Le Honeynet Project en 2010

Sebastien Tricaud
sebastien(@)honeynet.org

Honeynet Project CTO

Résumé Nous sommes en 2010 après J.C. ; une bande d'irréductibles individus curieux résiste encore et toujours à l'envahisseur. Le slogan du projet honeynet est simple : "Améliorer la sécurité d'Internet sans coût pécunier pour le public". Nous atteignons notre objectif dans la recherche et le développement de nouveaux outils et en fournissant des informations sur les menaces de sécurité observées afin que les individus et les organisations puissent augmenter leur sécurité. Le but de cet article est de vous présenter nos activités publiques afin que vous puissiez à votre tour en bénéficier.

"Comment pouvons-nous nous défendre contre un ennemi quand nous ne savons pas qui il est vraiment ?"

1 The Honeynet Project

Le projet Honeynet est une organisation à but non lucratif (501c3, registre de la ville de Chicago) qui a été fondée par Lance Spitzner en 1999. L'organisation fonctionne autour d'un panel de directeurs (décisions financières, techniques, stratégiques, de recherche, etc) et de chapitres répartis sur la planète terre (Figure 1). Financée par des sponsors [1], remplis d'expériences et de compétences variées (partant de la sociologie en allant au reverse engineering, en passant par la détection d'intrusion ou la recherche pure). Point important : **nous n'avons rien à vendre !**

En pratique, il s'agit de capturer la réalité des attaques diffusées principalement sur Internet. Pour se faire, nous déployons des réseaux et machines dans le but d'être piratés. Par exemple, mettre en place une plage réseau inutilisée (nommé **darknet**) que l'on surveille est un bon moyen de vérifier que nous sommes confrontés à une attaque ciblée, ou non.

1. Si vous êtes intéressé, merci de vous faire connaître. Contactez- moi !

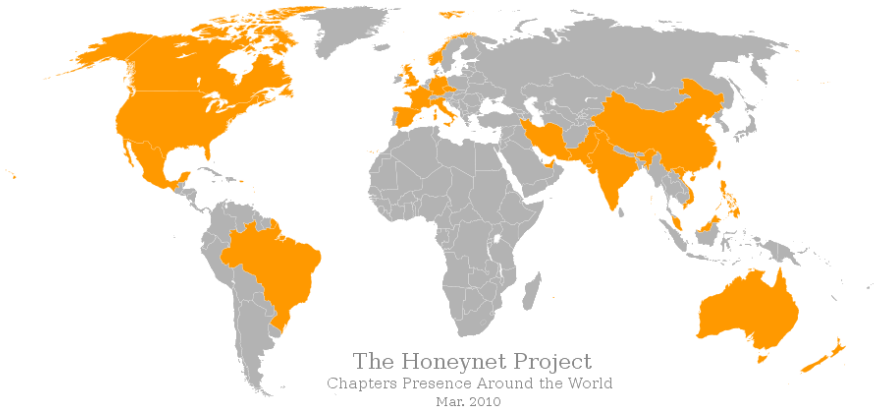

FIGURE 1. Carte du monde des chapitres

Grâce à ce que nous faisons, nous arrivons petit à petit à appréhender les outils, les tactiques et les motivations impliquées par les attaques ciblant des machines et des réseaux. Nous pouvons dès lors partager les leçons apprises. Nous pouvons aussi dégager des tendances et des évolutions.

Pour atteindre notre but1, nous mettons en ligne des documents axés sur trois points :

- **sensibilisation** : sensibiliser autour des menaces qui existent ;
- **information** : pour ceux qui sont sensibilisés, enseigner et informer sur les menaces ;
- **recherche** : donner la possibilité aux organisations d'apprendre plus par elles-mêmes.

Concrètement, cela se passe sous la forme de publication d'outils, de papiers *KYE* (Know Your Enemy), de papiers *KYT* (Know Your Tools) et l'échange de nos travaux est régulièrement

publié dans diverses conférences de sécurité informatique. En ce qui concerne la menace de cybercriminalité, nous observons plusieurs milliers de scans par jour avec de sympathiques statistiques telles que : le honeypot compromis le plus rapidement se fait en 15 minutes et cela descend à moins de 60 secondes pour un ver. Dans la section 3.2, quelques logs récents permettent de vous donner un bon avis de la réalité de propagation des virus ainsi que leur détection.

En ce qui concerne les systèmes d'exploitation, un systéme windows vulnérable est attaqué en moins de trois heures alors qu'un systéme Linux n'est attaqué que sous trois mois. Il n'y a pas de conclusion a tirer ici sur la robustesse d'un systéme d'exploitation donné : l'un est largement plus utilisé que l'autre. La plupart des attaques ciblent les systèmes windows, ipso facto, leurs utilisateurs.

Une menace réaliste est la propagation de vers pouvant travailler de maniére collaborative pour lancer une attaque sur commande, offrant à l'offenseur plusieurs milliers de machines (plusieurs millions pour les dernier botnet à la mode). Ces botnets deviennent de plus en plus complexes. Ils sont conçus par des personnes qui maîtrisent le fonctionnement d'Internet dans son ensemble. Par exemple, les fast-flux [2] permettent une délocalisation aisée et rendent extrémement difficile la capture de la source.

Si nous regardons les motivations, elles restent assez simple : l'*argent*. Tout d'abord, nous observons que les motifs se sont déplacés de pirates isolés vers de vrais criminels organisés et très compétents.

La cible observée s'articule clairement autour des utilisateurs dits grand public ; ils n'ont aucune connaissance en matière de sécurité, pensent que cela ne les concerne pas (bien qu'ils aillent sur le site de leur banque en ligne...), ne savent pas choisir un mot de passe cohérent et sont par conséquent des cibles faciles.

La tendance est maintenant claire : les serveurs sont de moins en moins attaqués en frontal au profit des postes clients qui sont déjà sur le réseau.

Enfin, afin de comprendre ces menaces, nous avons développé une grande série d'outils que l'on peut classifier de cette manière :

- honeypot serveur ou client de **basse interaction** : émulation des applications ou du système déxploitation ;
- honeypot serveur ou client de **haute interaction** : véritables applications et systèmes déxploitation.

Les honeypots de **basse interaction** ont un risque et une complexité limitée pour leur mise en œuvre, au contraire des honeypots de **haute interaction** qui demandent beaucoup plus de temps et

2. Un fast-flux se détecte en remarquant une valeur très faible de TTL pour les renouvellements DNS

peuvent être réellement dangereux (imaginez les dégâts pouvant résulter d'une intrusion sur votre réseau !).

2 Construisons un honeypot à la main

Une manière aisée pour comprendre ce qui arrive sur votre réseau consiste à logger les accès que vous refusez. Il devient intéressant de logger les services qui sont importants pour votre infrastructure, tel que le protocole UDP et le port 5060, si vous opérez de la voix sur IP en utilisant le protocole SIP. Nous mettons donc sur le réseau un serveur qui ne fait rien d'autre que de logger toute requête arrivant sur ce port :

```
winiepot:~# iptables -I INPUT -p udp --dport 5060 -j LOG
```

Lorsqu'un individu tape sur ce protocole/port, cela nous donne la ligne suivante :

```
# tail -n1 /var/log/messages
Apr  1 23:16:22 winiepot kernel: [28260303.311652] IN=eth0 OUT=
     \
MAC=00:1c:c0:27:7f:fb:00:1d:45:bd:9c:7f:08:00 SRC=192.168.0.23
    \
DST=192.168.0.78 LEN=35 TOS=0x00 PREC=0x00 TTL=58 ID=2573 DF \
PROTO=UDP SPT=38283 DPT=5060 LEN=381
```

Rien qu'en partageant déjà ces informations avec un groupe d'individus qui font la même chose que vous, vous voilà avec un début de compréhension de qui vous enquiquine.

Malgré cela, nous ne pouvons pas dire grand chose si nous ne voyons pas le message, il vous faut récupérer le message. Vous avez plusieurs possibilités, entre-autres passer par ulogd2 ou bien en utilisant DaemonLogger. Une fois que vous l'avez fait, vous pouvez regarder ce que l'attaquant cherche à faire **exactement**. Dans notre cas, nous voyons :

```
INVITE sip:test@wallinfire.net SIP/2.0
Via: SIP/2.0/UDP evil.onfastflux.com:5060;branch=
    z9hG4bKOvWWTphl;
To: "Test" <sip:test@wallinfire.net>
From: "Alice" <sip:alice@onfastflux.com>;tag=AAmhnpRM
Call-ID: esUcrsXK@wallinfire.net
CSeq: 1 INVITE
Contact: <sip:alice@onfastflux.com>
```

```
Max-Forwards: 68
Content-Type: application/sdp
Content-Length: 0
```

La constante répétition du message nous indique qu'il y a une tentative de découverte de mot de passe. Mais nous ne pouvons pas en être sûr car nous ne répondons pas au message tel qu'il faut le faire à la sauce SIP [3]. Il faut donc décoder le message SIP et se faire passer pour un serveur SIP pour voir ce que l'attaquant cherche à faire.

Pour plus d'informations sur les attaques ciblant SIP, vous pouvez vous référer au blog de Sjur [4] ainsi qu'à l'application *artemisa* [5].

3 Rapide tour de quelques projets

3.1 LibEMU

Le projet LibEMU [6] fournit des fonctions d'émulation processeur x86 et de détection de shellcode en utilisant les heuristiques GetPC.

La LibEMU supporte la lecture de code binaire x86, l'émulation des registres, une émulation basique de FPU... Il est aisé de se placer sur l'emulateur pour voir ce qui se passe. Voici ce que peut donner le graphe d'appels du ver SQL Slammer (Figure 2).

3. Ahah, et bon courage pour la lecture, la RFC 3261 est longue et il faut ajouter plusieurs dizaines d'extensions
4. http://www.usken.no
5. http://artemisa.sourceforge.net
6. http://libemu.carnivore.it

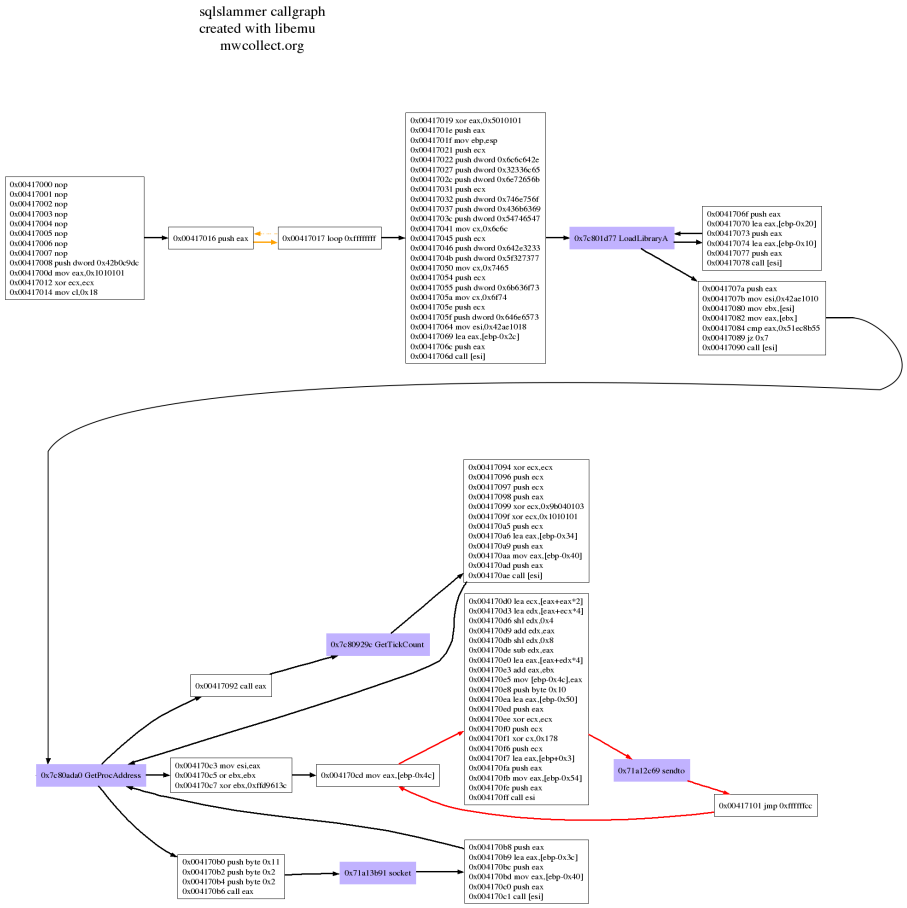

FIGURE 2. Graphe d'appels du ver SQL Slammer effectué par libemu

3.2 Nepenthes

Nepenthes est un honeypot de basse interaction qui émule des vulnérabilités connues pour collecter des informations sur des menaces potentielles. En déployant Nepenthes il ne faudra pas longtemps pour commencer à voir des virus arriver. En regardant les cinq premières lignes du fichier */var/log/nepenthes/logged_submissions*, nous pouvons observer :

```
[2010-01-01T00:10:06] 88.173.53.163 -> 192.168.0.23 \
  link://88.173.53.163:3737/MPe2+A== 725
       c1f3ef623cbd811a9acc6c40ad07c
[2010-01-01T00:12:56] 88.185.87.220 -> 192.168.0.23 \
  link://88.185.87.220:46509/D2oeOQ== 954
       a98c971fda498f9d1211f18e75cd7
[2010-01-01T00:24:36] 88.83.48.36 -> 192.168.0.23 \
  link://88.83.48.36:35368/+BmAdg==
       be36334377890a52b56c9023de688fe7
[2010-01-01T01:19:50] 88.181.154.124 -> 192.168.0.23 \
  link://88.181.154.124:15998/+Hgf+A== 9
       c9c6929c0ce56356aac1ca45b724213
[2010-01-01T01:56:35] 88.83.34.82 -> 192.168.0.23 \
  link://88.83.34.82:41267/+Lcq+A== 56048779
       db776914d5d11939b30a8331
```

Les fichiers sont stockés sous la forme de condensats MD5 et envoés par défaut dans le répertoire */var/lib/nepenthes/binaries/*. Une fois stockés, on peut lancer un **clamscan** ou utiliser **virustotal** afin de déterminer le potentiel de détection d'un virus.

Si nous prenons la journée du premier avril 2010, nous avons reçus 2211 binaires, qui fûrent 597 binaires uniques (nous ne stockons qu'un seul condensat MD5). Au final, comme nous l'observons dans la figure 3, une liste non-négligeable de virus (32) ne sont pas détectés par ClamAV.

Avec les malwares capturés, il vous est possible de les classifier pour ensuite les étudier en fonction de votre sujet de prédilection.

3.3 Dionaea

Dionaea [7] est le successeur de Nepenthes, il embarque python comme langage de script, utilise libemu pour détecter les shellcodes. Il supporte IPv6 et TLS. Il vient récemment de se doter d'un système de soumissions XMPP. Si Dionaea et est le successeur naturel de

7. http://dionaea.carnivore.it/

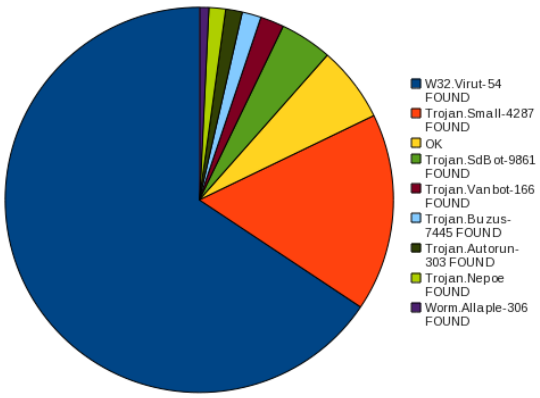

FIGURE 3. Top dix des virus récupérés le 1er Avril 2010

Nepenthes, c'est tout simplement car il est fait par le même auteur (l'excellent *Markus Koetter*).

3.4 PhoneyC

PhoneyC [8] est un honeypot client écrit en Python qui va chercher une URL, puis va :

- comprendre les tags HTML pour les liens distants ;
- exécuter tout code javascript trouvé à travers spidermonkey [9] ou script visual basic (avec vb2py) et l'exécuter avec libemu (y compris celui trouvé dans les PDF) afin de capturer les malwares qu'un poste client peut récupérer en naviguant sur la toile ;
- utiliser son module de détection ActiveX ;
- supporter différentes méthodes de détection (modules de vulnérabilités, ClamAV...)

3.5 Glastopf

Considérant le fait que les attaques sur les applications web concernent environ 60% des attaques totales, Glastopf est né comme

8. http://code.google.com/p/phoneyc/
9. http://www.mozilla.org/js/spidermonkey/

honeypot émulant plusieurs milliers de vulnérabilités orientées web. Le principe est trivial : renvoyer une réponse correcte à l'attaquant qui cherche à tirer profit d'une application particulière. Il est possible de télécharger ce projet à l'adresse suivante : http://glastopf.org/. Comme beaucoup de projets que nous avons, Glastopf est aussi écrit en Python.

4 Les challenges

Nous avons récemment renoué avec notre tradition d'organisation des challenges [10]. Il s'agit de mettre à disposition un cas d'attaque et de demander au public des réponses qui sont ensuite notées dans le but de dégager un vainqueur.

Les trois premiers nouveaux challenges ont porté sur :

1. attaque réseau contenue dans un pcap ;
2. attaque de navigateurs ;
3. attaque à travers un PDF malicieux.

Les challenges restent un terrain de jeu sur lequel il est possible de faire ses armes à partir de données réelles. Nous essayons de coller au maximum avec des attaques le plus en phase avec les attaques du moment. C'est une véritable source de stimulation que de préparer ces challenges, appréhender la manière dont certaines personnes résolvent un problème, ou encore écrire les outils pour automatiser la détection de certains types d'attaques.

5 Conclusion

Nous essayons, à travers le Honeynet Project, d'être le plus efficace possible dans la compréhension de la réalité des problèmes de sécurité. Le fait d'être une organisation internationale nous permet d'avoir une vue à l'échelle globale afin d'améliorer notre compréhension de l'attaque et de la défense.

Nous sommes toujours surpris par l'inventivité des attaquants, il est parfois relativement complexe de comprendre plusieurs bouts d'une attaque. C'est en mélangeant différentes spécialités que cela

10. http://www.honeynet.org/challenges

peut nous éclairer sur la façon dont nous pouvons percevoir un problème.

Peut-être que demain nous ne serons plus aussi efficaces qu'aujourd'hui face aux attaques à venir, mais nous continuons à nous battre et recruter des gens talentueux afin de déncourager les attaquants au maximum. Nous faisons en sorte de publier le maximum, ceci afin que n'importe qui, n'importe quelle organisation puissent profiter de notre travail, non pas dans le but de reconnaissance, mais pour pouvoir se défendre sans rien devoir à qui que ce soit.

6 Remerciements

Je tiens à remercier le projet Honeynet dans son ensemble, il n'y a personne à nommer car il s'agit d'un véritable travail collectif. Je tiens aussi à remercier Jean-Philippe Gaulier du comité du SSTIC pour sa relecture minutieuse.