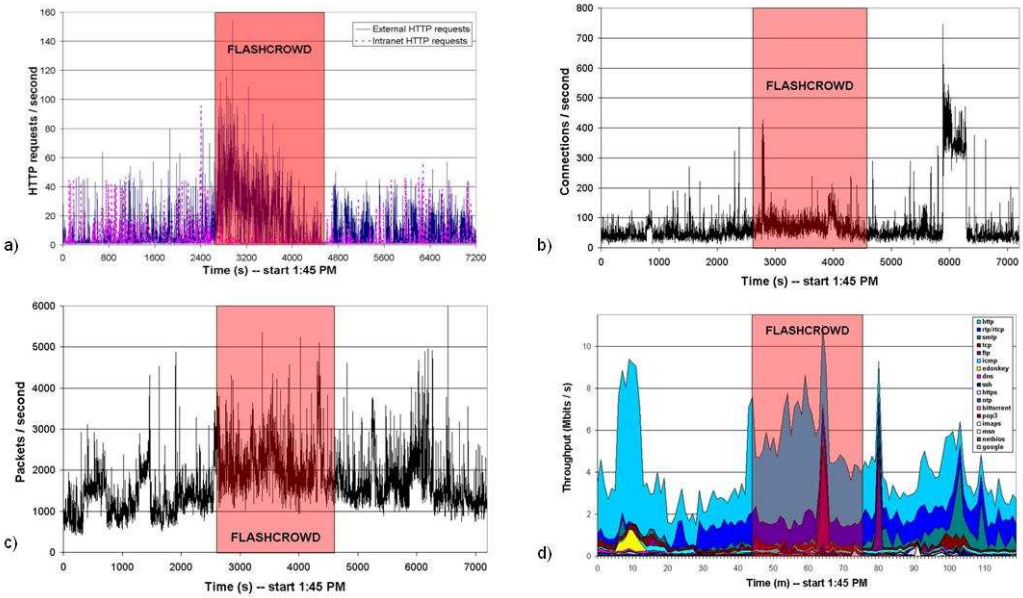


Figure 2 - Trace avec Flash Crowd HTTP GET requests. (a) Nombre de requêtes HTTP, (b) de connexions, (c) de paquets et (d) distribution des débits par applications (en Mbits/s).

La seconde Flash Crowd disponible dans notre base correspond à un téléchargement massif de fichiers situé sur le serveur web du LAAS. Tous les téléchargements ont débuté à la même heure

et ont été interrompu au bout de 20 minutes. Les téléchargements ont été effectués par plusieurs sources distinctes et ont eu un impact important sur le trafic global du LAAS : plus de 70% du trafic était du à ces téléchargements. Pour des raisons de place, nous ne développons pas cette anomalie ici.

4.2 Anomalies illégitimes

Les anomalies illégitimes représentent la majorité des traces contenues dans la base. Les objectifs de nos expérimentations étaient multiples :

- Avoir des attaques avec des intensités fortes, pour permettre la calibration des outils de détection ;
- Avoir des attaques avec des intensités faibles. Ces expériences sont les plus nombreuses dans la base. Elles peuvent correspondre à la détection sur un réseau opérateur proche de certaines sources de l'attaque, quand tous les flux attaquants ne sont pas encore agrégés. On peut aussi imaginer que ces expériences peuvent correspondre à une détection sur le réseau cible, mais avant que l'attaque soit complètement avérée, c'est-à-dire quand seuls quelques flux attaquants commencent à arriver sur la cible ; Ces traces doivent permettre de mesurer les performances des outils de détection ;
- Avoir des attaques distribuées, c'est-à-dire avec de multiples attaquants ;
- Avoir des attaques utilisant des logiciels spécialisés dans le déni de service distribué, afin de conserver le maximum de vraisemblance dans nos traces.

Cinq campagnes d'expérimentations ont été effectuées. Chaque campagne est constituée de plusieurs expérimentations utilisant le même logiciel d'attaque, mais avec des paramètres différents (intensité, …). Les différentes figures présentées ci-après représentent l'évolution du nombre de paquets par seconde au cours du temps.

Les deux premières campagnes ont été effectuées en utilisant les logiciels Hping et Iperf. Les deux campagnes suivantes ont utilisé Trinoo et TFN2k. Enfin, la dernière campagne a utilisé une version modifiée de TFN2k, qui permet d'effectuer du flooding discontinu. Nous détaillons un peu plus en détails chaque campagne dans la suite de ce paragraphe.

Campagne Hping (novembre - décembre 2004)

Logiciel	Type attaque	Durées traces	Durées attaques	Intensités
Campagne de novembre-décembre 2004				
Hping	TCP flooding	1h23mn 3h3mn 30mn	15mn 13mn 3mn 7mn 8mn 5mn	30,77% 27,76% 90,26% 70,78% 45,62% 91,63%
	UDP Flooding	16h20mn	5mn	99,46%

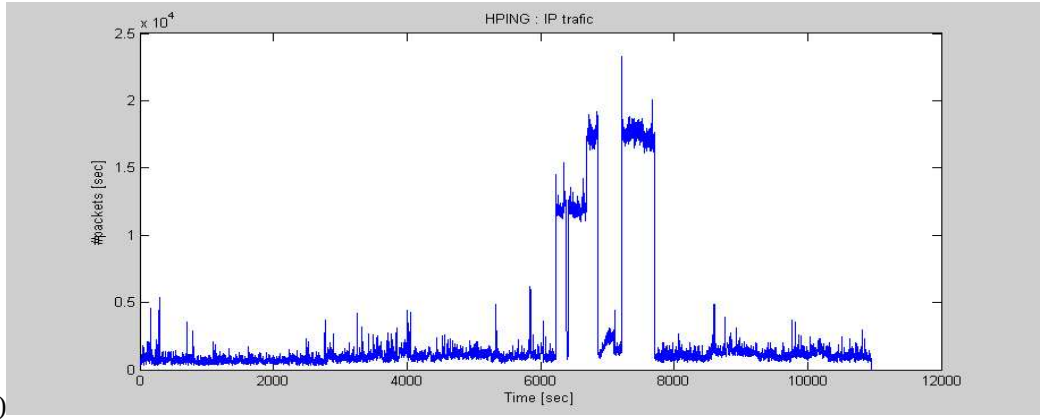
Tableau 2 - Description des attaques contenues dans la base

Lors de la première campagne de génération d'anomalies illégitimes, nous avons utilisé le logiciel HPING. L'objectif de cette campagne était de pouvoir calibrer nos outils de détection. En conséquence, les intensités des attaques étaient fortes, et l'impact sur la cible était important. La liste des expérimentations effectuées est donnée dans le tableau 2.

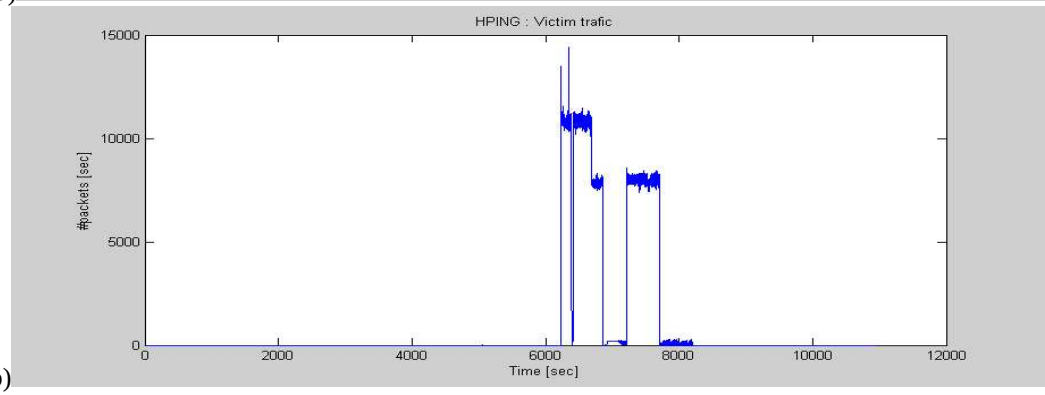
HPING nous a permis de réaliser des attaques en UDP et TCP flooding. Nous détaillons maintenant l'une d'entre elles. Il s'agit de la trace contenant trois attaques successives en TCP flooding, en provenance d'un seul zombie HPING (2ème ligne du tableau 2). Cette trace a une longueur de 3h et 3mn. Les intensités des 3 attaques sont respectivement de 90,26% pour la première, 70,78% pour la seconde et 45,62% pour la dernière. Les trois attaques ont des durées variables, et sont séparées par une période d'activité normale du réseau.

Sur la figure 3 (a), on peut voir le trafic TCP généré par le bot Hping à destination du service SSH de la machine cible du LAAS-CNRS. Lors de la seconde attaque, on peut noter que la puissance diminue fortement au milieu de l'attaque. Lors du démarrage de cette expérimentation, le service SSH sur la machine cible n'était pas lancé. La coupure au deux tiers de la seconde attaque correspond au moment où le service SSH a été activé sur la cible. Cette machine a donc commencé à répondre aux requêtes de l'attaquant, et a consommé une partie de la bande passante du réseau de cet attaquant. En conséquence, le nombre de paquets d'attaque émis par l'attaquant a diminué. La figure 3 (b), qui représente le trafic total circulant sur le réseau de la cible, montre bien qu'à partir des deux tiers de la seconde attaque, le nombre de paquets circulant sur le réseau augmente très sensiblement.

Il faut noter que lors de cette attaque, le service SSH de la cible n'a pas été affecté : en effet, dès qu'il a été activé, il a pu répondre à toutes les requêtes. Nous n'avons donc pas observé de dégradation sensible de QoS, malgré une intensité assez forte. On est donc bien dans une situation où il faut détecter l'attaque, avant qu'elle n'ait affecté la QoS du service attaqué.



(a)



(b)

Figure 3 – Trace avec attaque Hping. (a) Trafic d'attaque TCP et (b) trafic total TCP

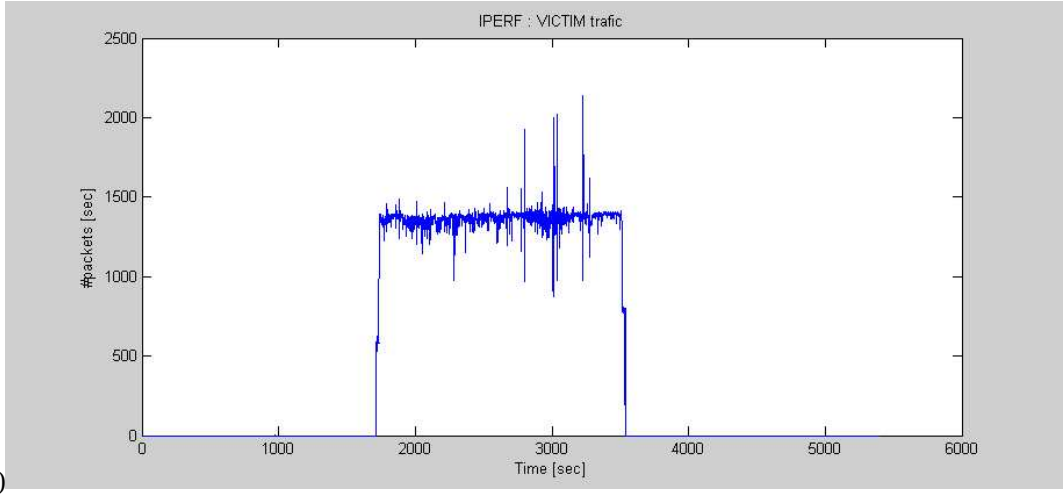
Campagne Iperf (juin 2005)

Logiciel	Type attaque	Durées traces			Durées attaques			Intensités		
Campagne de Juin 2005										
IPERF	UDP Flooding	1h30	1h30	1h30	30mn	30mn	30mn	17,06%	14,83%	21,51%
		1h30	1h30	1h30	41mn	30mn	30mn	33,29%	39,26%	34,94%
		1h30	1h30	1h30	30mn	30mn	30mn	40,39%	36,93%	56,40%
		1h30			30mn			58,02%		

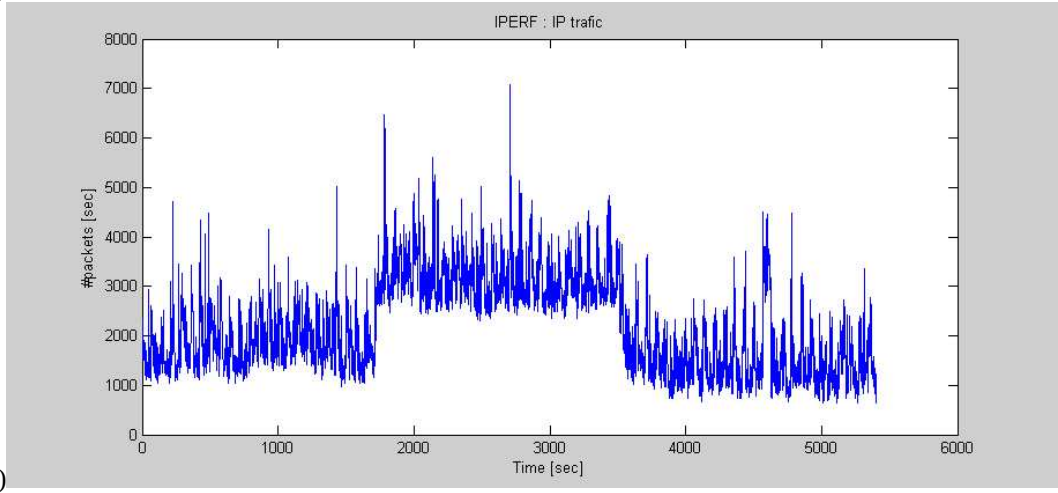
Tableau 3 - *Description des attaques contenues dans la base*

La deuxième campagne a été effectuée en utilisant l'outil IPERF. Cet outil ne sait faire que des attaques en UDP flooding. Pour cette campagne, nous voulions avoir un éventail de captures avec des anomalies d'intensités différentes, et plus faibles que lors de la première campagne. Les intensités obtenues s'échelonnent donc de 14,83% à 58,02%. Contrairement à la première campagne, nous avons utilisé plusieurs attaquants, situés sur des sites géographiques différents : LIP6, ENS Lyon, IUT de Mont de Marsan, ESSI à Nice.

Dans l'exemple de la figure 4, la trace ne contient qu'une seule attaque. Elle est perpétrée par trois clients (bots) situés à Paris, Lyon et Mont de Marsan. Sur la figure 4 (a), on peut voir le nombre de paquets envoyés depuis ces trois sources. Cela représente environ 1375 paquets par seconde. Chaque paquet a une taille de 1500 octets. Le volume total moyen de données destinées à flooder la victime est donc de 15,75 Mbits/s. Cet envoi massif de datagrammes UDP modifie « visuellement » l'aspect du trafic circulant sur le réseau du LAAS-CNRS : on peut facilement distinguer le début et la fin de l'attaque sur la figure 4 (b).



(a)



(b)

Figure 4 - Trace avec attaque Iperf. (a) Trafic total des bots et (b) Trafic total du LAAS

Campagne Trinoo – TFN2k (mars à avril 2006)

L'objectif de cette campagne est de capturer du trafic comportant une attaque effectuée par un logiciel de dénis de service distribués : TRINOO ou TFN2k. Pour cela, il a d'abord été nécessaire

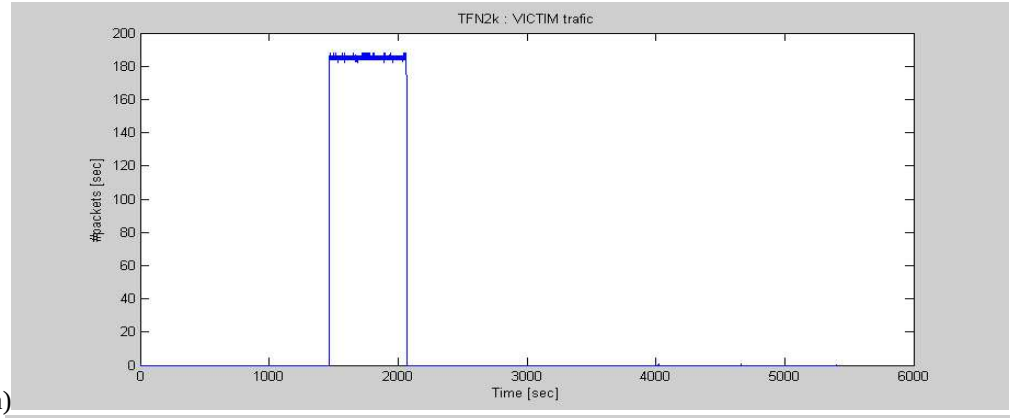
de mettre en place des bots TRINOO et TFN2k sur plusieurs sites : Nice, Mont de Marsan et Lyon. Nous avons volontairement bridé ces bots, pour que les puissances des attaques soient plus faibles que lors des campagnes précédentes. L'objectif final est de cacher au maximum ces attaques dans le trafic légitime, afin d'en rendre la détection plus difficile encore.

Logiciel	Type attaque	Durées traces			Durées attaques			Intensités		
Campagne de Mars 2006										
Trinoo	UDP Flooding	2h	1h	1h	10mn	10mn	10mn	7%	22,90%	86,80%

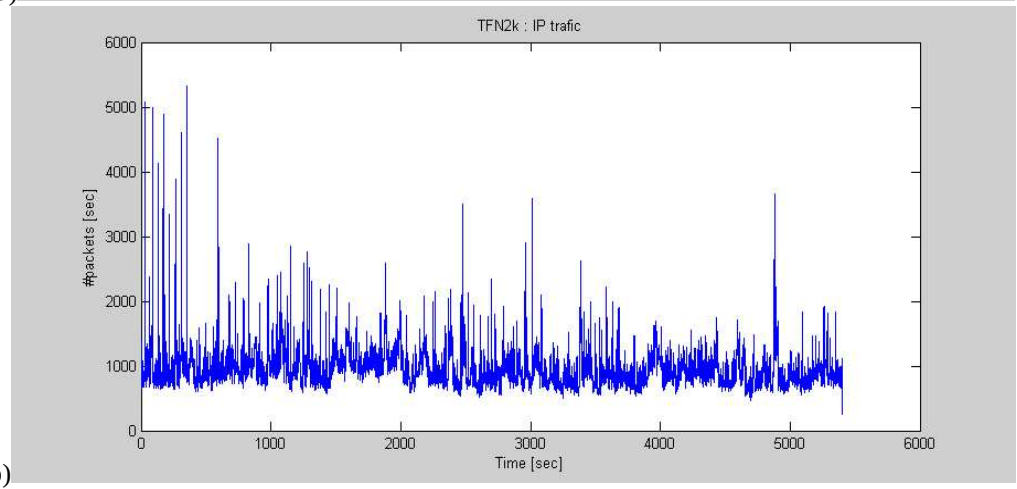
Tableau 4- *Attaques TRINOO*

Logiciel	Type attaque	Durées traces			Durées attaques			Intensités		
Campagne d'Avril à Juillet 2006										
TFN2k	UDP Flooding	2h	1h	30mn	11mn	10mn	10mn	92%	4,00%	7%
	ICMP Flooding	1h30	1h		20mn	10mn		13%	9,80%	
	TCP SYN Flooding	2h	1h		10mn	10mn		12%	33%	
	Mixed Floofing	1h			10mn			27%		
	Smurf	1h			10mn			4%		

Tableau 5- *Attaques TFN2k*



(a)



(b)

Figure 5 – Trace avec attaque TFN2k. (a) Trafic arrivant sur la cible (b) Trafic total du LAAS-CNRS

L'exemple proposé est une trace contenant une anomalie de type ICMP flooding. Elle a été réalisée à l'aide de trois bots TFN2k, situés à Mont de Marsan, Nice et Lyon, et tous trois contrôlés par un master situé à Mont de Marsan. La puissance de l'attaque est de 13%. La liste des expérimentations effectuées est donnée dans les tableaux 4 et 5.

La figure 5(a) montre le trafic arrivant sur la machine cible, et provenant d'un des bots TFN2k. On voit apparaître les paquets envoyés par les trois zombies à destination de la victime. La figure 5(c) montre le trafic total arrivant sur le réseau du LAAS-CNRS. On n'observe pas d'anomalie flagrante : le trafic d'attaque est « caché » dans le trafic global.

Campagne TFN2k avec bursts (novembre – décembre 2006)

La dernière campagne effectuée a utilisé une version modifiée du zombie TFN2k afin de supprimer le comportement continu du zombie. Les objectifs étaient similaires aux campagnes TFN2k et TRINOO, à savoir réaliser des attaques de faibles intensités, masquées dans le trafic global. Mais le profil des attaques était différent : chaque zombie envoie des salves (bursts) de paquets, et non plus des paquets en flots continus. Ce type d'attaque a été récemment mis à jour dans plusieurs travaux, comme par exemple dans [29].

Campagne de novembre-décembre 2006														
		Durée trace	Pic n°1	Pic n°2	Pic n°3	Pic n°4	Pic n°1min	Pic n°2 min	Pic n°3 min	Pic n°4 min	Pic n°1max	Pic n°2 max	Pic n°3 max	Pic n°4max
TFN2k	Flat Burst	2h	5mn	6mn	11mn	12mn	0,83%	0,74%	0,80%	1,02%	0,83%	1,54%	1,53%	2,36%
	Ramp Burst	2h	5mn	5mn	9mn	9mn	4,62%	5,54%	0,89%	1,59%	4,62%	5,54%	2,92%	3,66%
	Variable Burst	2h	5mn	8mn	5mn	6mn	3,28%	1,03%	1,03%	1,70%	3,28%	4,93%	5,60%	6,57%
	Asynchon Burst	2h	7mn	9mn	9mn	10mn	0,42%	0,65%	0,99%	1,31%	1,21%	2,39%	3,46%	3,48%

Tableau 6- *Description des attaques contenues dans la base*

Les salves d'attaques que nous avons générées (moyennant modification du code du bot TFN2k) peuvent prendre plusieurs formes : « flats burst », « ramp burst », « variable burst », « asynchron burst ». Les flats bursts sont équivalentes à l'enchaînement de plusieurs flooding classiques (forme carrée). Les « ramp bursts » permettent de progressivement faire croître la puissance de chaque salve (forme dent de scie). Les « variable bursts » permettent d'obtenir des flats bursts, avec une intensité très variable (forme carrée, avec beaucoup de « bruit »). Enfin, les « asynchron bursts » sont des flats bursts, générés par des bots TFN2k non synchronisés.

Nous illustrons ici les « ramp bursts ». L'attaque est constituée de 4 salves croissantes exécutées par 3 bots TFN2k situés à Mont de Marsan. Chacun des bots TFN2k a floodé la cible en utilisant un protocole différent (TCP SYN, UDP, ICMP) afin de réaliser une attaque de type MIX FLOODING.

Sur la figure 6(a), on peut observer le comportement du bot effectuant de l'ICMP flooding. On voit effectivement le comportement discontinu de l'attaque et l'évolution de la puissance au court de chaque salve. Sur la figure 6(b), on observe le trafic de la machine ciblée du LAAS-CNRS. On peut remarquer que les puissances et les formes de chaque salve sont différentes. En effet, les trois zombies attaquants calculent une durée variable d'attente entre chaque salve (entre 5 et 10 minutes). La conséquence est que l'agrégation des trois flots sur la cible n'est pas forcément synchrone : sur les deux dernière salves, on voit parfaitement apparaître les paquets issus des trois bots (trois rampes différentes dans la même salve). Notons que ce mécanisme a été mis en place pour éviter de faire apparaître une périodicité des salves dans les mesures statistiques du trafic, et donc compliquer leur détection. La figure 6(c) montre le trafic global du LAAS-CNRS. On n'y distingue aucune salve, conformément à notre objectif initial.

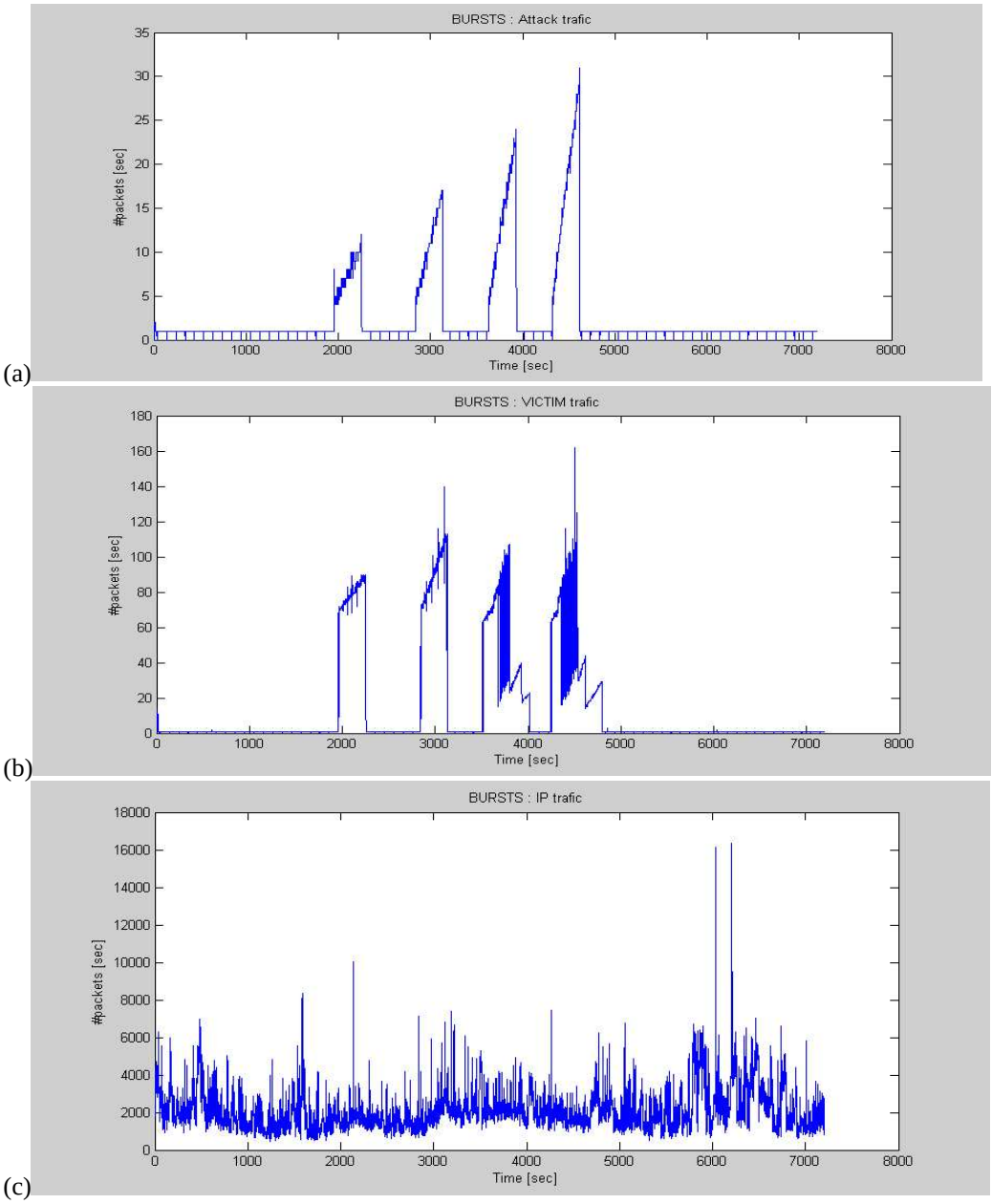


Figure 6 – Trafic avec attaque TFN2k Burst. (a) Trafic du bot de Mont de Marsan, (b) Trafic arrivant sur la cible et (c) Trafic total du LAAS

5. Conclusion et perspective

Dans cet article nous présentons la base d'anomalies qui a été réalisée dans le cadre du projet METROSEC de l'ACI Sécurité et Informatique [7]. Les traces contenues dans cette base balaient un éventail des attaques en déni de service réseau que l'ont peut retrouver dans un réseau de grande envergure. Cette base contient des exemples d'anomalies provoquées par des logiciels de flooding, mais aussi des anomalies provoquées par des actions coordonnées de personnes légitimes (foules

subites). Enfin, des attaques en flooding discontinues ont été réalisées. Cette base de traces peut servir de benchmark aux différentes méthodes et outils de détection.

A ce jour, cette base n'est pas accessible en ligne. Très prochainement, des parties seront rendues publiques, avec la documentation permettant d'identifier les anomalies contenues dans les traces.

Trois perspectives principales sont envisageables :

- Compléter la base par des traces contenant d'autres types d'anomalies légitimes ou illégitimes, comme par exemple des traces de vers ou virus
- Utiliser de nouvelles métriques sur les différentes traces. A ce jour, seuls le nombre de paquet par seconde, et le nombre d'octets par seconde ont été utilisés. Il serait intéressant de regarder d'autre métriques, comme le nombre de connexions TCP par seconde, ou le nombre de flux par seconde, ou encore le temps d'inter arrivée entre les paquets. Ces résultats permettraient de documenter encore mieux la base de traces.
- Utiliser notre base pour calibrer et tester des NIDS, et plus particulièrement les procédures de détection développées dans le cadre du projet METROSEC.

Remerciements

Les auteurs remercient L. Bernaille (LIP6, Paris 6), A.Scherrer (CITI, INSA de Lyon), le CRI de l'ENS Lyon, le CIUPPA de PAU, pour leur aide dans la collecte de données de trafic et dans la conduite des expérimentations d'attaques. Ils remercient aussi tous les collègues qui ont gracieusement accepté de prendre part aux expérimentations de foules subites. Ce travail a été rendu possible grâce au support financier du MNRT dans le cadre du programme ACI *Sécurité et Informatique* 2004 (projet METROSEC).

Références

[1] *Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures*, Stephen M. Specht, Ruby B. Lee, ISCA PDCS 2004, pp.543-550, September 2004

[2] *A Taxonomy of DDoS Attacks and Defense Mechanisms*, J. Mirkovic, P. Reiher, ACM SIGCOMM Computer Communications Review, Volume 34, Number 2, pp. 39-54, April 2004.

[3] *Design principles for accurate passive measurement*, J.Cleary, S.Donnelly, I.Graham, A.McGregor, M.Pearson, In Passive and Active Measurements (PAM2000), Hamilton, New Zealand, April 2000.

[4] *Laasnetexp.fr*, http://www.laas.fr/~owe/laasnetexp.fr/laasnetexp.fr.htm

[5] *Analyse spectrale d'outils classiques de DDoS*, L.Gallon, J.Aussibal, Colloque sur les Risques et la Sécurité d'Internet et des Systèmes (CRISIS05), Bourges, France, October 2005.

[6] *Internet traffic characterisation – an analysis of traffic oscillations*, P.Owerzarski, N.Larrieu, 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'04), Toulouse, France, July 2004

[7] *METROSEC project*, http://www.laas.fr/METROSEC

[8] *QoSMOS Traffic Designer*, http://www.qosmos.net

[9] *Détection d'attaques de déni de service par un modèle non gaussien multirésolution*, P. Borgnat, N. Larrieu, P. Owezarski, P. Abry, J. Aussibal, L. Gallon, G. Dewaele, K. Boudaoud, L. Bernaille, A. Scherrer, Y. Zhang, Y. Labit, Colloque Francophone d'Ingénierie des Protocoles (CFIP'2006), Tozeur, Tunisie, November 2006.

[10] *Détection d'attaques de ``Déni de Service" : ruptures dans les statistiques du trafic*, P. Borgnat, N. Larrieu, P. Abry, P. Owezarski, Colloque GRETSI-05, Louvain-la-Neuve, Belgique, September 2005.

[11] *IPERF, The TCP/UDP bandwith Measurement Tool* http://dast.nlanr.net/Projects/Iperf/

[12] *TFN2k, An analysis*, http://packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt

[13] *HPING2*, http://sourceforge.net/projects/hping2

[14] *The DoS Project's "Trinoo" distributed denial of service attack tool* http://staff.washington.edu/dittrich/misc/tinoo.analysis

[15] *Somme issues raised by Dos attacks and the TCP/IP suite*, S.Farraposo, K.Boudaoud, L.Gallon, P.Owezarski, SAR'2005, Batz-sur-mer, France, June 2005.

[16] *A framework for classifying denial of service attacks*, A.Hussain, J.Heidemann, C.Papadopoulos, In SIGCOMM, Karlsruhe, Germany, August 2003

[17] *Inferring Internet Denial-of-Service activity*, D.Moore, C.Shannon, D.Brown, G.Voelker, S.Savage, ACM Transactions on Computer Systems (TOCS), 2006

[18] *On the relationship between file sizes, transport protocols, and self-similar network traffic*, K. Park, G. Kim, M. Crovella, International Conference on Network Protocols, IEEE Computer Society, page 171, Washington DC, USA, October 1996.

[19] *A Markov chain model of temporal behavior for anomaly detection*, N. Ye, Workshop on Information Assurance and Security, West Point, NY, June 2000

[20] *Diagnosing Network-Wide Traffic Anomalies*, A. Lakhina, M. Crovella, C. Diot, SIGCOMM, Portland, USA, August 2004

[21] *A signal analysis of network traffic anomalies*, P. Barford, J. Kline, D. Plonka, A. Ron, ACM/SIGCOMM Internet Measurement Workshop, Marseille, France, November 2002.

[22] *Flash Crowds and Denial of Service Attacks : Characterization and Implications for CDNs and Web Sites*, J. Jung, B. Krishnamurthy, M. Rabinovitch, International WWW Conference, Honolulu, HI, May 2002.

[23] *Results of the DARPA 1998 Offline Intrusion Detection Evaluation*, Richard P.Lippmann, Robert K.Cunningham, David J.Fried, Issac Graf, Kris R.Kendall, Seth E. Webster, Marc A. Zissman, slides presented at RAID 1999 Conférence, West Lafayette, Indiana, September, 1999.

[24] *Traffic Classification using Clustering Algorithms*, J.Erman, M.Arlitt, A.Mahanti, SIGCOMM'06 MineNet Workshop, Pisa, Italy, September 2006

[25] *NLANR project, Auckland IV*, http://pma.nlanr.net/Traces/long/auck4.html

[26] *DARPA Intrusion Detection Data sets*, http://www.ll.mit.edu/IST/ideval/data/data_index.html

[27] *CAIDA Project*, http://www.caida.org/home/

[28] *LBNL repository*,http://ita.ee.lbl.gov/index.html

[29] *Low-Rate TCP-Targeted Denial of Service Attacks—The Shrew vs. the Mice and Elephants*, A. Kuzmanovic, E. W. Knightly, SIGCOMM 2003, Karlsruhe, Germany, Aug.2003.

[30] *Collaborative Detection and Filtering of Shrew DDoS Attacks using Spectral Analysis*, Yu Chen, Kai Hwang, Journal of Parallel and Distributed Computing, Special Issue on Security in Grids and Distributed Systems, 2006

[31] *Prefixpreserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme*, J.Xu, J.Fan, Mostafa H. Ammar, Sue B. Moon, 10th IEEE International Conference on Network Protocols, November 2002.

[32] *Testing Intrusion Dectection Systems : a critique of the 1998 and 1999 DARPA Intrusion Detection System evaluations as performed by Lincoln Laboratory*, John McHUGH, ACM Transactions on Information and System Security, vol.3, n° 4, November 2000, pages 262-294

[33] *Inferring Internet Denial-Of-Service Activity*, David Moore, 10th Usenix Symposium, Washington, DC, August 2001

[34] *Knowledge Discovery in Databases Archive (KDD99)*, http://kdd.ics.uci.edu/