LES DOSSIERS TECHNIQUES

# Bots et Botnets

septembre 2009


Espace Menaces

**CLUB DE LA SECURITE DE L'INFORMATION FRANÇAIS**

30, rue Pierre Sémard, 75009 PARIS

Tél. : +33 1 53 25 08 80 – Fax : +33 1 53 25 08 88 – e-mail : clusif@clusif.asso.fr

Web : http://www.clusif.asso.fr

# REMERCIEMENTS

Le CLUSIF tient à mettre ici à l'honneur les personnes qui ont rendu possible la réalisation de ce document, tout particulièrement :

| | |
|---|---|
| **BERTIN Michel** | |
| **BIZEUL David** | *Société Générale* |
| **CONSTANT Paul** | *Consultant* |
| **LE CHEVALIER Rémy** | *LCS Conseil* |
| **MAKOWSKI Stéphane** | *Ministère de la Défense* |
| **PAGET François** | *McAfee* |
| **SAULIERE Pascal** | *Microsoft* |
| **SIMON Jean-Charles** | *Michelin* |
| **VEYSSET Franck** | *France Télécom* |

Nous remercions aussi les membres ayant participé à la relecture.

# SOMMAIRE

# I - Périmètre de l'étude

En mai 1999, un nouveau ver ciblant les environnements 32bits de Windows voyait le jour. Sous le nom de W32/Pretty.worm (alias PrettyPark), il ne fit pas particulièrement parler de lui. Il reprenait une idée développée en décembre 1993 (avec EggDrop – programme non malveillant) puis en avril 1998 (avec GTbot – programme malveillant) en envoyant à son auteur des données issues de l'ordinateur infecté et en attendant les ordres au travers d'un canal IRC (Internet Relay Chat) spécifique.

Ce nouveau venu fut vite oublié et ce n'est qu'à partir de 2002 que le terme « robot » apparu dans les médias grand public pour définir des programmes pouvant former un ensemble de machines esclaves, organisées en réseau et contrôlées à distance par un individu à priori malveillant.

Aujourd'hui, les botnets sont symptomatiques de l'évolution vers la criminalité informatique organisée. Ils permettent à leurs auteurs de réaliser des attaques qui leur sont bénéfiques d'un point de vue économique (vol d'identité, émission de spam, chantage) et politique (activisme). Pour mieux rentabiliser leurs affaires, les possesseurs de botnets peuvent aussi louer leurs services et leurs réseaux à d'autres individus malfaisants.

En dépit de quelques condamnations spectaculaires [1], les botnets ne sont pas sur le déclin. La combinaison du développement des méthodes de camouflage face aux antivirus et IDS/IPS et du nombre de machines non correctement administrées, connectées en permanence au haut débit, amène le CERT-Renater à annoncer que ce problème restera encore longtemps une préoccupation pour les personnes s'intéressant à la sécurité réseau [2].

Ce document s'adresse aux responsables sécurité, aux directeurs des systèmes d'information et à leurs équipes. Il est aussi destiné à tous ceux qui ont des responsabilités dans le domaine de la sécurité informatique. Il a pour but de présenter d'un point de vue pédagogique et technique les différents aspects liés à cette menace.

Après les diverses définitions d'usage, nous présenterons les grandes familles de robots et les motivations de ceux qui les exploitent. Les principaux aspects techniques seront développés dans un paragraphe s'intéressant au fonctionnement et à la propagation de ces programmes. Le document abordera ensuite les problématiques de détection, d'interception et de prévention.

[1] Operation Bot Roast: http://www.fbi.gov/pressrel/pressrel07/botnet061307.htm
[2] Les botnets : http://www.sg.cnrs.fr/fsd/securite-systemes/revues-pdf/num61.pdf

# II - Définition

Les robots sont des composants malveillants et autonomes installés sur des machines compromises dont l'ensemble constitue un *botnet* (roBOT NETwork) à même de :

- servir de relais de spamming et/ou de phishing ;
- identifier et infecter d'autres machines ;
- participer à des attaques groupées (DDoS) ;
- être utilisé dans la diffusion de programmes indésirables ;
- générer de façon abusive des clics sur un lien publicitaire au sein d'une page web (fraude au clic) ;
- capturer de l'information sur les machines compromises ;
- effectuer des opérations de calcul distribué ;
- servir à mener des opérations de commerce illicite par gestion de proxy d'accès à des sites de ventes de produits interdits ou de contrefaçons.

En 2004 et 2005, les robots étaient présentés comme des programmes à part des virus et des chevaux de Troie. Leurs noms se terminaient en « BOT » (W32/sdbot, W32/spybot, W32/gaobot…). Basées sur le nombre de variantes rencontrées dans les bases de référence des outils de détection, les statistiques montrent une croissance dépassant les 300% (estimation globale – toutes familles confondues - de l'augmentation du nombre de malware entre 2007 et 2008).

On constate depuis lors que de nombreux programmes malveillants ont une fonctionnalité de type robot. Qu'il s'agisse de vers, de virus ou de chevaux de Troie, ils sont conçus pour recevoir des ordres et les exécuter à divers moments de leur existence.

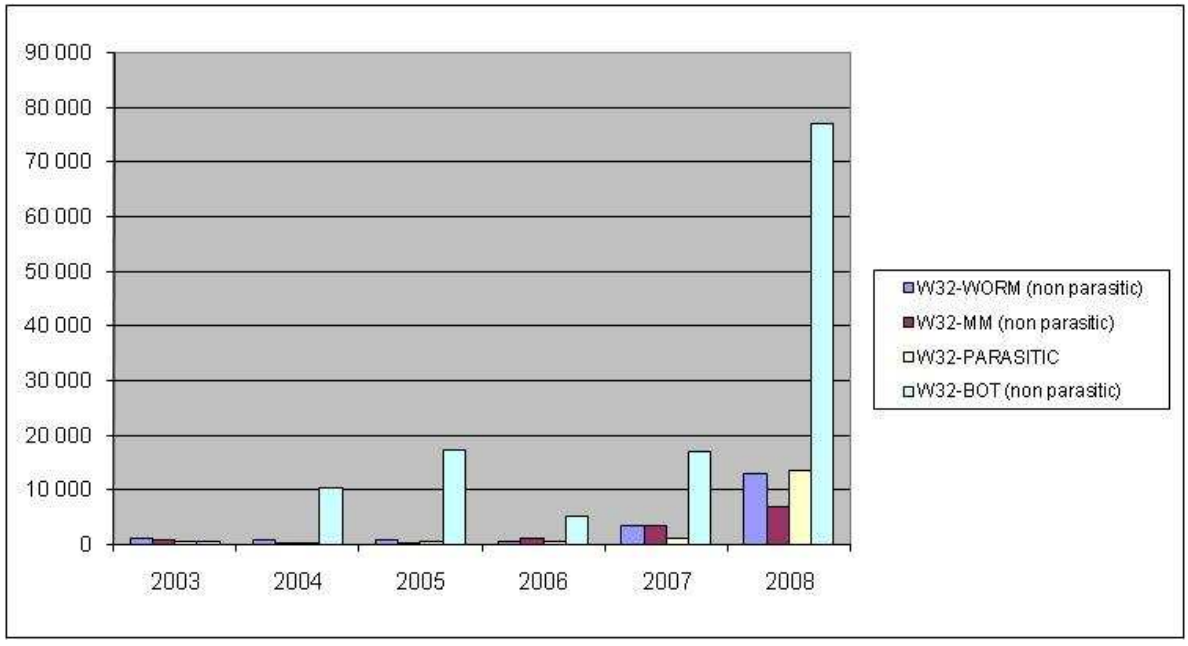


**Figure 1 : Evolution non cumulative du nombre de programmes de type robot, comparée à celle des vers, mass-mailers et virus (comptabilisation faite aux niveaux des variantes principales)**

Pour mesurer toute l'ampleur du phénomène, il faut donc rajouter à ces précédentes variantes facilement identifiables par leurs noms se terminant en « BOT », d'autres programmes malveillants connus aujourd'hui sous des noms que rien ne distingue (W32/Nuwar, W32/Mytob, Spam-Samburg, Srizbi, Backdoor-DIX, etc.).

Le niveau du trafic généré ou induit permet aussi de mesurer l'étendue du problème. On comptait hier le nombre de robots dans les bases de références des outils de détection, aujourd'hui on se doit aussi de compter, à un instant donné, le nombre de robots en activité.

Diverses organisations proposent ce type de graphiques sur Internet. Shadowserver est l'une d'entre elles ; elle offre sur son site toute une série de courbes [3] tenant compte de la date de dernière activité suspecte constatée. Si, sur un certain nombre de jours (30, 10 ou 5), rien de suspect n'est détecté sur une machine (identifiée par son adresse IP), cette dernière est alors retirée du total. A titre d'exemple, la courbe suivante prend en compte, sur une période d'un an, les activités remontant à moins de 10 jours.

Début 2009, la tendance était clairement à la hausse avec une augmentation de 300 000 à plus de 700 000 machines (dans le cas d'activités remontant à moins de 30 jours) ou de 60 000 à 140 000 (dans le cas d'activités remontant à moins de 5 jours).

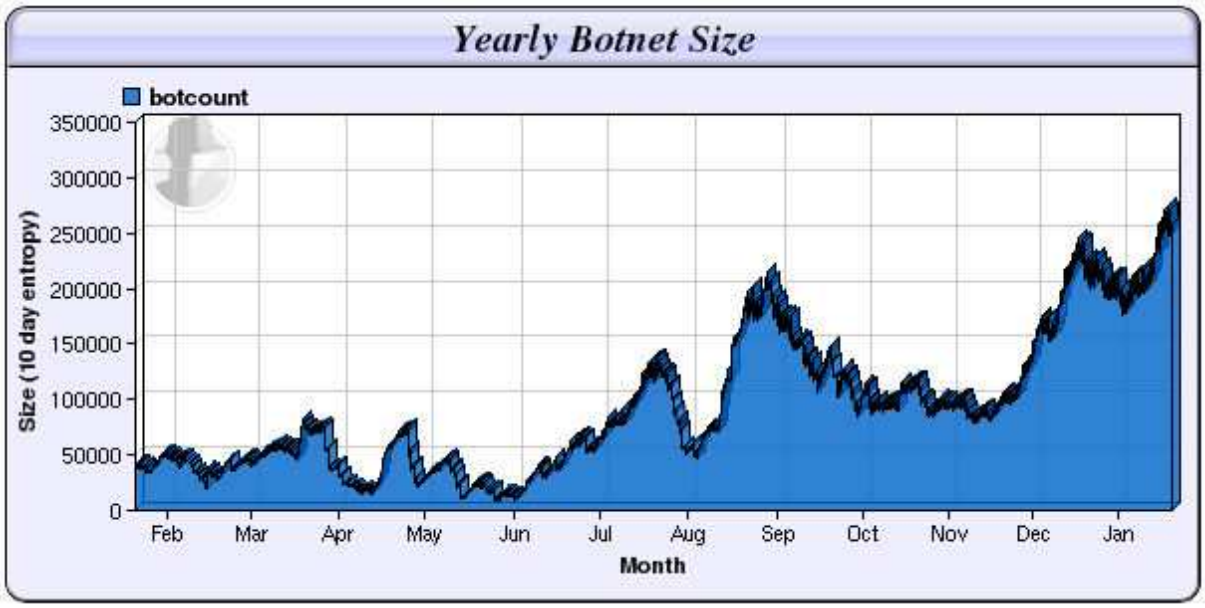


**Figure 2 : Estimation du nombre de machines zombies (source ShadowServer – activité remontant à moins de 10 jours – période de février 2008 à janvier 2009)**

Le site comptabilise aussi une estimation du nombre de postes de commande et de contrôle (C&C). Il avoisine les 2 800 unités depuis août 2008.

[3] http://www.shadowserver.org/wiki/pmwiki.php/Stats/BotCounts

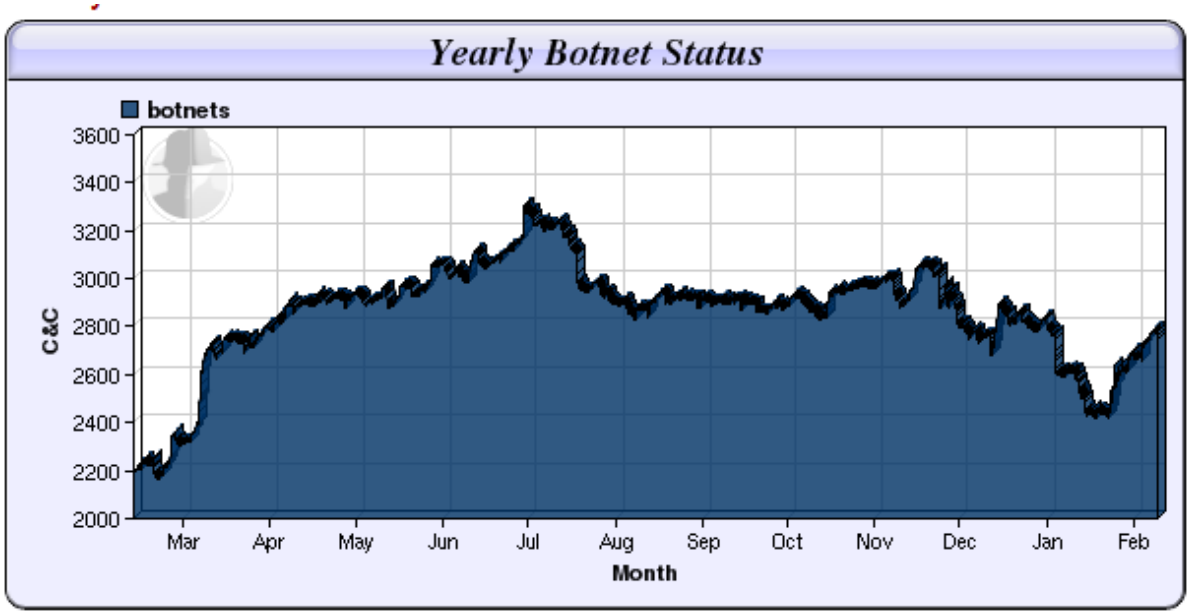


**Figure 3 : Estimation du nombre de machines pilotes (source ShadowServer – période de mars 2008 à février 2009)**

Ces chiffres sont sans aucun doute minimalistes car les botnets gagnent en furtivité. Les réseaux sont plus petits que par le passé et ne se limitent plus à l'utilisation de canaux de communication de type IRC, ou plus récemment du flux HTTP. Certains se basent maintenant sur le modèle Peer-To-Peer (P2P) afin d'assurer une résilience du réseau.

## II.1 - Taille

La taille des botnets est très variable, allant de quelques dizaines de machines zombies à plusieurs dizaines de milliers. La moyenne est sans doute de l'ordre du millier d'éléments.

On notera cependant que des chiffres bien plus impressionnants circulent. En 2005, la police néerlandaise annonçait avoir arrêté un trio de jeunes pirates accusés d'avoir compromis 1.5 million de machines [4]. De son côté, à Davos, en janvier 2007, Vinton Cerf, l'un des fondateurs de l'Internet, n'hésitait pas à affirmer qu'une machine connectée à Internet sur quatre ferait partie d'un botnet [5].

Plus récemment, Joe Stewart, directeur de la recherche sur les malware chez SecureWorks, insistait sur l'importance des botnets dans la diffusion du spam. Il présentait à la conférence RSA (avril 2008) les chiffres suivants [6] :

- Srizbi - 315 000 machines - 60 milliards de spam/jour.
- Bobax - 185 000 machines - 9 milliards de spam/jour.
- Rustock - 150 000 machines - 30 milliards de spam/jour.
- Cutwail - 125 000 machines - 16 milliards de spam/jour.
- Storm - 85 000 machines - 3 milliards de spam/jour.

---

[4] Botnet operation controlled 1.5m PCs: http://www.vnunet.com/vnunet/news/2144375/botnet-operation-ruled-million

[5] How Many Bot-Infected Machines on the Internet? : http://www.avertlabs.com/research/blog/index.php/2007/01/29/how-many-bot-infected-machines-are-on-the-internet/

[6] Un million de PC robots pour 100 milliards de spam : http://www.neteco.com/135320-pc-robots-100-spam.html

- Grum - 50 000 machines - 2 milliards de spam/jour.
- Onewordsub - 40 000 machines.
- Ozdok - 35 000 machines - 10 milliards de spam/jour.
- Nucrypt - 20 000 machines - 5 milliards de spam/jour.
- Wopla - 20 000 machines - 600 millions de spam/jour.
- Spamthru - 12 000 machines - 350 millions de spam/jour.

En janvier 2009, MessageLabs annonce des chiffres tout aussi impressionnants :

| Rank (average spam per day) | Botnet | Estimated botnet size: at least X active Ips in last 30d | average spam per day |
|---|---|---|---|
| 1 | Mega-D (Ozdok) | 660,000 | 38,225,669,306 |
| 2 | Cutwail (Pandex) | 1,080,000 | 7,741,703,816 |
| 3 | Rustock (Rustock) | 410,000 | 6,219,110,041 |
| 4 | Xarvester | 260,000 | 4,438,707,255 |
| 5 | DonBot | 800,000 | 4,015,511,013 |
| 6 | Gheg | 140,000 | 2,736,881,174 |
| 7 | Grum (Grum) | 100,000 | 888,549,737 |
| 8 | Bagle (Beagle) | 150,000 | 505,413,807 |
| 9 | Unknown New (TBC) | 20,000 | 163,011,924 |
| 10 | Warezov/Stration | 10,000 | 131,401,720 |

**Figure 4 : Estimation de taille des botnets en Janvier 2009 (source MessageLabs)**

## II.2 - *Les grandes familles de robots*

La qualification des familles s'appuie sur leur mode de communication [7] :

### II.2.1 - Protocole IRC (Internet Relay Chat).

C'est l'architecture traditionnelle des botnets. Dans ce modèle, un ou plusieurs serveurs ou réseaux IRC sont utilisés comme canal de commande et contrôle (C&C). Après infection, les machines clientes du botnet, les bots ou zombies, se connectent au serveur IRC sur un canal privé afin de signaler leur existence, d'obtenir des mises à jour ou des instructions. L'approche est très pratique du point de vue du botnet, car il suffit à son propriétaire de se connecter au même salon de discussion (*channel*) que les machines compromises pour leur donner ses instructions. Mais c'est aussi très simple à repérer : le trafic IRC, légitime ou malveillant, s'identifie parfaitement sur un réseau, au milieu des protocoles plus communs tels que HTTP. Une fois repéré, le fait de neutraliser le salon de discussion tue le botnet. Si une machine se connecte sur un canal sans les habilitations nécessaires, certains botnets déclenchent des actions de protection. Cette mesure permet au réseau de s'auto-protéger des chercheurs en cybercriminalité.

[7] Ce sous-paragraphe est une libre reprise d'un article de SecurityVibes : http://www.securityvibes.com/botnets-web-20-jsaiz-news-2001032.html

### II.2.2 - P2P

En s'appuyant sur une architecture décentralisée similaire à celle des réseaux de partage de fichiers, les bots ne dépendent plus d'une tête centrale. C'est le cas pour Slapper**,** Sinit, Phatbot et Storm (alias Nugache) qui utilisent les réseaux tels Gnutella pour communiquer. Mais là aussi, il est théoriquement possible de décapiter ces réseaux. Storm, par exemple, s'appuie sur une "liste" de démarrage de 22 serveurs qu'il est possible de blacklister. Cependant, cette liste étant dynamique (mise à jour automatique), la tâche s'avère particulièrement difficile. Le trafic généré reste important et peut être repéré par analyse statistique. Bien que chaque bot fasse aussi office de centre de contrôle pour ses pairs, le botnet dépend tout de même d'un enregistrement DNS, qui représente le point faible : en supprimant la résolution DNS, le botnet devient inutilisable.

### II.2.3 - HTTP

Beaucoup plus difficiles à détecter, des malware comme Black Energy ou Mocbot, non seulement se fondent dans le trafic HTTP traditionnel, mais surtout ne dépendent pas d'une connexion permanente avec leur centre de commande. Black Energy émet une simple requête POST, par laquelle il reçoit en retour un ordre encodé. Après exécution, il reviendra chercher de nouvelles instructions. Entre temps, il n'aura aucun contact avec son centre de commande. Une telle approche permet aux botnets HTTP une furtivité accrue à laquelle ne peuvent prétendre les versions basées sur IRC ou sur les protocoles P2P, qui exigent une connexion permanente et sont donc plus simples à identifier. Autre avantage du protocole HTTP, le serveur de contrôle peut être n'importe quel serveur Unix / Linux / BSD compromis pour l'occasion, et le pirate peut donc en changer très rapidement.

### II.2.4 - Web 2.0 / Ajax

Cette nouvelle technique utilise une recherche anodine sur le web pour identifier le centre de commande (utilisation d'un mot clé), puis exploite des messages Ajax pour communiquer (en passant par le port 80 HTTP). La dissimulation est ici optimale. Gozi semble être l'un des premiers botnet à utiliser cette technologie [[8]].

[8] Présentation SecureWorks (page 6): http://www.wsta.org/content/download/8003/102669/file/SecureWorks.pdf

# III - Motivations & Moyens

Si, il y a quelques années, la motivation des auteurs de malware était, en majorité, le plaisir ou la recherche de la « gloire », elle est aujourd'hui, avant tout, pécuniaire (appât du gain) et parfois idéologique (activisme/hacktivisme). Ceci est particulièrement vrai du côté des programmes robots dont le principal intérêt est l'effet de masse obtenu par la démultiplication des machines contrôlées par un seul pirate et agissant dans un même but. Les principales fonctionnalités des robots sont les suivantes :

- l'inondation : attaques en DDoS, envoi de spam, diffusion d'adware, propagation de malware ;
- la collecte d'informations et le vol de données sensibles ;
- le camouflage : diverses machines constituant le botnet et servant à masquer l'adresse du site malveillant vis-à-vis de l'utilisateur final.

## *III.1 - Motivations pécuniaires*

Les botnets offrent de nombreuses possibilités d'enrichissement.

### III.1.1 - Chantage

Capables de mener des attaques en déni de service distribué, les réseaux de machines zombies peuvent être utilisés comme outil d'intimidation auprès d'une société à qui le pirate demandera le versement d'une somme plus ou moins importante, pour éviter que son site ne soit soumis à une attaque de plus grande envergure.

En 2003, un trio de pirates basés en Russie a demandé le versement d'une somme de 10 000$ à une société australo anglaise de pari en ligne (CANBET). Celle-ci, ayant refusé de céder au chantage, fut soumise à une attaque DDoS qui lui fit perdre plusieurs centaines de milliers de dollars.

Plus récemment, aux Pays-Bas, une source officielle annonça que des propriétaires de botnet contrôlaient plus de 1,5 million d'ordinateurs. Un individu de 20 ans, arrêté en Californie, était à la tête d'un réseau formé de 400 000 ordinateurs zombies. Fin août 2005, l'organisateur allemand de loteries jaxx.de, victime d'actes de chantage répétés, a promis une prime de 40 000 euros en échange d'informations permettant d'identifier leurs auteurs.

Les attaques par déni de service distribué (DDoS) coûtent cher aux entreprises visées. A ce jour Protx (prestataire britannique de transactions en ligne), confronté à plusieurs attaques, a déboursé près de 500 000 US$. Il est vrai que les motifs sont parfois strictement politiques et non financiers. Au deuxième semestre 2005, des sites allemands de protection des consommateurs ont été victimes d'attaques par Internet [9].

### III.1.2 - Location ou vente

Certains propriétaires de botnet («botherder», « botmaster ») louent ou vendent tout ou partie de leurs réseaux. Ils peuvent aussi offrir contre rémunération tel ou tel service associé (DDoS,

[9] Melani - Rapport semestriel 2005/2:
http://www.melani.admin.ch/dokumentation/00123/00124/00161/index.html?lang=fr

diffusion de spam, etc). Cette tendance : le MaaS (Malware as a Service) intéresse les fraudeurs qui n'ont pas la connaissance technique suffisante pour créer de toute pièce leur propre réseau. A titre d'exemple, on a noté en 2008 les tarifs pratiqués suivant [10] :

- **15$ :** infection de 1 000 systèmes ;
- **25 à 100$ :** mise en place d'une attaque en déni de service (DDoS), puis 20$ l'heure et 100$ la journée. Les 10 premières minutes sont offertes ;
- **495$ :** 20 millions de spams sur une période de 14 jours.

Le faible coût de ces attaques les rend facilement abordables pour envisager des opérations de déstabilisation vis-à-vis de PME ou de grandes entreprises.

Au Pays-Bas, en août 2008, un individu de 19 ans a été arrêté par la police. Il était à la tête du botnet "Shadow" qui regroupait plus de 100 000 machines. Le jeune homme allait vendre son réseau pour 25 000 € à un brésilien de 35 ans, lui aussi interpellé.

### III.1.3 - Vol et revente d'informations

Il y a quelques années, la recherche de clés d'activation de logiciel était une fonctionnalité courante. Aujourd'hui, les informations le plus fréquemment volées sont les ensembles « identifiant - mot de passe » et les numéros de carte de crédit avec codes CVV2.

L'agencement en réseau avec poste de pilotage et système de centralisation des données récoltées s'avère être une structure idéale pour des opérations de vol de données personnelles et confidentielles. Une étude de juillet 2007 [11] démontre aussi qu'il est envisageable d'utiliser un botnet dans un but de récupération d'informations ciblées (espionnage à des fins industrielles, politiques ou terroristes).

En février 2008, la Sûreté du Québec a procédé à l'arrestation de 17 pirates informatiques suspectés d'avoir infecté plus d'un million d'ordinateurs à des fins de vol de données. Plus de 100 pays semblent avoir été touchés et les dommages causés sont estimés à plus de 45 millions de dollars.

### III.1.4 - Spam

L'émission de spam passe aussi par les botnets. On estime ainsi que 80% à 90% du spam mondial est diffusé grâce à la création et l'utilisation de réseaux de machines zombies [12]. Si le spammeur ne peut développer son propre système, il sous-traitera les envois auprès de sociétés qui n'hésitent pas à s'afficher sur le Net. Le prix est fonction du nombre de messages émis sur une période donnée.

Robert Alan Soloway, l'un des principaux acteurs de ce marché a été arrêté en mai 2007. Il est accusé de 35 chefs d'accusation dont le vol d'identité et le blanchiment d'argent. Un expert estime que, lors de son arrestation, Soloway envoyait plusieurs milliards de messages électroniques par jour. En juillet 2008, il a été condamné à 47 mois de prison.

---

[10] http://www.globalsecuritymag.fr/G-Data-revele-les-tarifs-de-la,20080630,3824
[11] Black Market Botnets : http://pages.cpsc.ucalgary.ca/~aycock/papers/bmbotnet.pdf

[12] Zombies et Botnets: http://assiste.com.free.fr/p/abc/a/zombies_et_botnets.html

### III.1.5 - Manipulation

De façon anecdotique des programmes exécutés à distance ont permis la manipulation de cours de bourse [13]. La crainte est que, en montant des réseaux de plusieurs milliers de machines zombies dans de larges botnets, les pirates influencent largement les cours de bourse visant de grandes sociétés à travers le monde.

Une affaire a mis en évidence en Estonie la prise de contrôle, par des pirates, de 25 comptes de bourse en ligne, leur permettant de réaliser plus de 350 000$ de profits. Dans le même temps, les internautes, dont les accès de bourse en ligne ont été piratés, se sont retrouvés avec des portefeuilles d'actions ne valant plus rien.

### III.1.6 - Diffusion d'adware et de malware

Des sociétés commerciales légitimes et des groupes clairement malhonnêtes rémunèrent leurs affiliés en fonction du nombre de machines pour lesquelles ils ont su convaincre leur propriétaire d'installer un logiciel gratuit ou payant. Bien qu'il soit spécifié que l'accord de l'internaute est obligatoire, il est possible, avec un botnet, d'installer, de manière silencieuse et en quantité, ce type de programmes.

A titre d'exemple, Jeanson James Ancheta [14], qui a été arrêté en novembre 2005, déclara avoir gagné 58 000$ en plaçant des publicités par l'intermédiaire de son botnet. En mai 2006, il a été condamné à 5 ans de prison.

### III.1.7 - Gestion des clics frauduleux

Tout comme ils le font avec la diffusion de programmes, certains publicitaires passent des contrats avec des responsables de sites web pour qu'ils y placent des bannières. En échange des clics générés par leur domaine, les webmasters reçoivent une rémunération. Avec un botnet, il est facile de simuler des clics depuis les diverses machines infectées.

En mai 2006, SANS Institute mettait à jour une telle fraude. Les clics publicitaires étaient automatiquement effectués par une centaine de PC zombies pilotés à distance et au détriment de Google Adsense.

Selon le dernier rapport de « Click Forensics » (octobre 2008), le nombre de clics frauduleux imputables aux botnets a augmenté de 10% au 3e trimestre 2008 et représente maintenant 27% de l'ensemble de ces fraudes.

## *III.2 - Motivations idéologiques*

Les attaques en déni de service distribué, menées par des groupes d'activistes ou de patriotes plus ou moins couverts par les gouvernements des pays depuis lesquels ils agissent, se multiplient depuis 2 ans.

En avril 2007, le déplacement, depuis le centre de la capitale de l'Estonie vers un cimetière militaire, d'un monument en hommage aux soldats russes, a été à l'origine de nombreuses

[13] La manipulation de cours de bourse dans un objectif de chantage : http://www.cases.public.lu/fr/risques/2006/panorama/4_manip_cours_bourse/4_2_manip_cours_bourse_chant/index.html

[14] Botnet farmers play the international exchange game : http://www.channelregister.co.uk/2008/03/19/botnet_spyware_scam/

attaques à l'encontre de sites web estoniens. Des attaques du même ordre ont été menées en juillet 2008 à l'occasion du conflit entre la Russie et la Géorgie [15].

Le 26 avril 2008, les autorités biélorusses furent soupçonnées d'avoir couvert une attaque visant à bloquer les sites web de la radio américaine : Radio Free Europe/Radio Liberty, émettant depuis Prague vers une vingtaine de pays [16]. Ce jour là, devait avoir lieu une commémoration en faveur des victimes de Tchernobyl (22e anniversaire de l'explosion de la centrale). Les sites furent bloqués plusieurs heures (jusqu'à 2 jours).

Les « hacktivistes » des pays de l'Est ne sont pas les seuls à mener ce type de combat. La Chine est également régulièrement montrée du doigt. A l'occasion des derniers jeux olympiques, et en réponse aux critiques apparues dans les médias occidentaux sur le parcours de la flamme olympique, et l'attitude de la Chine envers le Tibet, une préparation d'attaque contre CNN par un groupe de pirates chinois (« Revenge of the flame ») a été détectée. L'attaque n'a pas eu lieu car, la préparation ayant été découverte, elle ne pouvait plus bénéficier de l'effet de surprise.

On notera aussi que divers pays occidentaux réfléchissent aux avantages que l'utilisation de tels réseaux apporterait pour perfectionner la défense de leurs territoires [17].

## III.3 - Camouflage

Outre l'utilisation de rootkits sur la machine locale infectée, la technique de camouflage la plus utilisée porte le nom de *fast-flux*. Dans ses deux variantes principales, les techniques *fast-flux* exploitent les faiblesses des services d'enregistrement de noms de domaine et de résolution de noms. Ils utilisent pour cela un botnet dont chacune des machines compromises sert d'intermédiaire entre la machine de la victime et le site qu'elle cherche à atteindre.

La première variante, intitulée *single-flux* utilise des modifications rapides du DNS pour masquer l'emplacement des sites Web et des services Internet frauduleux. Dans la seconde variante, appelée *double-flux*, les pirates complètent le réseau qui masque les sites Web par un second réseau hébergeant des serveurs DNS. Le fonctionnement détaillé de ces architectures est décrit en Annexe, section VII.

Des casinos virtuels frauduleux et de nombreux sites de vente en ligne de contrefaçons et stimulants sexuels utilisent ce principe. En janvier 2008, la revue XMCOpartners a analysé les liens menant au site morningcan.com plus connu comme « Canadian pharmacy » [18]. Une première connexion menait à une adresse IP liée à un site hébergé à Hong Kong. Quelques minutes plus tard, l'adresse IP n'était plus la même ; le site était, cette fois, hébergé en Corée.

## III.4 - Vengeance

En février 2007, une attaque en DDoS a été menée contre le site Castlecops, organisation à but non lucratif dédiée à la lutte anti-spam, qui a aujourd'hui cessé ses activités. Les attaquants utilisèrent un botnet de 7 000 machines. L'arrêt du site a causé à la société une

---

[15] http://www.domainesinfo.fr/actualite/1653/cyber-attaques-en-georgie.php

[16] Radio Free Europe DDoS : http://asert.arbornetworks.com/2008/04/radio-free-europe-ddos/
http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyId=11&articleId=9082258&intsrc=hm_topic

[17] http://www.armedforcesjournal.com/2008/05/3375884

[18] http://www.xmcopartners.com/actu-secu/XMCO-ActuSecu-Janvier2008.pdf

importante perte financière. La motivation du pirate, un certain Greg C. King [19] était, dans ce cas, la vengeance, et celle-ci durait depuis 4 ans. Il considérait avoir touché des sommes dérisoires pour le travail qu'il avait effectué [20].

Plus près de nous, en novembre 2008, le site de Grande-Bretagne *bobbear.co.uk* subit une attaque en DDOS rendant son site indisponible. Le même mois, une 2e attaque, cette fois-ci, par usurpation d'identité, vise à ternir l'image du site, en envoyant quantité de messages à connotation pornographique. Du fait du nombre important d'adresses destinatrices erronées rattachées à ce message, le site fut, en plus, submergé de messages de non distribution. A noter qu'en octobre 2007, le site avait déjà subi ce même type d'attaque. Les analystes sont convaincus que ces attaques sont lancées par des pirates mécontents de l'activité d'information des utilisateurs effectuée par le site de Bobbear [21].

[19] http://www.usdoj.gov/criminal/cybercrime/kingSent.pdf
[20] http://www.theregister.co.uk/2007/10/04/bot_herder_profile/
[21] http://blogs.zdnet.com/security/?p=2188

# IV - Fonctionnement et propagation

Pour constituer son botnet, le pirate doit répandre son robot sur un grand nombre de machines. Il emploie les méthodes communes aux virus et aux chevaux de Troie. La messagerie électronique reste l'un des principaux vecteurs de propagation des robots. Citons ici l'exemple de Storm (alias Nuwar) qui, avec ses nombreuses cartes virtuelles et son *social engineering*, utilise très intelligemment l'actualité et le calendrier.

La propagation se fait aussi par le biais de sites web préalablement piégés (injection SQL, *PHP include*, ou *cross site scripting* - XSS).

Pour plus d'efficacité, le programme injecté peut aussi, de lui-même, rechercher d'autres victimes sur le réseau en essayant d'exploiter des vulnérabilités que son auteur lui a transmises.

## *IV.1 - Modes de contrôle*

Pour dialoguer avec celui qui les pilote (le *botmaster*), les machines constituant un botnet échangent, en simultané, des informations avec un centre de commande et de contrôle (*Command & Control*, ou *C&C*) distant. L'existence de ce canal de contrôle est le principal élément différenciant un robot d'un autre malware (cheval de Troie ou virus). Le pilotage se fait essentiellement sur deux modèles.

Le premier modèle est dit centralisé. Les communications passent au travers de canaux de communication de type IRC, ou, plus récemment, au travers de trafic HTTP ou de canaux cachés [22] pour tenter de s'affranchir des limites des firewalls. Une machine unique est le point de contact de tous les bots. L'ensemble des bots se connecte alors à ce point central et attend des instructions. Ce modèle a l'avantage d'être simple à implémenter et de présenter des temps de réponse rapides pour un grand nombre de bots. Son principal inconvénient est le rôle crucial joué par ce serveur central, rendant la survie du botnet fortement lié à cette machine. Son choix est donc vital pour l'attaquant (connexion permanente, bande passante élevée, etc.).

Le second modèle, en expansion aujourd'hui, est basé sur le système *peer-to-peer* (*P2P*) afin d'assurer une résilience du réseau. L'inconvénient de ce modèle réside dans les temps de réponse parfois élevés et dans la difficulté à supporter un grand nombre de bots (plusieurs milliers).

Les paragraphes suivants donnent quelques détails sur ces modes de contrôles.

### IV.1.1 - IRC

Le protocole IRC existe depuis 1988 [23]. Très vite, des automates, ou robots, ont été utilisés pour contrôler les canaux (*channels*), puis pour ajouter des fonctionnalités plus évoluées comme des jeux ou des serveurs de fichiers. La simplicité à automatiser l'emploi de ce

[22] Canaux Cachés:
http://laure.gonnord.org/pro/teaching/MIF30/projets2009/boukhemis_boutchicha_bendriss_rapport.pdf
[23] http://www.irc.org/history.html

protocole a permis la disponibilité de code et scripts, utilisant, par exemple, le moteur de script de mIRC [24].

IRC est depuis son essor du début des années 90, la plate-forme de communication privilégiée des individus qui développent les *bots*. Quoi de plus naturel alors pour ces développeurs que d'utiliser le protocole qu'ils maîtrisent et utilisent quotidiennement comme canal de communication entre pirates et *bots :* le premier botnet connu (GTBot, 1998) était une copie de mIRC avec quelques scripts.

Depuis que les concepteurs de botnet ont constaté que les réseaux IRC publics ne suffisaient plus pour héberger un botnet important, ils ont adapté à la charge souhaitée des serveurs IRC disponibles en Open Source pour reprogrammer leur propre serveur, voire l'agrémenter de fonctions supplémentaires comme du chiffrement.

Dans un botnet IRC, les commandes peuvent être envoyées aux bots par plusieurs voies : le *topic* (message de bienvenue) du canal, les messages au canal ou les messages privés vers les *bots*.

Les avantages d'IRC comme protocole de C&C sont donc :

- disponibilité : celle de l'infrastructure, des bases de code, des scripts, des compétences ;
- performance : il supporte un nombre de clients important, la bande passante nécessaire reste très faible ;
- simplicité de mise en œuvre : sur des infrastructures existantes ou des serveurs dédiés ;
- possibilité d'avoir des commandes persistantes (*topic*).

Inconvénients d'IRC :

- facile à détecter et tracer : la plupart des anciens bots IRC ne chiffraient pas leurs communications, ce qui en simplifiait la détection. Il suffisait ensuite, pour surveiller le réseau, de se connecter au serveur de C&C en se comportant comme un bot ;
- dépendant de l'architecture DNS ;
- en cas de serveur IRC privé, aisé à démanteler : en déconnectant le serveur de C&C central du réseau (*« Single Point of Failure »*) ;
- nécessité d'une connexion TCP permanente entre le bot et le serveur et ne passant pas par un proxy (serveur mandataire) d'entreprise.

Exemples de botnets IRC : GTBot [25], rbot, sdbot, agobot, spybot.

Exemple de commandes d'un bot IRC : voir annexe II

### IV.1.2 - HTTP

Bien qu'IRC soit toujours le mode de contrôle le plus répandu, HTTP connaît aussi un certain succès. Ceci est dû principalement au fait que le protocole est quasiment garanti de pouvoir « sortir » des entreprises en traversant sans problème leurs dispositifs de sécurité d'accès au réseau extérieur.

---

[24] http://www.mirc.com/
[25] http://swatit.org/bots/gtbot.html

Le bot s'identifie auprès du serveur C&C, un serveur Web, par une requête HTTP. L'attaquant peut ensuite envoyer des commandes via les réponses HTTP. Etant donné le caractère non permanent de ce type de connexion, le bot doit se connecter régulièrement à son serveur.

L'attaquant dispose en général d'une page web de contrôle de son botnet. Eventuellement et si la charge le permet, il peut utiliser SSL (TLS) pour masquer le trafic entre ses bots et son serveur.

### IV.1.3 - P2P

Que ce soit en IRC ou en HTTP, l'utilisation d'un serveur central unique a un inconvénient majeur pour le pirate propriétaire d'un botnet : l'élimination du serveur et de l'enregistrement DNS correspondant signifie la perte du botnet dans son ensemble. L'adoption de protocoles *Peer-to-Peer* (P2P) a constitué une évolution majeure apparue au grand jour en 2004 avec Phatbot [26], un descendant d'Agobot. Phatbot utilisait une version modifiée de WASTE, protocole P2P à l'origine chiffré, et les serveurs de cache de Gnutella, pour que les bots puissent se trouver et se connecter.

L'avantage immédiat du P2P pour le pirate, est qu'il le rend plus difficile à repérer, préservant ainsi son activité délictueuse. Au lieu de se connecter à un serveur central, le pirate se connecte comme un autre membre du réseau pour envoyer ses commandes aux bots.

Storm/Peacomm est un autre exemple caractéristique des botnets P2P [27], décrit en détail dans la suite de ce document. Il a également fait l'objet d'une analyse particulièrement intéressante lors de la conférence Black Hat USA 2008 à Las Vegas en août 2008 [28].

## *IV.2 - Analyses de cas, étude(s) statique(s) et dynamique(s).*

### IV.2.1 - Storm

*Storm-Worm,* également connu sous les noms de *Zhelatin* et *Peacomm,* est un ver informatique apparu pour la première fois en janvier 2007. En septembre 2007, MSRT (Microsoft Malicious Software Removal Tool) l'a éliminé sur plus de 274 000 machines [29]. De son côté, SecureWorks a estimé qu'entre juin et juillet 2007, la taille du botnet atteignait 1,7 million de machines [30]. A cette même date, IronPort annonçait que durant l'été, 1,4 million de machines émettait chaque jour du spam alors que d'autres sources comptabilisaient plus de 10 millions de machines infectées par ce même virus [31].

Storm représente une évolution majeure dans le paysage des malwares : les antivirus sont inefficaces, et il utilise les techniques *fast-flux* et P2P. Son objectif principal est de transformer les PC infectés en zombies utilisés par les malfaiteurs pour diverses tâches, la principale d'entre elles étant l'envoi de spam.

---

[26] http://www.secureworks.com/research/threats/phatbot/
[27] Peer-to-Peer Botnets: Overview and Case Study – http://www.usenix.org/event/hotbots07/tech/full_papers/grizzard/grizzard_html/
[28] Inside the Storm: Protocols and Encryption of the Storm Botnet – Joe Stewart, GCIH, Director of Malware Research, SecureWorks – http://www.blackhat.com/presentations/bh-usa-08/Stewart/BH_US_08_Stewart_Protocols_of_the_Storm.pdf
[29] http://blogs.technet.com/antimalware/archive/2007/09/20/storm-drain.aspx
[30] http://www.secureworks.com/media/press_releases/20070802-botstorm
[31] http://blog.washingtonpost.com/securityfix/2007/10/the_storm_worm_maelstrom_or_te.html#more

Ce ver a particulièrement attiré l'attention des chercheurs en sécurité depuis son apparition et ce pour plusieurs raisons :

- il est à l'origine du plus gros réseau de zombies de l'année 2007, contrôlant plusieurs dizaines de milliers de machines. ; il est l'un des premiers à utiliser un réseau P2P entièrement décentralisé comme canal de contrôle ;
- Lors de sa « mise en service », le niveau de sophistication de ce réseau était particulièrement élevé par rapport à l'existant.

#### IV.2.1.1 - Infection

La compromission des machines victimes se fait de manière classique :

- par l'envoi d'un mail contenant une pièce jointe "alléchante" ;
- en invitant la personne à visiter un site web "intéressant" qui tentera d'exploiter diverses failles du navigateur et de ses plugins ou, en dernier ressort, en poussant l'utilisateur à installer une application infectée.

Il est à noter que StormWorm a très bien utilisé divers événements d'actualités (tempêtes en mer du Nord, conflit en Irak, fête de la St Valentin ou Noël…) pour propager des mails alléchants, encourageant des utilisateurs novices à les ouvrir.

Une fois l'infection réalisée, le malware installe un *rootkit*. Ce dernier désactive la plupart des antivirus et dissimule le processus de son malware.

#### IV.2.1.2 - Protections

Le bot dispose d'un certain nombre de protections visant à ralentir et à empêcher sa détection : il refuse de s'exécuter s'il est lancé dans une machine virtuelle, tente de leurrer les sandbox en retardant l'exécution du code principal [32]. Mais surtout le binaire principal est chiffré, afin d'empêcher les antivirus de le détecter aisément à l'aide de signatures : l'algorithme et la clef de chiffrement changent quasiment à chaque variante. Des taux de mutation d'une fois par heure ont été enregistrés.

De plus, les serveurs de distribution du malware sont dissimulés derrière la technologie *fast-flux* qui rend très difficile voire impossible la détection des serveurs d'origine.

#### IV.2.1.3 - Protocole

Comme précisé auparavant, le contrôle des PC zombies se fait à l'aide d'un réseau peer-to-peer, en l'occurrence celui du logiciel OverNet [33]. Il s'agit d'un protocole implémentant une table de hachage distribuée (basée sur la spécification Kademlia [34]) : il permet de stocker des valeurs dans le réseau sans utiliser de serveur central pour l'indexation. En pratique, le bot contient une liste de nœuds appartenant au réseau qu'il va contacter lors de son initialisation. Une fois qu'il aura acquis une connaissance suffisante du réseau, il va récupérer les commandes qu'il devra exécuter. Pour ce faire, il va générer un hash à partir de la date du jour, d'une clef secrète et d'une valeur aléatoire (parmi 32) et rechercher des nœuds ayant connaissance de ce hash sur le réseau. Il va ensuite recevoir les réponses, qui correspondent à un couple IP/Port TCP auquel il va se connecter pour récupérer les commandes.

---

[32] Peacomm.C - Cracking the nutshell: http://www.reconstructer.org/papers.html

[33] Wikipedia - Overnet : http://en.wikipedia.org/wiki/Overnet

[34] Wikipedia - Kademlia : http://en.wikipedia.org/wiki/Kademlia

Une fois la connexion TCP effectuée, les deux machines vérifient qu'elles possèdent bien la même clef unique de 4 octets. Ensuite, le bot télécharge un ensemble de commandes compressées en zlib spécifiant le spam à envoyer.

Les deux clefs de chiffrement que l'on retrouve dans le protocole - pour la recherche et à l'établissement de la connexion TCP - laissent penser que les propriétaires du botnet souhaitent en louer une partie. En effet, il est possible de commander une sous partie du botnet en utilisant des clefs de chiffrement correspondant à une variante en particulier.

### IV.2.2 - BlackEnergy

En octobre 2007, le CERTA décrivait le réseau de machines zombies « BlackEnergy » [35]. A la date de l'analyse, sa principale motivation était l'attaque en déni de service.

Reprenant une étude d'Arbor Networks [36], le bulletin d'activité du CERTA indiquait que ce réseau était constitué d'un nombre limité de machines, localisées en Asie et en Europe de l'Est.

Ce qui était intéressant dans BlackEnergy, c'était son mode de communication s'appuyant sur des requêtes HTTP valides via des serveurs Web compromis et une utilisation de PHP et de MySQL. Les machines infectées cherchaient alors leurs instructions sur ces serveurs, par des requêtes HTTP en mode POST [37]. Dans l'exemple décrit par l'article, il s'agissait d'une requête de la forme :

```
POST /dot/stat.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 1.1.4322)
Host: **********
Content-Length: 31
Cache-Control: no-cache
id=xxxxxxxxxxxxxxx&build_id=yyyyy
```

La machine émettrice incluant son identifiant dans l'en-tête, le serveur lui répondait en retour :

```
HTTP/1.1 200 OK
Date: Tue, XX Sep 2007 08:30:13 GMT
Server: Apache/2.0.59 (Unix) FrontPage/5.0.2.2635 PHP/5.2.3
mod_ssl/2.0.59 OpenSSL/0.9.7e-p1
X-Powered-By: PHP/5.2.3
Content-Length: 80
Connection: close
Content-Type: text/html
MTA7MjAwMDsxMDswOzA7MzA7MTAwOzM7MjA7MTAwMDsyMDAwI3dhaXQjMTAjeENSMl8yN
(...)
```

Ici, la dernière ligne de données transmises correspond à la commande attendue. Elle peut préciser les paramètres d'une attaque par inondation de trames, ou d'un fichier à télécharger et à exécuter.

Les moyens d'identifier ce canal de communication n'étant pas simples à mettre en œuvre, la question posée par le bulletin était donc : que peut-on détecter ? En réponse, il proposait :

---

[35] http://www.certa.ssi.gouv.fr/site/CERTA-2007-ACT-042/index.html

[36] J. Nazario, Arbor Networks, "BlackEnergy DDoS Bot Analysis", octobre 2007 : http://atlas-public.ec2.arbor.net/docs/BlackEnergy+DDoS+Bot+Analysis.pdf - http://asert.arbornetworks.com/2007/10/blackenergy-ddos-bot-analysis-available/

[37] Contrairement à la méthode GET dans laquelle les informations envoyées au serveur sont encodées à la suite de la ressource après le symbole '?' dans l'url, la méthode POST permet d'envoyer ces informations dans le corps du message d'une requête HTTP.

- l'identification des requêtes vers le serveur distant, si celui-ci est connu comme étant compromis ;
- des recherches de contenus sur des chaînes de caractères particulières, mais pouvant provoquer beaucoup de fausses alertes, ou faux positifs.

# V - Prévention, Détection et Interception

En entreprise, la prévention doit être prise en compte dans la politique de sécurité. Comme tout aspect de la sécurité des systèmes d'information, la lutte contre les logiciels malveillants doit être abordée selon trois axes d'égale importance : les procédures, la technique et les personnes. Par exemple, si une procédure rend obligatoire la présence d'un antivirus à jour et l'installation immédiate des mises à jour de sécurité de tous les composants des postes de travail, des mesures techniques peuvent automatiser l'application de ces mesures, et une sensibilisation des utilisateurs doit les informer de leur responsabilité dans le bon déroulement de ces opérations.

En ce qui concerne les utilisateurs non administrés (PME ou domestiques), l'accent devrait être mis sur l'utilisation responsable de l'Internet, en préalable au discours sur les protections techniques.

La détection des botnet passe généralement par l'analyse du trafic (depuis l'Internet jusqu'au réseau local et inversement). Au niveau local, et tout comme il sait le faire pour les autres types de malware, un antivirus standard à jour, avec le mode de détection « à l'accès » activé, pourra le plus souvent détecter l'arrivée du programme de type robot avant son implantation. Si celui-ci est déjà sur la machine, la détection, a posteriori, s'avère plus périlleuse du fait de l'utilisation de plus en plus fréquente d'un rootkit. L'utilisation d'un pare-feu personnel peut, dans certains cas, permettre de détecter ce type de menaces.

Au niveau réseau, l'analyse du trafic de contrôle (IRC, interrogations DNS suspicieuses, tunnels HTTP, etc.) ou du trafic lié à des attaques (DDoS, envoi massif de SPAM, scans réseaux, etc.) peut révéler la présence de machines participant à un botnet. L'analyse peut également permettre la localisation du système de commande et de contrôle.

Une détection peut évidemment être faite directement sur la machine suspecte car les mécanismes de dissimulation sont sensiblement identiques à ceux utilisés dans le monde des virus, vers et chevaux de Troie.

## *V.1 - Niveau réseau - aspects administrateur*

Différentes techniques peuvent être utilisées pour éviter l'infection et se protéger des machines qui appartiennent à un botnet.

### V.1.1 - Listes noires, listes maintenues

Appelées RBLs (Real-time Black Lists) ou DNSBL (DNS-based Blackhole List), ce sont des listes d'adresses IP de machines accueillant des programmes zombies connus pour aider, accueillir, produire ou retransmettre des spams, mener des attaques en DDoS ou fournir un service pouvant être utilisé comme support à ces activités : OpenSMTP Relay, Open Proxy List (OPL).

La gestion de ces listes est, entre autres, exposée dans le livre blanc du *College of Computing (Georgia Institute of Technology) : Revealing Botnet Membership Using DNSBL Counter-Intelligence* [38].

[38] http://www.cc.gatech.edu/~feamster/publications/dnsbl.pdf

### V.1.2 - Observation du trafic

C'est l'une des méthodes les plus utilisées. Elle s'applique principalement aux botnets utilisant des canaux IRC (Internet Relay Chat) comme vecteur de transmission des commandes.

L'identification du botnet passe par l'interception de commandes non standard ou la mesure de réactivité des clients pour différencier les bots des humains.

C'est ainsi qu'il a été démontré [39] que les bots IRC étaient en attente la plupart du temps (c'est-à-dire silencieux), alors qu'ils répondaient plus rapidement qu'un humain dès l'apparition d'une commande.

L'analyse du trafic lié à la distribution massive de spam ou des attaques de DDOS peut également permettre de mesurer l'importance des attaques. Des sociétés telles que Shadowserver [40] proposent diverses statistiques en ce sens.

Avant de mettre en œuvre de telles solutions à base d'écoute, il est nécessaire de solliciter un conseil juridique. En effet la surveillance des réseaux intra-entreprise par les employeurs est une question délicate à concilier avec d'autres impératifs, et notamment avec les considérations de protection des correspondances privées.

### V.1.3 - Empreinte (hash)

Dans un document intitulé: *Locating Zombie Nodes and Botmasters in Decentralized Peer-to-Peer Botnets*, des universitaires américains décrivent une méthode de détection pouvant s'appliquer aux botnets utilisant l'architecture décentralisée des réseaux P2P [41] tels que W32/Nuwar (alias Storm).

Dans une telle architecture, les nœuds compromis sur le réseau peuvent être identifiés par leur empreinte (hash) et ainsi permettre de remonter aux machines infectées et au poste de commande.

### V.1.4 - Protection du réseau local

Les moyens de prévention sont ceux communs aux autres malware. Ils sont décrits dans le document « les virus informatiques » disponible gratuitement sur le site du CLUSIF [42].

La liste ci-dessous - qui n'a pas la prétention d'être exhaustive - pourra aussi vous aider :

- Apportez une attention particulière à la gestion des mots de passe et des comptes utilisateurs.
- Mettez en place une bonne politique de gestion des correctifs de sécurité (systèmes et logiciels) aussi bien sur les serveurs et équipements réseau que sur les postes clients.
- Prenez en compte la gestion des postes nomades.
- Gérez la lutte anti-spam dès la passerelle de messagerie.
- N'accordez que des droits restreints aux utilisateurs pour limiter la propagation des codes malveillants.

---

[39] ftp://www.tik.ee.ethz.ch/pub/students/2003-2004-Wi/MA-2004-01.pdf
[40] http://www.shadowserver.org/wiki/pmwiki.php/Stats/DDoSCharts
[41] https://www.os3.nl/_media/2007-2008/students/matthew_steggink/rp1/p2pdetect_conceptpaper.pdf?id=2007-2008%3Astudents%3Amatthew_steggink%3Arp1&cache=cache
[42] http://www.clusif.asso.fr/fr/production/ouvrages/pdf/VirusInformatiques.pdf

- Suivez avec vigilance les applications PHP et ASP et serveurs installées sur les serveurs web.
- Cloisonnez votre réseau à l'aide de filtres sur les routeurs et de pare-feu. Mettez en place des réseaux virtuels (VLAN).
- Mettez en place des outils de détection et de prévention d'intrusion (IDS, IPS).
- Analysez les journaux d'évènements pour repérer les machines infectées et les attaques subies. L'analyse des traces réseaux produites par un «bot» peut permettre de localiser la machine infectée. Ces traces peuvent provenir d'outils de métrologie, de fichiers journaux de routeurs, de pare-feu et de systèmes de détection d'intrusion.

## *V.2 - Niveau local, aspects utilisateur*

Dans le meilleur des cas, pour l'utilisateur, la présence d'un programme robot se traduit, le plus souvent, par une simple alerte virale. Si le nom du programme détecté ne contient pas la chaîne de caractère « BOT », il devra consulter une encyclopédie virale pour espérer apprendre que le programme qu'il vient de rencontrer comporte une fonctionnalité de type robot (exemple W32/Nuwar, alias Storm).

Pour protéger le poste client, on appliquera notamment aussi les recommandations générales liées à la lutte antivirale présentée dans le document « les virus informatiques ».

En complément, rappelons que des moyens de prévention et de protection doivent être mis en place à tous les niveaux (poste de travail, ressources partagées et passerelles Internet) :

- l'utilisateur ne doit pas travailler en mode administrateur. La stratégie de sécurité établie pourra aller jusqu'à n'autoriser que l'installation et l'exécution de programmes de confiance.
- le déploiement des correctifs et des mises à jour (système, antivirus et applications) est impératif. Il est recommandé qu'il soit automatique. On ne doit pas oublier les serveurs.
- le mode d'analyse à l'accès de l'antivirus doit être activé en permanence. En entreprise, le fait de désactiver l'anti-virus doit être considéré comme une faute grave.
- la mise à jour automatique peut avoir été désactivée par un programme malveillant inconnu. Il faut vérifier régulièrement la date de dernière mise à jour de votre antivirus.
- en cas de doute, l'utilisateur doit savoir comment exécuter efficacement une analyse à la demande.
- si la station locale doit contenir des données utilisateurs, il est judicieux de les stocker sur un disque logique séparé (disque D:) en limitant l'usage du premier disque logique C: au système d'exploitation et aux programmes. Cela permet de réinstaller le système d'exploitation tout en préservant les données de l'utilisateur.
- l'antivirus du poste local doit prendre en compte l'analyse temps réel des fichiers distants manipulés (ouverture, fermeture), et ceci malgré une éventuelle dégradation des performances.
- les accès aux ressources partagées doivent être protégés par des mots de passe robustes, choisis selon des règles strictes.

- les exécutables stockés sur une ressource partagée doivent être en lecture seule (*read-only*), sauf impératif contraire.
- les applications créées en interne (dans l'entreprise) doivent faire l'objet d'un contrôle d'intégrité au moment de leur lancement.
- un anti-virus adéquat doit être déployé à tous les niveaux. Les journaux d'alerte doivent être sauvegardés et centralisés
- il faut connaître ou disposer d'une source fiable et sécurisée qui pourra être utilisée pour télécharger les versions complètes et à jour des anti-virus dont on a la licence. Il est bon de pouvoir disposer de supports amovibles auto-amorçables et protégés en écriture.
- l'activation du pare-feu local (ou l'installation s'il agit d'une machine au système d'exploitation ancien) est très fortement recommandée, elle est indispensable sur les postes nomades.
- anti-spam, filtrage de contenu, d'URL et de port, IDS, IPS sont autant de produits complémentaires qui amélioreront le niveau de sécurité global de l'entreprise dans le respect des lois en vigueur sur la protection de la vie privée.
- la messagerie électronique doit être utilisée de manière rigoureuse :
    - lire, dans la mesure du possible, les messages au format texte plutôt que HTML.
    - rester vigilant face aux spam. Quelques uns d'entre eux échappent toujours aux divers moyens de filtrage et peuvent s'avérer dangereux.
    - ne pas répondre aux messages de type spam, ni suivre les liens proposés.
    - éviter d'ouvrir inconsidérément les pièces jointes, notamment si aucun document n'est attendu. Si nécessaire, les enregistrer sur le disque dur du PC, puis les passer au contrôle de l'antivirus.
    - ne pas oublier que les éditeurs de logiciels n'envoient jamais de correctifs par le biais de la messagerie électronique.
- des messages arrivent de plus en plus fréquemment par le biais des réseaux sociaux et de la messagerie instantanée. Les bonnes pratiques s'appliquant à la messagerie électronique sont, ici aussi, adaptées.
- la navigation sur Internet peut s'avérer risquée, même face à des sites de confiance, des sites institutionnels ou des sites d'organismes bien connus.
    - ne pas répondre précipitamment « oui » ou « ok » aux fenêtres d'alertes que votre navigateur peut afficher au cours de votre navigation.
- le téléchargement est toujours une opération risquée :
    - éviter de rapatrier des programmes aux sources peu fiables ou inconnues.
    - privilégier le site de l'éditeur pour télécharger un programme ou une mise à jour le concernant
    - vérifier à l'aide d'un antivirus tout programme inconnu téléchargé.

# VI - Conclusion

Les programmes robots sont aujourd'hui l'une des principales menaces du Net et il ne se passe pas une semaine sans qu'un nouveau botnet ne soit mis à jour par tel ou tel éditeur de solutions de sécurité. Alors que l'un d'entre eux annonçait que plus de 500 entreprises et agences gouvernementales du Royaume-Uni détenaient des machines infectées [43], des chercheurs canadiens accusaient la Chine ou ses « hackers patriotes » d'utiliser un autre botnet à des fins d'espionnage [44].

Chevaux de Troie ou virus aux capacités étendues, ces programmes malveillants sont souvent difficilement repérables une fois implantés sur le poste de travail. A l'aube de l'an 2000, nous écrivions qu'en cas de suspicion, et face à plusieurs anti-virus muets, le reformatage d'une machine n'était pas la bonne solution. Aujourd'hui les choses ont changé et, malgré les techniques présentées plus haut, les administrateurs sont plus démunis que par le passé. A condition de savoir ce que l'on cherche, une analyse « offline » à partir d'une autre machine est toujours praticable ; mais, face au doute, la réinstallation totale d'une machine douteuse nous semble être la meilleure formule.

Longtemps cantonnés dans les environnements Microsoft, les robots font aussi leur apparition sous Macintosh [45] et sous Linux [46]. La fin des programmes malveillants n'est donc pas pour demain.

[43] Un gigantesque botnet de 1,9 millions de PC mis à jour : http://micro.lemondeinformatique.fr/actualites/lire-un-gigantesque-botnet-de-1-9-millions-de-pc-mis-a-jour-2600.html
[44] Tracking GhostNet: Investigating a Cyber Espionage Network : http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network
[45] The new iBotnet: http://www.virusbtn.com/virusbulletin/archive/2009/04/vb200904-ibotnet (réservé aux abonnés).
[46] First ever Linux botnet?: http://www.linuxdevices.com/news/NS2300669830.html

# VII - ANNEXE 1 : Les techniques de type fast-flux

Les botnets sont régulièrement utilisés comme passerelle intermédiaire entre les cybercriminels et leurs victimes. Ils offrent discrétion, camouflage et rentabilité. Les techniques utilisées prennent les noms de *fast-flux*, *single* ou *double-flux* ou *RockPhish*.

Comme préalable à ces descriptions penchons nous en premier lieu sur un exemple de campagne de spam.

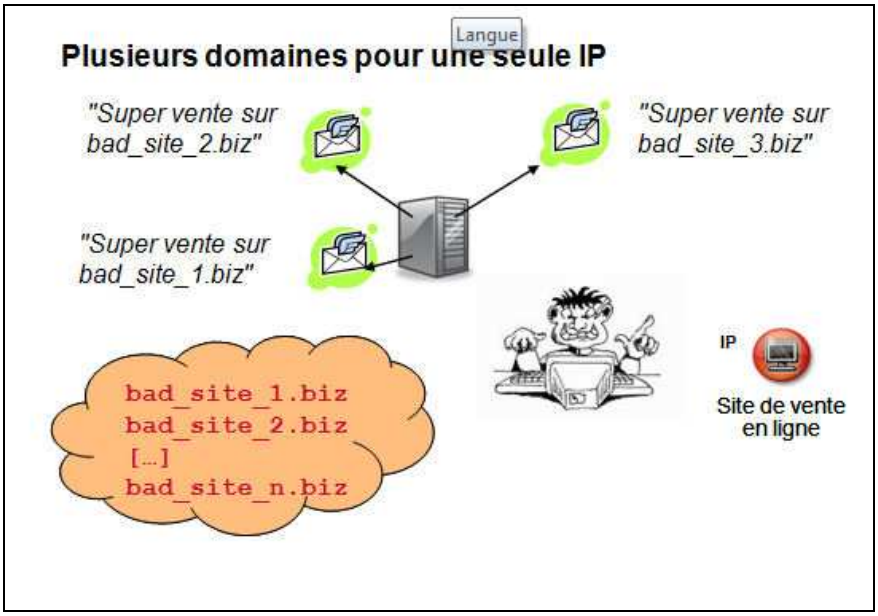


**Figure 5 : Représentation d'une campagne de spam où son auteur dispose de nombreux noms de domaines derrière une seule adresse IP**

Ici, le spammeur dispose de nombreux noms de domaine. Il en achète sans cesse de nouveaux en utilisant des numéros de cartes de crédit volées et s'en sert au gré des interruptions de service qui s'opèrent plus ou moins rapidement selon la vigilance et l'honnêteté des fournisseurs d'accès. Il peut aussi abuser de certains mécanismes lors du dépôt de noms de domaine, tels que le « domain tasting », pratique consistant à annuler l'achat d'un nom de domaine moins de 5 jours après son dépôt… ainsi, le nom de domaine n'est pas facturé (attention, mise à jour de l'ICANN en janvier 2008).

Une de ses machines contient son site ; il peut être dédié à la vente de médicaments ou de produits de luxe contrefaits.

Afin de tromper les logiciels antispam, les envois sont personnalisés avec du bruit de fond et des textes aléatoires. Pour plus de diversification, et grâce aux divers noms de domaine qu'il possède, son logiciel d'émission fait varier l'URL de son site en fonction des envois.

Lorsqu'une victime cherche à suivre le lien qui lui est proposé, un processus demande, au serveur de nom local, l'adresse IP de la machine correspondant à l'URL qui lui a été donnée.

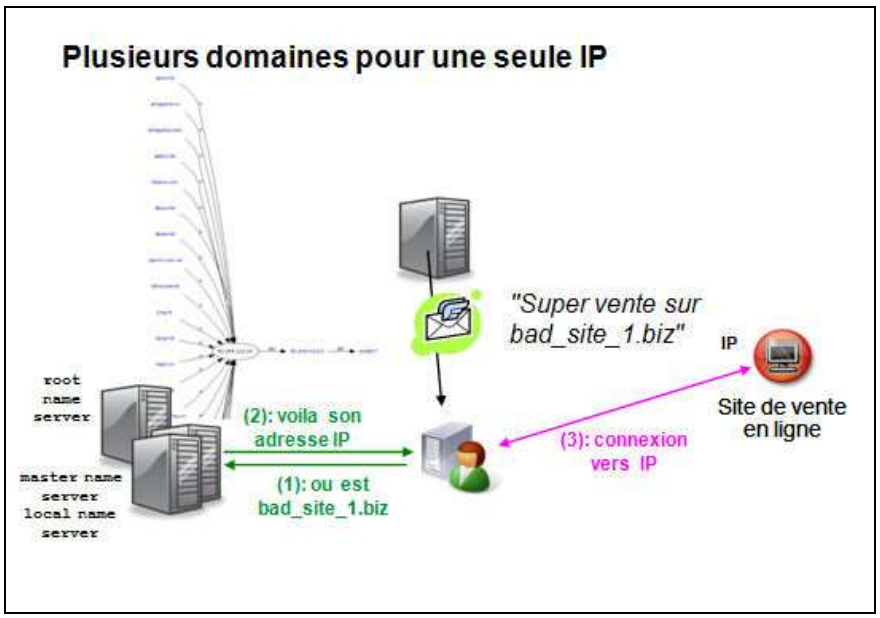


**Figure 6 : Représentation d'une campagne de spam où son auteur dispose de nombreux noms de domaines derrière une seule adresse IP**

Si l'information existe à ce niveau (mécanisme de cache) elle est directement renvoyée au demandeur. Dans le cas contraire, et si le lien est toujours valide, l'adresse IP recherchée n'est retournée qu'après interrogation d'un serveur racine et/ou d'un serveur primaire.

Des dizaines de noms de domaine différents peuvent donc pointer vers une même machine.

La figure suivante vous donne un exemple du résultat qui peut être obtenu grâce à cette méthode.


**Figure 7 : Exemples de messages émis par une campagne de spam où son auteur dispose de nombreux noms de domaines derrière une seule adresse IP**

Avec le phishing, les méthodes se complexifient. Cette courbe issue des statistiques de APWG ne s'attarde pas sur le nombre de victimes ; en forte hausse depuis deux ans.

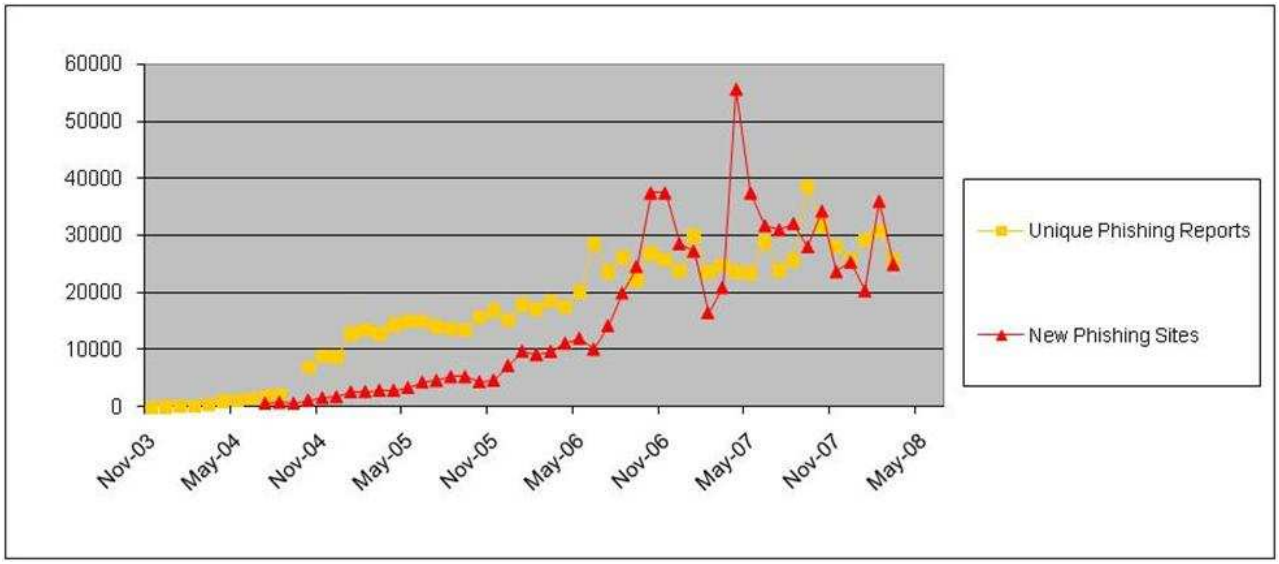


**Figure 8 : Statistiques de l'APWG**

Elle nous montre que depuis la mi-2006, le nombre total d'incidents (avec ou sans victime) reste stable.

Ce qui attire l'œil, ce sont les pics de novembre 2006 et surtout d'avril 2007. La question qui se pose est la suivante : comment peut-on avoir trois fois plus de sites de phishing que d'attaques recensées ?

La réponse se nomme *RockPhish* ; et pour mieux la comprendre, nous allons complexifier l'exemple précédent en étudiant tout d'abord deux techniques intermédiaires se cachant derrière le néologisme *fast-flux* : le *single-flux* et le *double-flux*. Toutes ces méthodes utilisent des réseaux de robots.

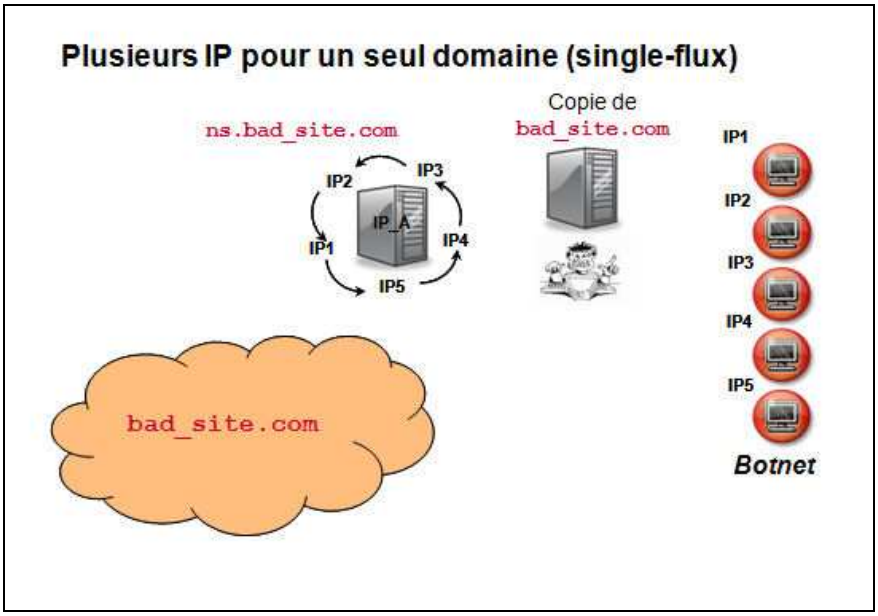


**Figure 9 : Eléments entrants dans une attaque *single-flux* : un unique domaine se voit attribuer, par roulement, plusieurs adresses IP**

Dans le cas du *single-flux* le pirate ne possède qu'un seul domaine. Grâce au soutien d'un fournisseur d'accès peu scrupuleux, il gère son propre serveur de nom de domaine. Il a

également à sa disposition un réseau de machines compromises qu'il utilise comme plateforme relais entre les victimes et son site.

L'utilisation de dates d'expiration DNS très courtes associées, par roulement, aux diverses adresses IP de ses machines zombies permet de changer, en permanence, une adresse physique factice utilisée pour rejoindre le site miroir à protéger.

Celui-ci n'en est que mieux dissimulé.

Lorsque la victime cherche à atteindre le site miroir, une requête est envoyée vers le serveur de nom ayant autorité sur la zone.

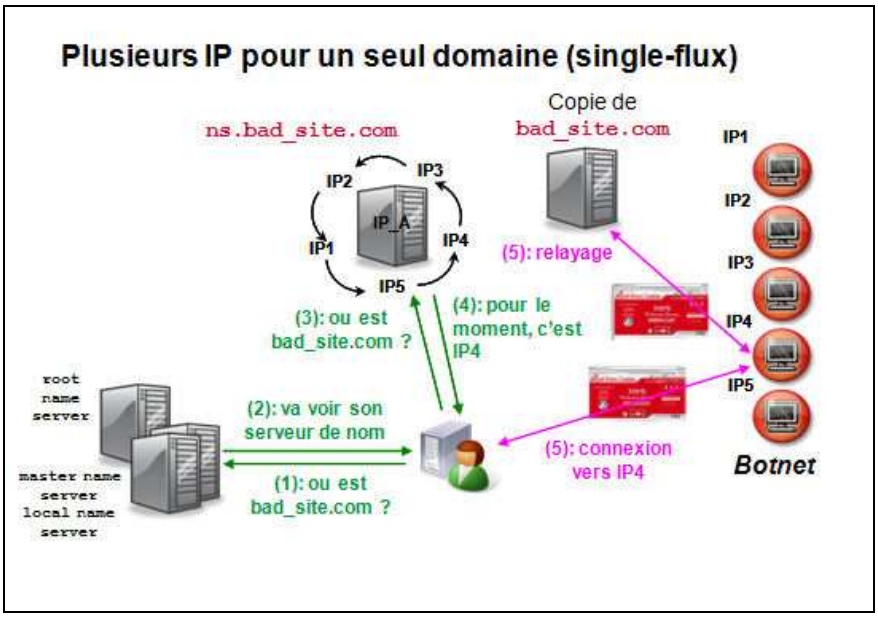


**Figure 10 : Représentation d'une attaque *single-flux* : un unique domaine se voit attribuer, par roulement, plusieurs adresses IP**

La durée de vie de l'adresse ne dépassant pas quelques dizaines de minutes, il n'y a généralement pas de solution en cache.

C'est donc le serveur de noms aux mains du pirate qui est interrogé.

L'adresse IP d'un des robots lui est renvoyée. Pendant les quelques minutes que durera la transaction, il relayera le trafic puis disparaîtra en rendant plus difficile la localisation et donc la neutralisation des sites essentiels.

Ici, il faut comprendre que l'adresse visible depuis le PC de l'internaute en train de naviguer est en fait celle d'une machine compromise (un robot) qui sert à son tour de relais (proxy) vers le vrai site. L'internaute n'a ainsi jamais connaissance de la véritable adresse IP du serveur principal.

Voici ici l'exemple d'un site de jeu en ligne utilisant la technique *single-flux*.

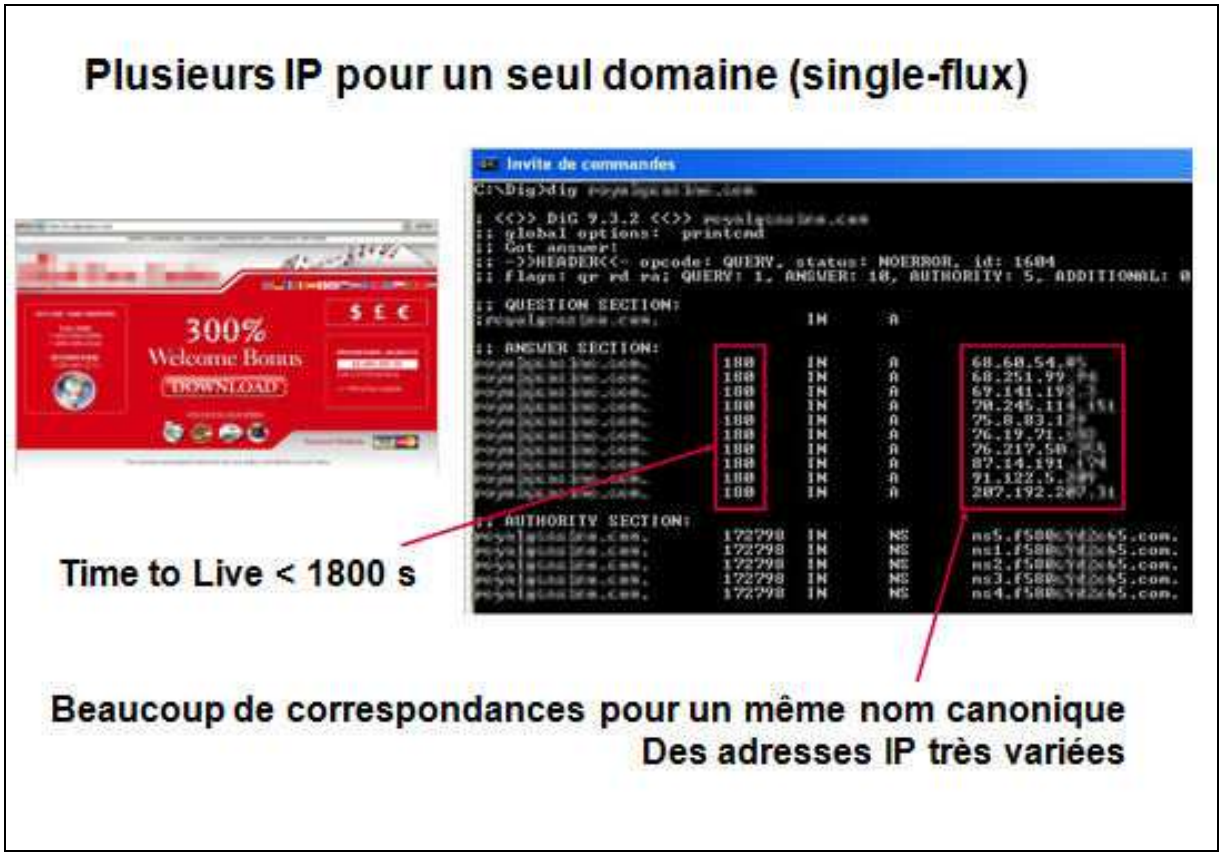


**Figure 11 : Exemple d'un site protégé par la technique *single-flux* : son unique domaine se voit attribuer, par roulement, plusieurs adresses IP**

Bien connu dans l'environnement UNIX, l'utilitaire dig (domain information groper) est similaire à nslookup. C'est une version transcrite pour Windows qui a permis l'étude de ce cas.

Les dates d'expiration sont ici très courtes (1800 secondes) et les adresses IP très variées. C'est le signe d'un camouflage selon la technique *fast-flux/single-flux*.

Il est possible d'améliorer le camouflage en faisant varier l'adresse IP du site à protéger ainsi que celle des serveurs de noms qui les définissent dans l'architecture DNS. C'est le *double-flux.*

Le pirate a ici, à sa disposition, un véritable poste de commande et de surveillance. Ce n'est pas « le top » en matière de dissimulation, mais on s'en approche.

Son réseau de robots va intégralement le protéger. Ces machines ne seront plus seulement là pour relayer le trafic http ; elles simuleront les serveurs de noms de domaine et retransmettront l'adresse IP qui, comme dans le cas précédent, ne sera valide qu'à un instant donné.

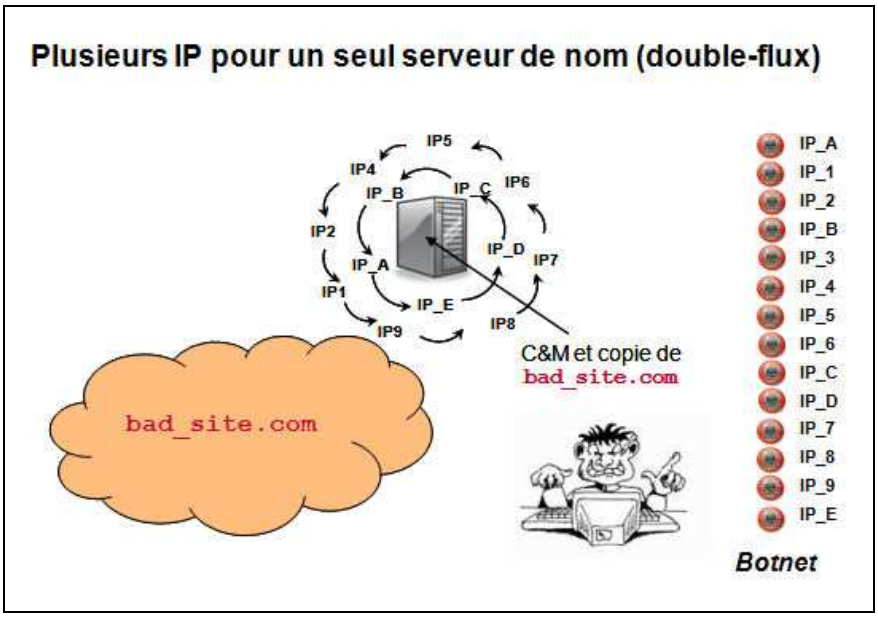


**Figure 12 : Eléments entrants dans une attaque *double-flux* : un unique domaine se voit attribuer, par roulement, plusieurs adresses IP. L'adresse IP derrière laquelle se cache le serveur de noms est aussi variable**

Lorsque la victime cherche à atteindre le site qu'elle souhaite visiter, une requête est envoyée vers le serveur de nom ayant autorité sur la zone. Tout comme précédemment, la faible durée de vie de l'adresse amène l'interrogation du serveur de nom aux mains du pirate.

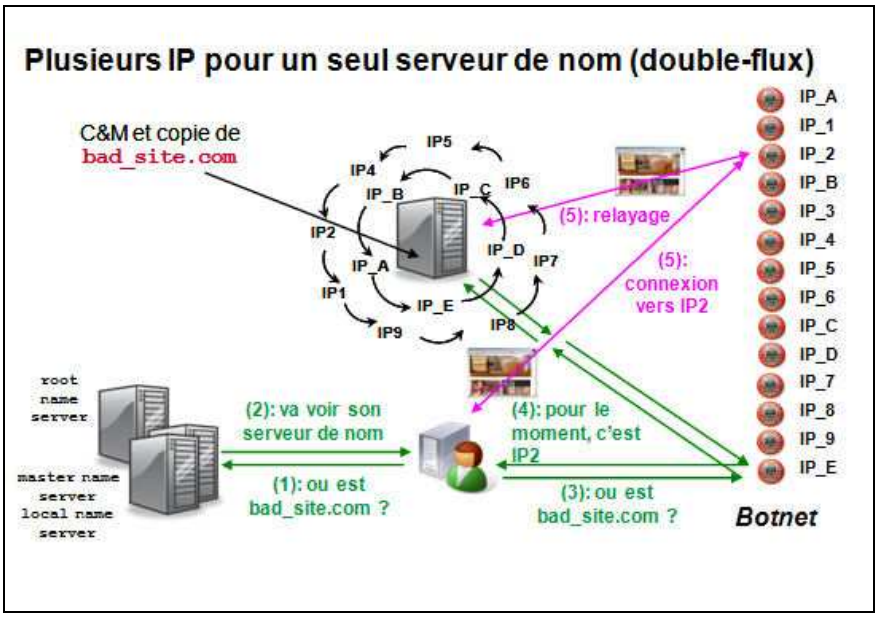


**Figure 13 : Représentation d'une attaque *double-flux* : un unique domaine se voit attribuer, par roulement, plusieurs adresses IP. L'adresse IP derrière laquelle se cache le serveur de noms est aussi variable**

Utilisée une première fois à ce niveau, la technique du *fast-flux* provoque une redirection de la requête vers une machine du botnet (*fast-flux* au niveau des serveurs de noms de domaine - IP_A à IP_E). Celle-ci demande la réponse au poste de contrôle et la retransmet au demandeur en utilisant une seconde fois la même méthode (*fast-flux* au niveau du site web à protéger - IP_1 à IP_9).

En conséquence, c'est l'adresse IP d'une autre machine zombie qui est renvoyée à la victime et c'est elle qui relayera le trafic en préservant l'anonymat du pirate.

Comme l'image rendue floue ci-dessous le laisse deviner, ce second exemple touche un site pour adulte qui cherche à rester discret quant à ses origines. Deux commandes dig lancées à quelques dizaines de minutes d'intervalle nous montrent le résultat obtenu.

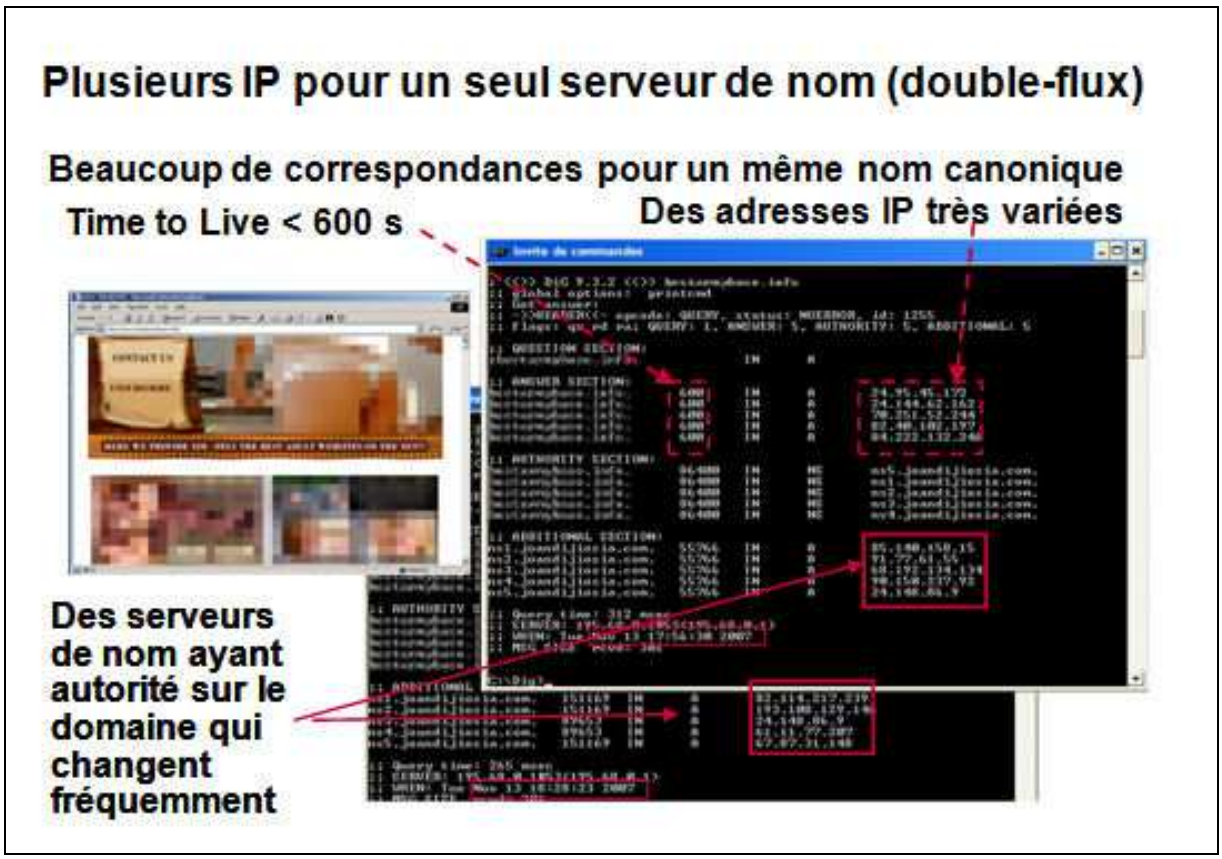


**Figure 14 : Exemple d'un site protégé par la technique *double-flux* : son domaine se voit attribuer, par roulement, plusieurs adresses IP. Les adresses IP derrière lesquelles se cachent les serveurs de noms sont aussi variables**

Les dates d'expiration sont réduites à 10 minutes (600 secondes) et les adresses IP du site sont très variées (*fast-flux* sur le serveur web à protéger). Il en est de même pour les serveurs de noms de domaine qui ont changé dans un court laps de temps (*fast-flux* sur le serveur de noms de domaine).

En réussissant à combiner les trois exemples précédents, on se rapproche de la méthode employée par le mystérieux groupe *RockPhish*. Les ingrédients sont :

- de nombreux noms de domaine,
- un réseau de botnet de type *fast-flux* en mode *double-flux*,
- un logiciel spécialisé qui se charge d'émettre des courriels de type phishing où chaque destinataire est affecté d'un index. Celui-ci se retrouve comme paramètre d'URL ; il est réutilisé au niveau du site miroir pour peu que la victime s'y connecte.

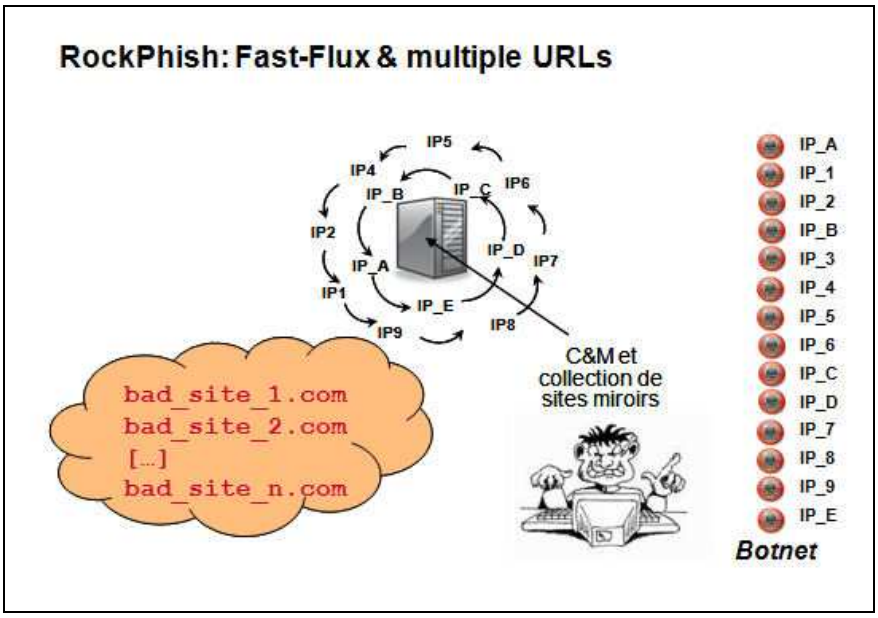


**Figure 15 : Eléments entrants dans une attaque *RockPhish* ou *double-flux* avec multiples URL : de multiples domaines se voient attribuer, par roulement, plusieurs adresses IP. Les adresses IP derrière lesquelles se cachent les serveurs de noms sont variables. Elles pointent sur de multiples machines.**

La seule vision des URL donne une idée de la complexité de l'attaque et montre qu'il s'agit d'un travail de professionnel.

Le nom des domaines hébergeurs varie ; les serveurs de noms de domaine également.

Le poste de commande et de contrôle gère, en temps réel, la structure du réseau ; n'oublions pas qu'il s'agit ici, en grande partie, d'un réseau de machines compromises (un botnet).

L'index est là pour assurer une redirection correcte en fonction des victimes, des banques, des machines à activer et du groupe de fraudeurs à qui doit profiter l'attaque.

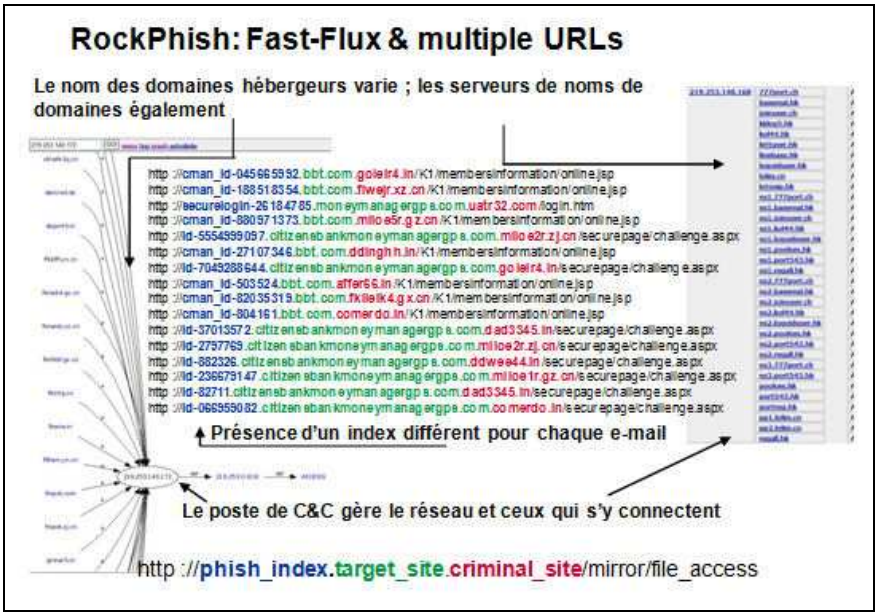


**Figure 16 : Exemple d'une attaque *RockPhish* ou *double-flux* avec multiples URL : de multiples domaines se voient attribuer, par roulement, plusieurs adresses IP. Les adresses IP derrière lesquelles se cachent les serveurs de noms sont variables. Elles pointent sur de multiples machines.**

# VIII - ANNEXE 2 : Exemple de commandes d'un bot IRC

Cet exemple liste les commandes du bot Agobot reçues en réponse à la commande IRC `commands.list` :

```
[14:40] <pirate> .commands.list
[14:40] <Ago-dktj> -[ command list ]-
[14:40] <Ago-dktj> 1. / "commands.list" / "Lists all available commands"
[14:40] <Ago-dktj> 2. / "cvar.list" / "prints a list of all cvars"
[14:40] <Ago-dktj> 3. / "cvar.get" / "gets the content of a cvar"
[14:40] <Ago-dktj> 4. / "cvar.set" / "sets the content of a cvar"
[14:40] <Ago-dktj> 5. / "cvar.loadconfig" / "loads config from a file"
[14:40] <Ago-dktj> 6. / "cvar.saveconfig" / "saves config to a file"
[14:40] <Ago-dktj> 7. / "mac.logout" / "logs the user out"
[14:40] <Ago-dktj> 8. / "login" / "logs the user in"
[14:40] <Ago-dktj> 9. / "bot.about" / "displays the info the author wants you to see"
[14:40] <Ago-dktj> 10. / "bot.die" / "terminates the bot"
[14:41] <Ago-dktj> 11. / "bot.dns" / "resolves ip/hostname by dns"
[14:41] <Ago-dktj> 12. / "bot.execute" / "makes the bot execute a .exe"
[14:41] <Ago-dktj> 13. / "bot.id" / "displays the id of the current code"
[14:41] <Ago-dktj> 14. / "bot.nick" / "changes the nickname of the bot"
[14:41] <Ago-dktj> 15. / "bot.open" / "opens a file (whatever)"
[14:41] <Ago-dktj> 16. / "bot.remove" / "removes the bot"
[14:41] <Ago-dktj> 17. / "bot.removeallbut" / "removes the bot if id does not match"
[14:41] <Ago-dktj> 18. / "bot.rndnick" / "makes the bot generate a new random nick"
[14:41] <Ago-dktj> 19. / "bot.status" / "gives status"
[14:41] <Ago-dktj> 20. / "bot.sysinfo" / "displays the system info"
[14:41] <Ago-dktj> 21. / "bot.longuptime" / "If uptime > 7 days then bot will respond"
[14:41] <Ago-dktj> 22. / "bot.quit" / "quits the bot"
[14:41] <Ago-dktj> 23. / "bot.flushdns" / "flushes the bots dns cache"
[14:41] <Ago-dktj> 24. / "bot.secure" / "delete shares / disable dcom"
[14:41] <Ago-dktj> 25. / "irc.disconnect" / "disconnects the bot from irc"
[14:41] <Ago-dktj> 26. / "irc.action" / "lets the bot perform an action"
[14:41] <Ago-dktj> 27. / "irc.getedu" / "prints netinfo when the bot is .edu"
[14:41] <Ago-dktj> 28. / "irc.gethost" / "prints netinfo when host matches"
[14:41] <Ago-dktj> 29. / "irc.join" / "makes the bot join a channel"
[14:41] <Ago-dktj> 30. / "irc.mode" / "lets the bot perform a mode change"
[14:41] <Ago-dktj> 31. / "irc.netinfo" / "prints netinfo"
[14:41] <Ago-dktj> 32. / "irc.part" / "makes the bot part a channel"
[14:41] <Ago-dktj> 33. / "irc.privmsg" / "sends a privmsg"
[14:41] <Ago-dktj> 34. / "irc.quit" / "quits the bot"
[14:41] <Ago-dktj> 35. / "irc.raw" / "sends a raw message to the irc server"
[14:41] <Ago-dktj> 36. / "irc.reconnect" / "reconnects to the server"
[14:41] <Ago-dktj> 37. / "irc.server" / "changes the server the bot connects to"
[14:41] <Ago-dktj> 38. / "http.download" / "downloads a file from http"
[14:41] <Ago-dktj> 39. / "http.execute" / "updates the bot from a http url"
[14:41] <Ago-dktj> 40. / "http.update" / "executes a file from a http url"
[14:42] <Ago-dktj> 41. / "http.visit" / "visits an url with a specified referrer"
[14:42] <Ago-dktj> 42. / "ftp.download" / "downloads a file from ftp"
[14:42] <Ago-dktj> 43. / "ftp.execute" / "updates the bot from a ftp url"
[14:42] <Ago-dktj> 44. / "ftp.update" / "executes a file from a ftp url"
[14:42] <Ago-dktj> 45. / "scan.netbios" / "scans weak netbios passwords"
[14:42] <Ago-dktj> 46. / "scan.locator" / "scans for locator exploit"
[14:42] <Ago-dktj> 47. / "scan.dcom" / "scans for dcom exploit"
[14:42] <Ago-dktj> 48. / "scan.dcom2" / "scans for dcom2 exploit"
[14:42] <Ago-dktj> 49. / "scan.webdav" / "scans for iis/webdav exploit"
[14:42] <Ago-dktj> 50. / "scan.stop" / "stops all scans running asap"
[14:42] <Ago-dktj> 51. / "ddos.pingflood" / "starts a Ping flood"
[14:42] <Ago-dktj> 52. / "ddos.udpflood" / "starts an UDP flood"
[14:42] <Ago-dktj> 53. / "ddos.spudpflood" / "starts a spoofed UDP flood"
[14:42] <Ago-dktj> 54. / "ddos.synflood" / "starts a spoofed SYN flood"
[14:42] <Ago-dktj> 55. / "ddos.httpflood"
          / "starts a HTTP flood, can also be used as .visit replacement"
[14:42] <Ago-dktj> 56. / "ddos.stop" / "stops all ddoses running"
[14:42] <Ago-dktj> 57. / "redirect.tcp" / "starts a tcp port redirect"
[14:42] <Ago-dktj> 58. / "redirect.gre" / "starts a gre redirect"
[14:42] <Ago-dktj> 59. / "redirect.http" / "starts a http proxy"
[14:42] <Ago-dktj> 60. / "redirect.socks" / "starts a socks4 proxy"
[14:42] <Ago-dktj> 61. / "redirect.stop" / "stops all redirects running"
[14:42] <Ago-dktj> 62. / "cdkey.get" / "makes the bot get a list of cdkeys"
[14:42] <Ago-dktj> 63. / "rsl.reboot" / "reboots the computer"
[14:42] <Ago-dktj> 64. / "rsl.shutdown" / "shuts the computer down"
[14:42] <Ago-dktj> 65. / "rsl.logoff" / "logs the user off"
```

# IX - Glossaire

Adresse IP : une adresse IP (avec IP pour Internet Protocol) est un numéro permettant l'identification de tout équipement informatique utilisant le protocole TCP/IP. Ces adresses peuvent être permanentes ou attribuées dynamiquement à chaque connexion au réseau par les bureaux d'enregistrement régionaux de l'ICANN.

APWG (Anti Phishing Working Group) : Organisation professionnelle de lutte contre le phishing, regroupant des industriels et des agences gouvernementales.

ASP (Active Server Pages) : langage de programmation développé par Microsoft permettant l'exécution de script sur le serveur, de la même façon que le PHP.

BSD (Berkeley Software Distribution) : famille de systèmes d'exploitation Unix, développés à l'Université de Californie (Berkeley).

*CERT* : les *CERT* (Computer Emergency Response Team) sont des organismes officiels chargés d'assurer des services de prévention des risques et d'assistance aux traitements d'incidents informatiques. Leur fonction première est d'être le point de contact, c'est-à-dire la structure que l'on appelle à l'aide et qui organise les secours en cas d'incident. Ils sont destinés aux entreprises et/ou aux administrations.

CERTA (Centre d'Expertise Gouvernemental de Réponse et de Traitement des Attaques informatiques) : Rattaché à l'Agence Nationale de la Sécurité des Systèmes d'information (ANSSI) au sein du Secrétariat Général de la Défense Nationale (SGDN), le CERTA est chargé d'assister les organismes de l'administration française à mettre en place des moyens de protection et à résoudre les incidents ou les agressions informatiques dont ils sont victimes.

CVV2 (code) : le code CVV2 (Card Verification Value 2) est le code de sécurité à trois chiffres imprimé au dos des cartes de crédit. Il permet de vérifier si la carte est bien en votre possession lors de la transaction.

DIG (commande) : DIG est l'acronyme de "Domaine Information Groper ». Cette commande permet d'envoyer des requêtes uniques à un serveur DNS aux fins de tests ou d'écriture. Elle se comporte comme nslookup mais n'est pas interactive.

DNS : le Domain Name System (en français : système de noms de domaine) est un service permettant d'établir la correspondance entre une adresse IP et le nom de domaine qui lui est associée.

*Double-flux* : ce terme se réfère à l'un des mécanismes présenté plus largement sous l'appellation *fast-flux*. Par rapport au *single-flux*, il fonctionne avec un second niveau d'indexation permettant de sélectionner le serveur assurant la résolution à un instant donné parmi plusieurs serveurs, en mettant ici encore à profit la possibilité de positionner une durée de vie très courte sur les enregistrements référençant les serveurs en charge de la résolution des noms de domaine.

*Fast-flux* : procédé permettant d'associer à un nom de domaine fixe une succession d'adresse IP selon un ordonnancement plus ou moins aléatoire et pour une durée d'association la plus courte possible. Ceci passe par le biais du positionnement du paramètre dit Time-To-Live (durée de validité de l'information délivrée par le serveur responsable du domaine) à une valeur de l'ordre de la minute.

HTML (Hypertext Markup Language) : format de données conçu pour représenter les pages web. Langage de balisage permettant d'écrire de l'hypertexte.

HTTP (HyperText Transfer Protocol) : connu en français littéral sous le nom de « protocole de transfert hypertexte », HTTP est un protocole de communication client-serveur développé pour le World Wide Web.

ICANN (Internet Corporation for Assigned Names and Numbers) : organisme privé de droit américain, à but non lucratif, responsable de la gestion des ressources communes de l'internet (attribution des adresses IP et des noms de domaines).

IDS/IPS (Intrusion Detection System / Intrusion Protection System) : équipement de détection ou de prévention d'intrusion.

IRC : abréviation de Internet Relay Chat (en français, « discussion relayée par Internet »). Protocole de communication sur Internet dédié à la communication instantanée. C'est un prédécesseur de la messagerie instantanée.

MySQL : gestionnaire de base de données SQL (Structured Query Language) Open Source. Il est très utilisé dans les projets libres et dans le milieu industriel.

Nslookup: programme informatique d'interrogation des serveurs DNS (Domain Name System) pour obtenir, soit l'adresse IP en fonction d'un nom de domaine, soit l'inverse.

P2P (Peer to Peer, Pair à Pair) : modèle d'organisation dans lequel il n'existe pas de serveur central, les utilisateurs étant interconnectés entre eux. Chaque nœud de réseau peut agir indifféremment en tant que client ou serveur.

PHP : applications utilisant le langage PHP (Hypertext PreProcessor), langage de programmation interprété conçu pour le développement d'applications web interactives et dynamiques.

Phishing: appelé hameçonnage en français, le phishing est une forme d'attaque informatique reposant sur l'ingénierie sociale et utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de réaliser une usurpation d'identité. La technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance — banque, administration, etc. — afin de lui soutirer des renseignements personnels.

Renater : le réseau RENATER (Réseau National de télécommunications pour la Technologie l'Enseignement et la Recherche) fédère les infrastructures françaises de télécommunication pour la recherche et l'éducation en fournissant une connectivité nationale et internationale.

Rootkit: programme permettant de rendre totalement furtif un autre programme en les rendant (lui et son rootkit) invisibles à un outil de sécurité tel qu'un anti-virus.

Sandbox (en français : bac à sable) : espace mémoire protégé réservé à l'exécution et à l'analyse de programmes douteux, d'où ils ne peuvent interagir avec l'extérieur.

*Single-flux* : ce terme se réfère à l'un des mécanismes présenté plus largement sous l'appellation *fast-flux*. Il a pour principale faiblesse de reposer sur un unique serveur de nom dont l'adresse IP est, elle, statique.

SSL (Secure Sockets Layers) : procédé de sécurisation des transactions effectuées via Internet reposant sur un procédé de cryptographie par clef publique. Il est maintenant normalisé sous la dénomination TLS.

TCP/IP (Transmission Control Program / Internet Protocol) : ensemble de protocoles Internet, conçu pour la transmission de données sur le réseau Internet. Il permet l'interconnexion de

réseaux hétérogènes (c'est à dire d'architectures différentes) et offre des services d'accès à distance, de transfert de fichiers, de courrier électronique.

TLS (Transport Layer Security) : norme de sécurisation par chiffrement du transport de l'information au sein des réseaux informatiques (anciennement SSL).

URL : acronyme de « Uniform Resource Locator ». Syntaxe utilisée sur Internet pour désigner la localisation d'un service ou d'un fichier. Dans le grand public, on dit abusivement « adresse Internet » à la place de URL.


L'ESPRIT DE L'ÉCHANGE

# CLUB DE LA SÉCURITÉ DE L'INFORMATION FRANÇAIS

30, rue Pierre Sémard
75009 Paris
✆ 01 53 25 08 80
clusif@clusif.asso.fr

*Téléchargez les productions du CLUSIF sur*

**www.clusif.asso.fr**