Le programme HACIENDA

Julian Kirsch, Christian Grothoff, Monika Ermert, Jacob Appelbaum, Laura Poitras, Henrik Moltke, Andreas Enge

Le programme HACIENDA

Julian Kirsch, Christian Grothoff, Monika Ermert,
Jacob Appelbaum, Laura Poitras, Henrik Moltke
Traduction française*: Andreas Enge

19 août 2014

1 Introduction

Depuis les débuts du protocole TCP, des scans de ports ont été utilisés par des hackers pour localiser des systèmes vulnérables. Des documents classifiés top secrets vus par Heise (portail allemand d'actualités technologiques) révèlent qu'en 2009, l'agence d'espionnage britannique GHCQ a transformé les scans de ports en un outil utilisé par défaut contre des pays entiers (figure 1). 27 pays sont mentionnés comme cibles du programme HACIENDA dans la présentation (figure 2), qui vient accompagnée d'une offre promotionnelle : des lecteurs souhaitant faire de la reconnaissance contre un autre pays sont invités à simplement envoyer un e-mail (figure 3). Les documents ne détaillent pas de procédure de vérification ou le besoin de justifier une telle action. La possibilité de scanner des pays entiers n'est pas une chimère folle ; en 2013 un scanner de port appelé Zmap a été développé qui arrive à parcourir l'espace mondial des adresses IPv4 en moins d'une heure depuis un simple ordinateur personnel [2]. Ainsi, l'utilisation massive de cette technologie peut transformer tout ordinateur, grand ou petit, partout dans le monde, en une cible de saboteurs informatiques criminels au service des états.

La liste des services ciblés inclue des services publics omniprésents tels que HTTP et FTP, ainsi que des protocoles d'administration usuels tels que SSH (Secure SHell protocol, utilisé pour accéder à un système distant) ou SNMP (Simple Network Management Protocol, utilisé pour l'administration de réseaux), voir figure 4. Sachant que des scanners de ports comme Zmap publiés entretemps permettent à n'importe qui de faire des scans massifs, ce n'est pas la technologie employée qui choque, mais plutôt l'échelle gargantuesque et l'ubiquité de l'opération.

Le chapitre suivant décrit la fonctionnalité des scanners de ports ainsi que les informations qu'ils permettent d'obtenir, illustrant les possibilités en découlant si un état s'en sert à grande échelle.

*Version légèrement remaniée de la première publication à `http://www.heise.de/ct/artikel/GCHQ-NSA-Le-programme-HACIENDA-2293122.html`

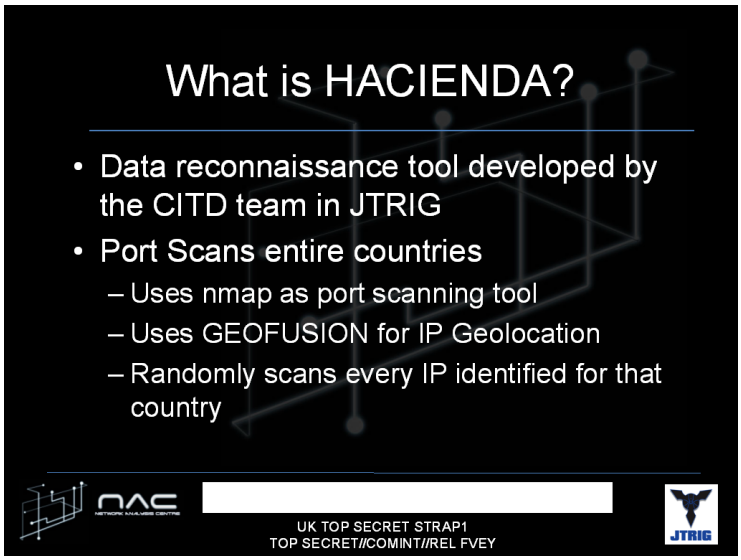


FIGURE 1 –

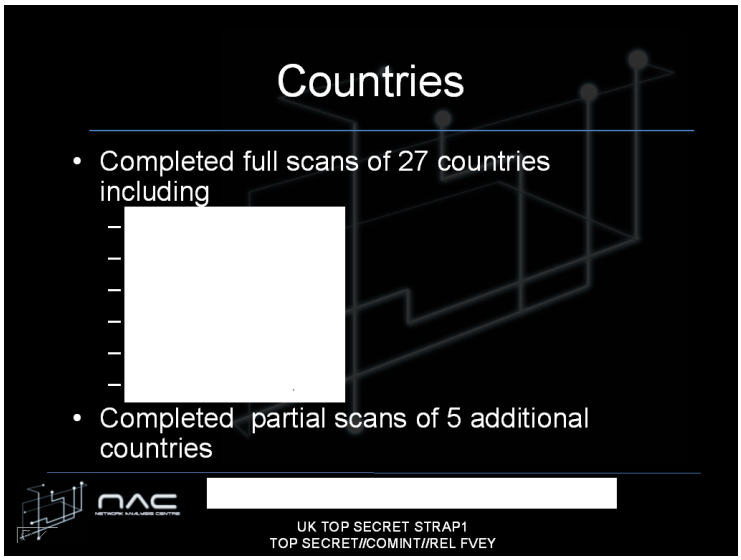


FIGURE 2 –

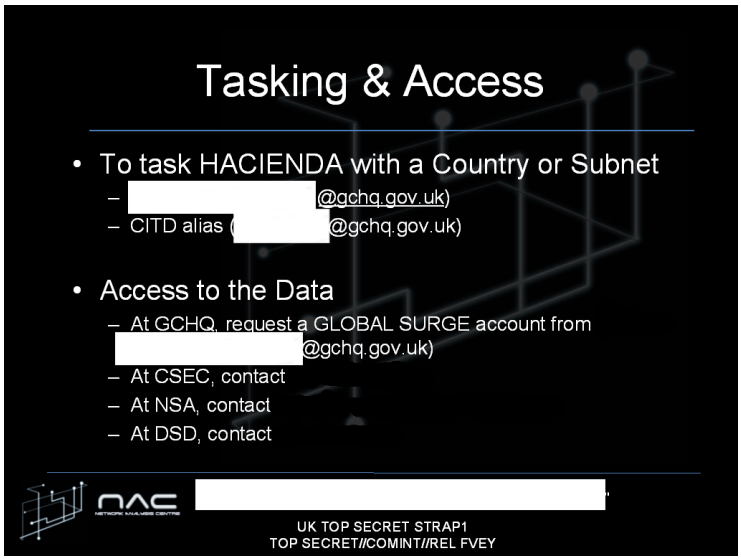


FIGURE 3 –

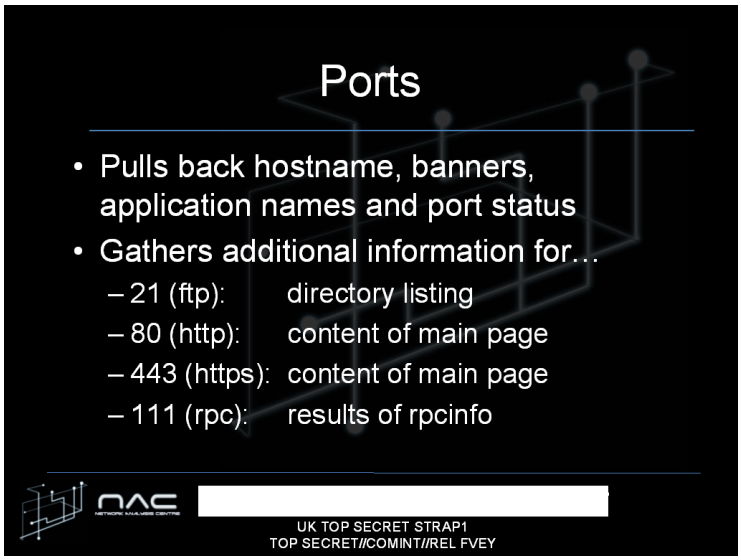


FIGURE 4 –

2 Contexte : Le Three-way-handshake de TCP

TCP (Transmission Control Protocol, protocole de contrôle de la transmission) est le protocole le plus répandu sur Internet. À chaque fois qu'un e-mail est envoyé ou qu'une page web est regardée, c'est TCP qui est responsable de la transmission fiable des paquets de données entre le client et le serveur. Pour déterminer quels services sont disponibles sur un ordinateur, les scanners de ports exploitent un problème structurel de TCP. Depuis l'aube des temps, ces scans sont utilisés par des attaquants pour localiser des serveurs vulnérables. À chaque fois qu'un client TCP voudrait communiquer avec un server TCP, les deux parties engagent un « tree-way-handshake » (« poignée de mains à trois tours »). D'ailleurs, c'est une erreur dans la conception de ce handshake qui rend le scan de ports possible, car lors du handshake, le serveur laisse fuire de l'information sur la disponibilité d'un service, et ce sans vérifier au préalable si le client est autorisé à l'utiliser.

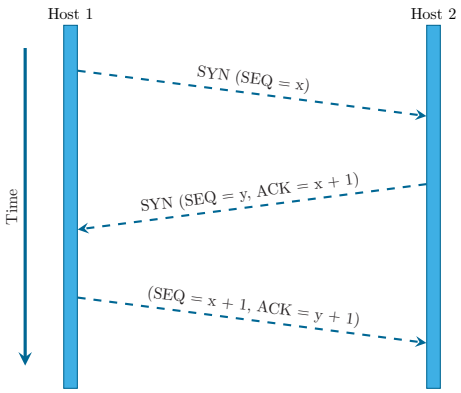


FIGURE 5 – Flux de paquets pour un three-way-handshake avec succès

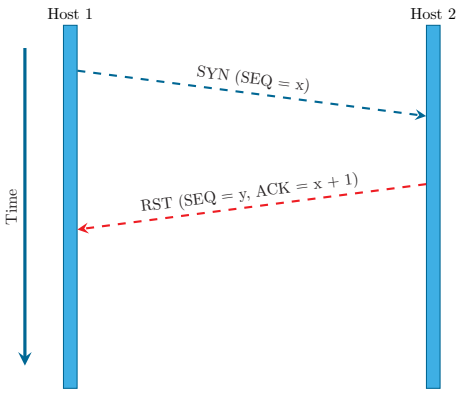


FIGURE 6 – Flux de paquets pour un essai de connexion sur un port fermé

La figure 5 illustre la séquence des paquets TCP envoyés pour établir une connexion : d'abord l'ordinateur qui voudrait ouvrir une connexion envoie un paquet TCP de type SYN (de synchronisation). Le serveur ciblé renvoie un SYN/ACK (synchronisation/accord) s'il accepte la requête. Après réception de cette réponse positive, un autre paquet ACK de la part du client complète le three-way-handshake, et la connexion est établie. C'est précisément cette poignée de main qui permet à un attaquant de savoir si un service TCP est proposé à un port donné par un serveur sur Internet : si le port TCP est fermé, le serveur réagit différemment ; au lieu du paquet TCP de type SYN pour indiquer un port ouvert, il renvoie un paquet RST (« reset », remise à zéro) en réponse au premier paquet SYN (figure 6). Ainsi un adversaire obtient aisément la carte des services réseau disponibles sur Internet en tenant compte des différentes réponses des serveurs dans le flux des paquets selon les figures 5 et 6, respectivement

3 L'ennemi en ligne

Le GCHQ ne se contente pas de simples scans de ports, mais collecte également des « bannières » et d'autres informations facilement obtenables (figure 4). Une bannière est un message qu'une application envoie par défaut à tout client qui se connecte à son port associé ; souvent, elle contient des informations détaillées sur le système et l'application, comme des numéros de version et d'autres informations utiles pour détecter des failles. L'échelle massive de ces opérations de reconnaissance relatées dans les documents montre qu'il s'agit non d'attaquer des cibles précises, mais de collecter activement et de cartographier complètement les systèmes vulnérables dans le monde. Au lieu du cliché d'une écoute passive ubiquitaire, les documents présentés montrent une interaction avec les réseaux et les systèmes.

En préparant des attaques contres des services accessibles par SSH ou SNMP, les agences d'espionnage ciblent des infrastructures critiques telles que les systèmes nécessaires pour le bon fonctionnement des réseaux. Comme illustré par les intrusions dans les systèmes de Belgacom [1] et de Stellar [11], dès que l'ordinateur ou les données personnelles d'un employé deviennent utiles, ces systèmes et personnes sont ciblés et attaqués.

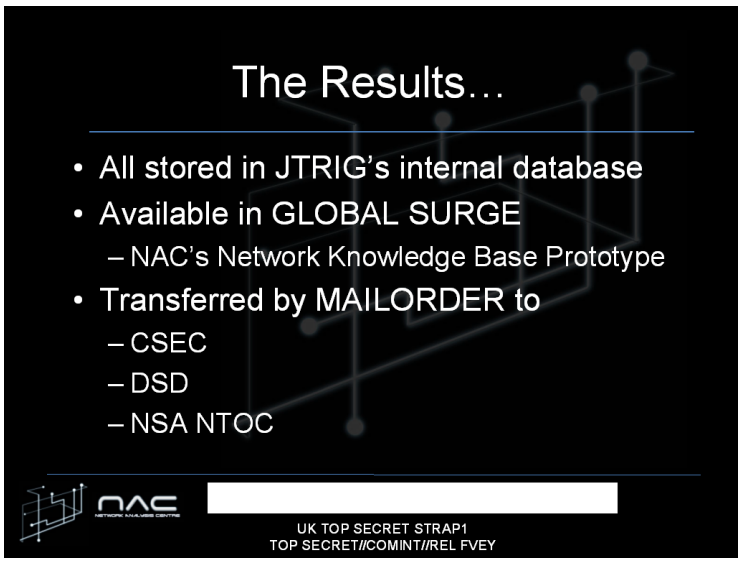


Figure 7 –

La base de données issue des scans est alors diffusé aux autres membres du club des espions des « Five eyes » (figure 7), c'est-à-dire les États unis, le Canada, le Royaume uni, l'Australie et la Nouvelle-Zélande. Le programme MAILORDER est décrit dans les documents comme un protocole d'échange de données sécurisé parmi les agences d'espionnage des Five eyes.

4 Tout est une cible

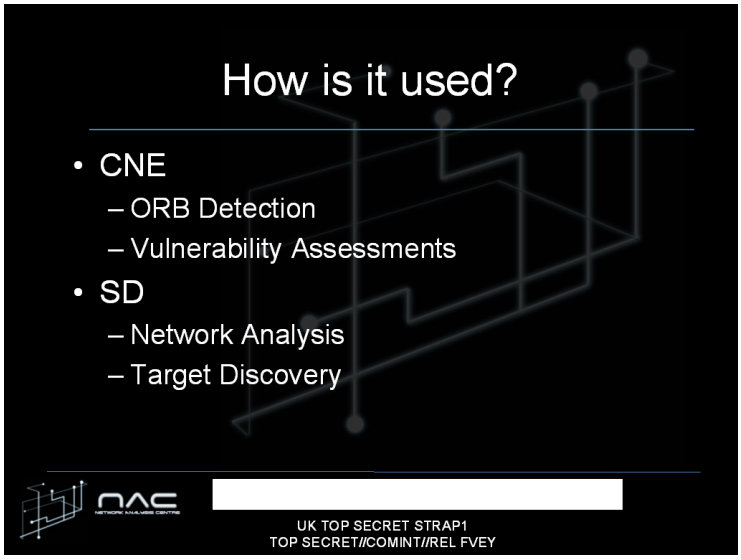


FIGURE 8 – CNE veut dire « Computer Network Exploitation », « Exploitation de réseaux d'ordinateurs »

Le scan de serveurs de pays entiers et la recherche de vulnérabilités dans l'infrastructure des réseaux a pour but suprême la « domination d'Internet » : « Mastering the Internet » est également le nom que le GCHQ a donné à un programme d'écoute de connexions réseau. Ces agences d'espionnages essaient d'attaquer chaque système, probablement dans le but d'obtenir l'accès à d'autres systèmes. Des systèmes peuvent devenir des cibles simplement parce qu'ils peuvent éventuellement aider à créer un chemin vers une cible précieuse pour les espion, même s'il n'y a aucune indication tangible que ce sera jamais le cas. Suivant cette logique, tout appareil est une cible digne de colonisation, car chaque cible exploitée devient théoriquement intéressante comme moyen d'infiltrer, de surveiller ou de se rapprocher de la cible suivante.

Les scans de ports et le téléchargement de bannières pour identifier les logiciels tournant sur le système espionné, ne sont que la première étape d'une attaque (figure 8). Des documents top secrets de CSEC, NSA et GHCQ vus par Heise illustrent que les agences d'espionnage impliquées suivent les mêmes méthodes que le cyber-crime organisé (figure 10) : la reconnaissance (figure 10) est suivie par la compromission (figure 11) et par la prise en main du système (figure 12) ainsi que le vol de données (figure 13). La présentation de la NSA montre que l'agence agit dans l'esprit de criminels. Les transparents discutent des techniques et montrent des captures d'écran de leurs propres outils pour soutenir cette approche criminelle (figures 14, 15 et 16).

FIGURE 9 –

FIGURE 10 –

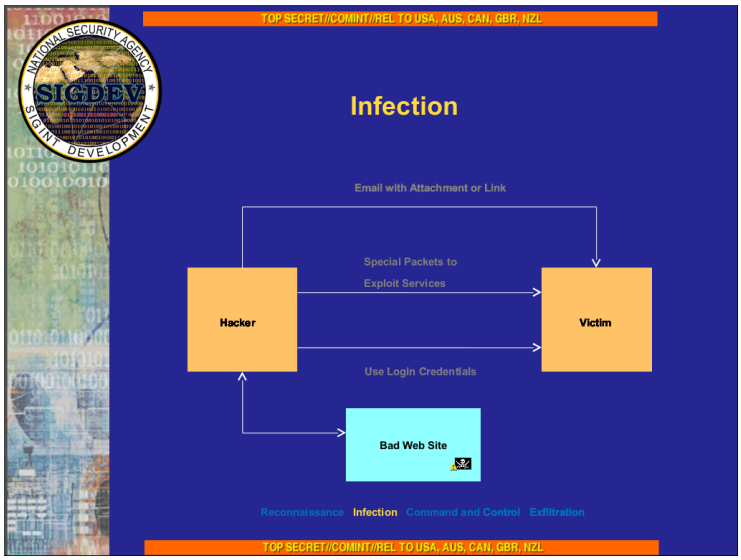


FIGURE 11 –

FIGURE 12 –

FIGURE 13 –

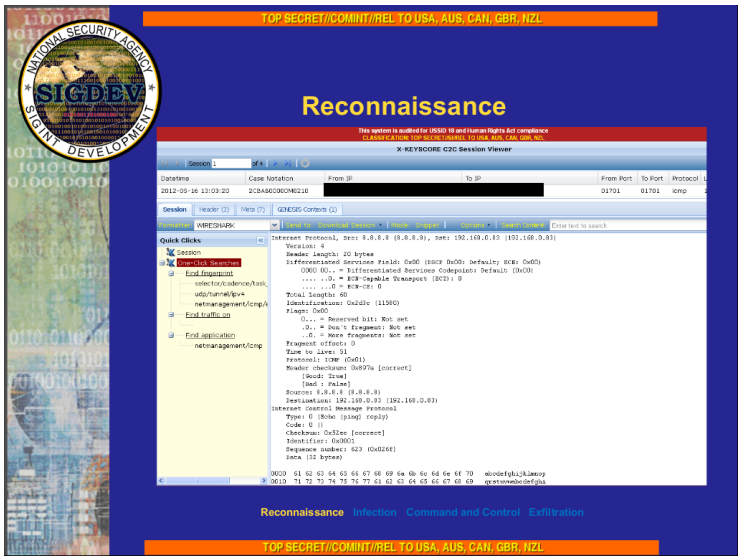


FIGURE 14 –

FIGURE 15 –

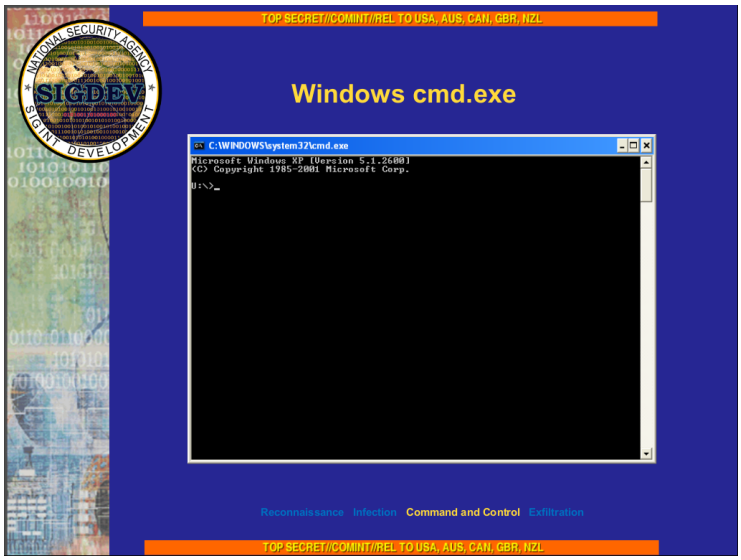


FIGURE 16 –

5 La colonialisation d'Internet

Il est connu que la NSA s'intéresse aux attaques jour zéro (« 0-day »), c'est-à-dire exploitant des vulnérabilités dans des logiciels qui ne sont pas encore connues du public et pour lesquelles il n'y a pas encore de remède sous forme d'un « patch ». Quand un attaquant ayant connaissance d'une telle vulnérabilité jour zéro trouve un service vulnérable tournant sur un serveur, toute résistance est futile. Selon toute vraisemblance, des pare-feux n'offrent pas assez de protection, soit parce que les administrateurs utilisent eux-mêmes un accès à distance pour leur travail, soit parce que les agences d'espionnage ont déjà infiltré le réseau local [3]. Qui plus est, l'intégration de nouveau matériel dans le réseau local, tel qu'un pare-feu configurable à travers SNMP, peut ouvrir de nouvelles brèches.

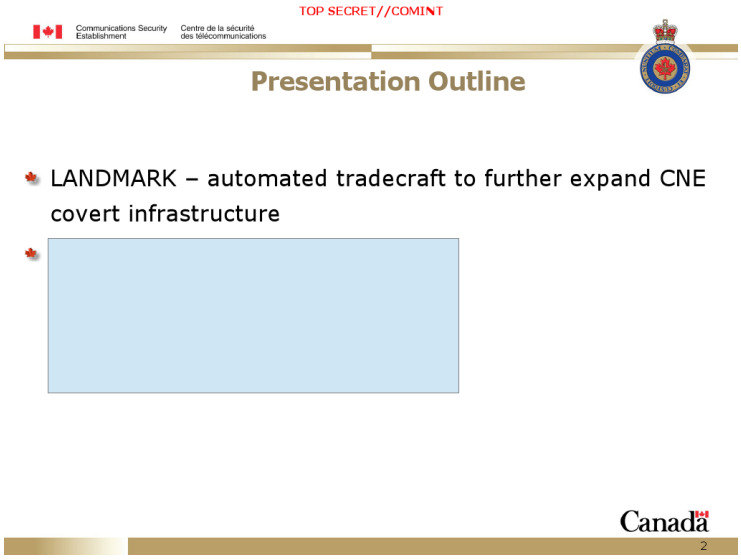


FIGURE 17 –

La figure 8 pointe un rôle particulier qu'HACIENDA joue dans l'infrastructure des Five eyes, notamment pour l'extension de leur infrastructure cachée. Les documents top secrets vus par Heise décrivent le programme LANDMARK mis en œuvre par l'agence d'espionnage canadienne CSEC pour l'expansion de l'infrastructure cachée (figure 17).

L'infrastructure cachée inclue des « Operational Relay Boxes » (ORB, boîtiers de relais opérationnel), utilisés pour dissimuler l'endroit où se trouve l'attaquant quand les Five eyes exploitent des vulnérabilités contre leurs cibles ou volent des données (figure 18). Plusieurs fois dans l'année le club d'espionnage tente ainsi de gagner le contrôle d'autant de machines que possible, tant qu'elles se trouvent à l'étranger. À titre d'exemple, en février 2010, 24 espions ont été capables lors d'une seule journée de travail de localiser plus de 3000 ORB potentiels (figure 19). Néanmoins, l'effort à fournir pour évaluer les résultats d'HACIENDA manuellement était considéré trop grand (figure 20), de sorte que le système OLYMPIA était programmé pour automatiser la procédure (figure 21). Ainsi, les espions se vantent qu'ils peuvent localiser des systèmes vulnérables à l'intérieur d'un sous-réseau en moins de cinq minutes (figure 22).

Mais les canadiens ne sont pas les seuls à chercher avec HACIENDA des machines compromises et susceptibles d'être transformées en ORB. La chasse aux ORB est organisée au GCHQ dans le cadre du programme MUGSHOT (figure 23). La procédure y est également automatisée,

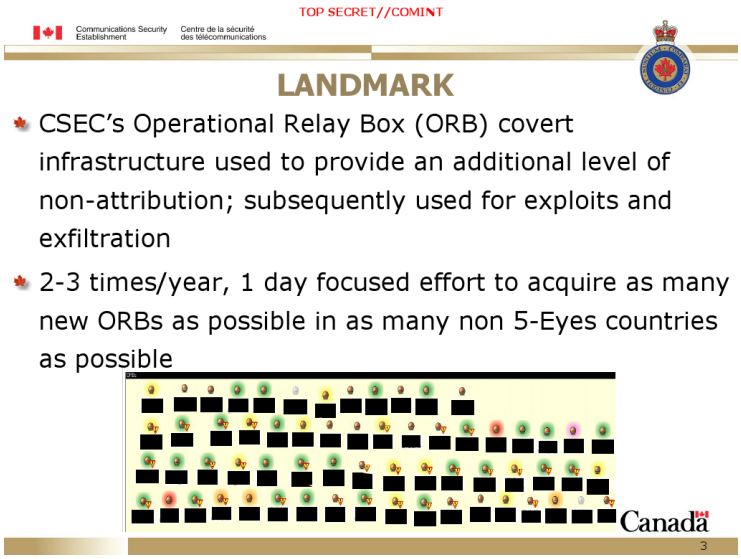


FIGURE 18 –

ce qui selon l'agence a permis une amélioration significative de la précision (figure 24). Encore une fois, les informations fournies par HACIENDA jouent un rôle important (figure 25). Point crucial : à l'aide de MUGSHOT, le GCHQ combine les résultats de scans actifs (HACIENDA) avec la surveillance passive (figure 26), pour « comprendre tout ce qui est important de toutes les machines sur Internet ».

Ainsi, les administateurs système et réseau se trouvent face aux menaces d'espionnage industriel, de sabotage et d'atteintes aux droits de l'homme fabriquées par des adversaires étatiques qui, sans distinction, attaquent l'infrastructure réseau et cambriolent les serveurs. Seule la perspective d'accès suffit à un tel adversaire pour justifier son comportement, et il est soutenu par des budgets de plusieurs milliards de dollars, par l'impunité des agents et par des entreprises privées des pays des Five eyes contraintes de collaborer. Tout administrateur système ou réseau doit se prémunir contre ces menaces inouïes. À cause de ces programmes, les citoyens des pays en dehors des Five eyes en particulier se trouvent face à un niveau considérablement baissé de sécurité, de confidentialité, d'intégrité et de robustesse.

TOP SECRET//COMINT

Communications Security Establishment | Centre de la sécurité des télécommunications

LANDMARK – the recent past....

- February 2010
- Operation encompassing the whole of LONGRUN solely using OLYMPIA (CSEC's network knowledge engine with automated tradecraft)
- 8 teams of 3 network exploitation analysts busy for 5-8 hours
- A list of 3000+ potential ORBs

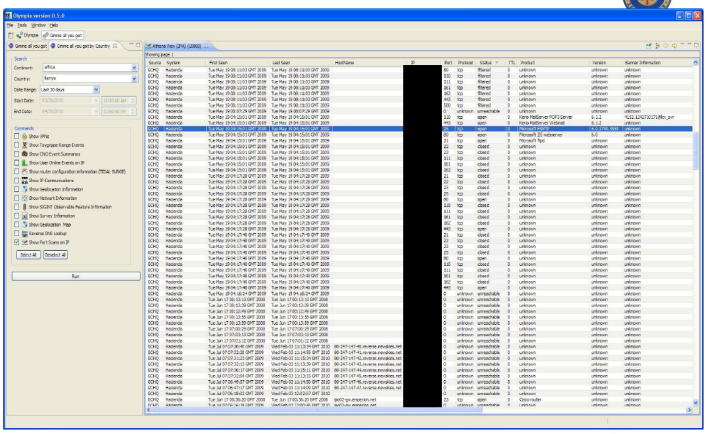
Canada

5

FIGURE 19 –

TOP SECRET//COMINT

Communications Security Establishment | Centre de la sécurité des télécommunications



- BUT, network analysis still manual!

Canada

6

FIGURE 20 –

TOP SECRET//COMINT

Communications Security Establishment | Centre de la sécurité des télécommunications

LANDMARK today…

- Network analysis tradecraft to determine vulnerable devices has been encoded within OLYMPIA

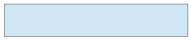
Canada

7

FIGURE 21 –

TOP SECRET//COMINT

Communications Security Establishment | Centre de la sécurité des télécommunications



-  GSM provider
- NSA TAO requested assistance gaining access to the network
- Network analysis using OLYMPIA:
  - DNS query to determine IP address
  - IP address to network range
  - Network range to port scan
  - Are there any vulnerable devices in that range?
- Duration: < 5 minutes

Canada

8

FIGURE 22 –

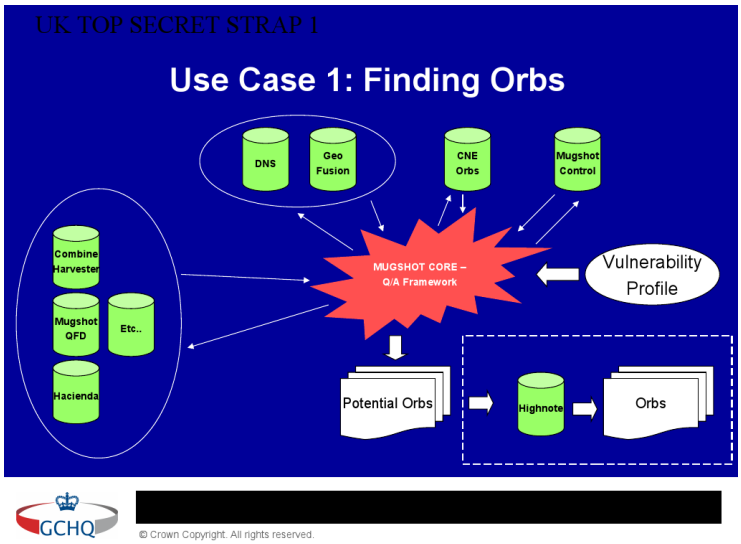


FIGURE 23 –

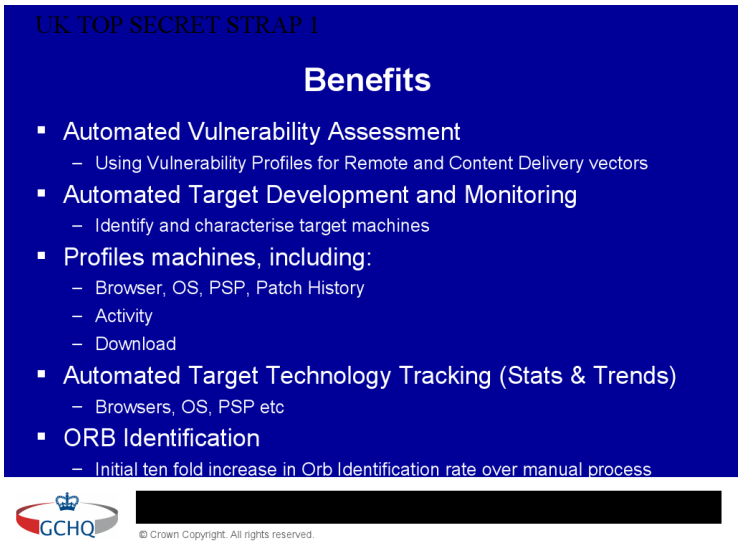


FIGURE 24 –

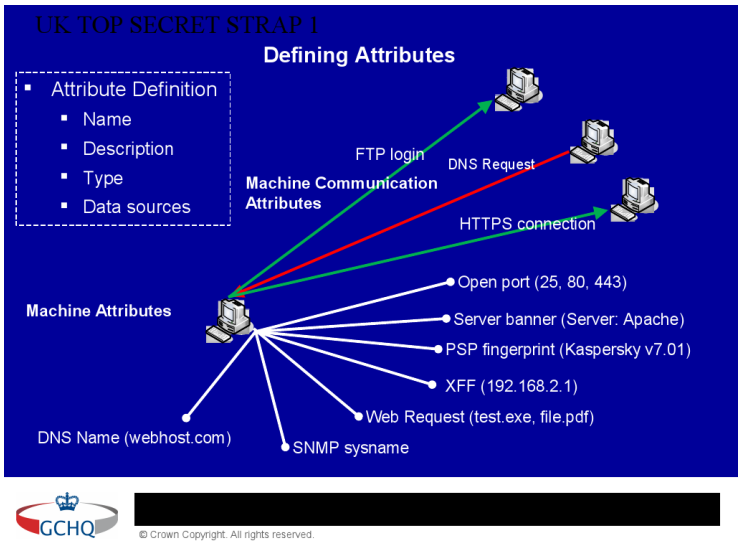


FIGURE 25 –

FIGURE 26 –

6 Résumé

Les services secrets profitent de leur capacité à prendre contrôle des système dans Internet pour augmenter leur pouvoir. Leurs activités suivent le comportement typique de cyber-criminels, qui utilisent des scans de ports pour identifier leurs victimes potentielles. Face à cette menace sérieuse, les administateurs système doivent renforcer leurs défenses et, en particulier, réduire la visibilité de services non publics. Patcher des services ne protège pas contre les attaques jour zéro, et les pare-feux peuvent être impraticables ou insuffisants. Dans la seconde partie de cet article, nous allons introduire une autre option pour des administrateurs système pour réduire la visibilité de services d'administration non publics face aux opérations de reconnaissance. En standardisant de telles techniques, la communauté d'Internet pourra affaiblir la capacité de services de sécurité à reigner sur Internet.

Assommer HACIENDA

Dans cet article, nous allons décrire une nouvelle variante du « port knocking » (« frapper à la porte ») pour contrer un adversaire étatique qui scanne activement ; ainsi, elle offre une certaine protection contre le programme HACIENDA et pourra contribuer à arrêter les espions au niveau de la reconnaissance.

7 Introduction

Tandis qu'il est difficile de se défendre contre des vulnérabilités non publiées dans des services publics, minimiser leur empreinte visible et ainsi leur surface d'attaque est bien plus aisé pour des services d'administration. « Port knocking » [8] est une technique bien connue pour réduire la visibilité de serveurs TCP sur Internet. L'idee de base est de ne pas faire répondre le serveur TCP (dans l'affirmatif) à une requête TCP de type SYN, à moins qu'un paquet particulier « knock » n'ait été reçu au préalable. Ceci peut augmenter la sécurité, puisqu'un attaquant qui ne peut établir de connexion TCP ne peut pas non plus attaquer le serveur TCP lui-même.

Néanmoins, les techniques de port knocking traditionnelles [10] ne prennent pas en compte des états comme adversaire moderne. Précisément, des scans de ports ne sont pas la seule façon comment un attaquant peut apprendre l'existence d'un service ; si le service est accédé à travers un réseau dans lequel l'attaquant est capable de surveiller tout le trafic, l'adversaire peut observer la connexion et en déduire l'existence du service. Un attaquant étatique pourra même être capable d'observer tout le trafic émis du client TCP et monter une attaque dite « man in the middle » (« de la personne interposée ») pour prendre contrôle de la connexion TCP juste après la complétion du handshake TCP initial. Un attaquant avancé qui contrôlerait les routeurs, pourrait également essayer d'identifier des port knocks insuffisamment déguisés à travers l'observation de motifs inhabituels dans le trafic réseau. Néanmoins, on peut toujours supposer qu'un tel adversaire ne déclarerait pas un handshake TCP standard comme suspect, car ceux-ci sont bien trop fréquents.

8 TCP Stealth

TCP Stealth est une ébauche (« draft ») de l'IETF [7] décrivant une variante du port knocking qui est facilement mise en œuvre et cachée. TCP Stealth intègre le jeton d'autorisation dans le TCP ISN (« initial sequence number », un champ du paquet TCP), et permet aux applications d'ajouter des protections pour la partie données du paquet. Ainsi, TCP Stealth est difficile à détecter dans le réseau comme le trafic est indistinguable d'un three-way-handshake ordinaire, et des attaques par interposition (« man in the middle) » ou de rejeu sont rendues plus difficiles par la protection au niveau des données. TCP Stealth est compatible avec IPv4 et IPv6.

TCP est utile pour tout service s'adressant à un groupe d'utilisateurs suffisamment petit pour rendre le partage d'un mot de passe avec tous ses membres praticables. On peut citer, par exemple, l'administration de serveurs par accès SSH ou FTP, des ponts Tor, des serveurs personnels pour POP3/IMAP(S), et des réseaux superposés pair-à-pair de type ami-à-ami. TCP Stealth s'utilise le plus facilement par intégration dans le système d'exploitation.

TCP Stealth est disponible pour les systèmes au noyau Linux par le patch Knock [6]. Pour des noyaux contenant ce patch, TCP Stealth peut être utilisé par des applications via un simple appel à `setsockopt()`, ou en préchargeant la bibliothèque partagée `libknockify` et en posant les variables d'environnement idoines.

9 Installation

Comme le noyau Linux stable ne contient pas encore Knock, le noyau de la machine censé l'utiliser doit être patché, ce qui se fait aisément comme suit :

1. Premièrement, obtenez les sources de la version du noyau recherchée de `https://www.kernel.org` si vous voulez utiliser un noyau standard non modifié. Notez que nombre de distributions adaptent le noyau et proposent les sources du noyau modifiées, qu'on pourra vouloir utiliser.
2. Une fois les sources du noyau disponibles, téléchargez le patch Knock correspondant depuis `https://gnunet.org/knock`. Notez que si vous voulez faire tourner une version du noyau non explicitement listée sur la page web de Knock, il est recommandé d'essayer les patchs de la version la plus proche qui soit disponible.
3. Changez dans le répertoire où résident les sources du noyau (concernant le nom du répertoire, remplacez la partie `<your-version>` selon votre choix de version du noyau et des patchs) et appliquez les patchs (pour plus d'informations sur comment appliquer et enlever des patchs aux sources du noyau, consultez les archives [5] de kernel.org) :

```
~$ cd linux-<your-version>/
~/linux $ patch -p1 < /path/to/knock/patch/tcp_stealth_<your-version>.diff
```

4. Récupérez la configuration du noyau qui tourne actuellement. Il y a deux méthodes usuelles, l'une ou l'autre desquelles peut être utilisée sans préférence :
   (a) Des distributions dérivées de Debian gardent une copie des paramètres de configuration du noyau dans le répertoire `/boot`. Vous pouvez la recopier vers les sources actuelles du noyau comme suit :

```
~/linux $ cp /boot/config-$(uname -r) .config
```

   (b) De nombreuses autres distributions compilent le noyau avec la possibilité de lire la configuration du noyau qui tourne depuis le système de fichiers `/proc/` :

```
~/linux $ zcat /proc/config.gz > .config
```

   (c) Si ces deux méthodes échouent, vous pouvez essayer d'utiliser la configuration du noyau par défaut en entrant

```
~/linux $ make defconfig
```

   Néanmoins, ne vous attendez pas alors à un noyau convaincant d'un point de vue de performance et de stabilité.
5. Choisissez les défauts pour tous les paramètres de configuration qui ne sont pas dans votre configuration actuelle. Une version différente du noyau pourrait introduire de nouvelles options de configuration :

```
~/linux $ yes "" | make oldconfig
```

6. Activez Knock dans votre configuration en sélectionnant dans le menu interactif : `Networking Support > Networking Options > TCP/IP networking > TCP: Stealth TCP socket support`.

```
~/linux $ make menuconfig
```

7. Le noyau est maintenant prêt à être compilé. Entrez

```
~/linux $ make bzImage && make modules
```

pour compiler le noyau et tous les modules supplémentaires. Ce pas peut prendre longtemps. Si vous avez une machine avec plusieurs cœurs, vous pouvez ajuster le nombre de threads de compilation à l'aide de l'option `-j` aux deux commandes `make`.

8. Si la compilation finit correctement, installez le nouveau noyau et tous ses modules. Puis, créez automatiquement un nouveau initramdisk pour le noyau que vous venez de compiler. Si vous avez installé sudo, entrez

```
~/linux $ sudo make modules_install && sudo make install
```

Sinon, entrez ces mêmes commandes (sans les `sudo`) dans une shell root.

9. Rebootez la machine et configurez le boot loader pour choisir le nouveau noyau. Vous disposez maintenant d'une machine avec Knock !

10 Activer Knock avec `LD_PRELOAD`

Knock peut être utilisé sans modifier le code source d'un logiciel. Ceci peut s'avérer utile quand le code source n'est pas disponible ou quand l'insertion des appels nécessaires à la libc n'est pas possible (par exemple à cause de restrictions imposées par la logique de l'application).

Pour utiliser Knock dans des applications existantes, une bibliothèque dynamique libknockify est proposée. L'usage de base de l'objet partagé libknockify qui active Knock pour le logiciel `example_program` est comme suit :

```
$ KNOCK_SECRET="shared secret"
$ KNOCK_INTLEN=42
$ LD_PRELOAD=./libknockify.so
$ ./example_program
```

Dans la suite, quand l'application `example_program` communique à travers TCP, libknockify applique les options du socket respectif pour activer l'utilisation de Knock dans le noyau. Dans l'exemple, le secret partagé est dérivé du texte « shared secret », et la protection de l'intégrité du contenu est limitée aux 42 premiers octets des données du flux TCP. Si la variable `KNOCK_INTLEN` n'est pas posée, la protection de l'intégrité du contenu est désactivée.

11 Utiliser TCP Stealth avec `setsockopt()`

Les développeurs d'applications peuvent intégrer TCP Stealth directement dans leur code. Ainsi, il est possible de contrôler quelles connexions TCP utilisent TCP Stealth, et l'utilisabilité pourra être améliorée. Pour activer le port knocking de base avec un noyau adapté à Knock, l'application doit faire un seul appel à `setsockopt()` après la création du socket TCP :

```
char secret[64] = "This is my magic ID.";

setsockopt (sock, TCP_STEALTH, secret, sizeof (secret));
```

Pour la protection de l'intégrité du contenu, les clients TCP doivent de plus spécifier les premiers octets de données qui seront transmis par un second appel à `setsockopt()` avant d'invoquer `connect()` :

	Port TCP		
Comportement	34343	80	443
Inchangé	126 (93%)	116 (82%)	128 (90%)
Mod. en sortie	5 (4%)	5 (4%)	6 (4%)
Mod. en entrée	0 (0%)	1 (1%)	1 (1%)
Mod. les deux	4 (3%)	13 (9%)	7 (5%)
Proxy (probablement mod. les deux)	0 (0%)	7 (5%)	0 (0%)
Total	135 (100%)	142 (100%)	142 (100%)

TABLE 1 – Modifications du ISN par NAT selon le port de destination mesurées par Honda et al. [4]

```
char payload[4] = "1234";

setsockopt(sock, IPPROTO_TCP, TCP_STEALTH_INTEGRITY, payload,
           sizeof(payload));
connect (sock, ...);
write (sock, payload, sizeof (payload));
```

Pour implanter la protection de l'intégrité du contenu, un serveur doit simplement faire un second appel à `setsockopt()` spécifiant le nombre d'octets à être protégés par TCP Stealth :

```
int payload_len = 4;

setsockopt(sock, IPPROTO_TCP, TCP_STEALTH_INTEGRITY_LEN,
           payload_len, sizeof(payload_len));
```

12 Limitations

De nos jours, la plupart des appareils accèdent à Internet derrière un routeur pare-feu qui fait de la translation d'adresses réseau (« network address translation », NAT). Tandis que TCP Stealth est conçu pour éviter l'utilisation d'informations usuellement modifiées par le routeur NAT, certains routeurs modifient l'horodatage TCP et les ISN et sont alors susceptibles d'interférer avec le fonctionnement du port knocking. Le tableau 1 fournit un résumé des expériences de Honda et al. qui montrent à quel point la modification des ISN par des routeurs NAT est peu répandue en pratique.

En ce qui concerne la sécurité, TCP Stealth est limité aux 32 premiers bits du champ ISN d'un paquet TCP, de sorte qu'un attaquant acharné peut encore rencontrer du succès par chance ou par force brute. Nous pensons néanmoins que TCP Stealth fournit un niveau de protection adéquat contre des attaques non ciblées (telles que HACIENDA). Transférer des services d'administration vers des ports inhabituels peut baisser encore le risque d'une découverte aléatoire par des scanners de ports actifs.

Tandis que techniquement la protection de l'intégrité du contenu est optionnelle avec TCP Stealth, du port knocking sans protection d'intégrité fournit peu de sécurité contre un attaquant qui observe le trafic réseau et prend le contrôle de la connexion après le handshake TCP initial.

Dès lors, des protocoles réseau futurs devraient être conçus pour échanger des clefs au début du premier paquet TCP. Malheureusement, ce n'est pas le cas pour SSH, qui présente à un attaquant une bannière avec de l'information sur la version utilisée bien avant le handshake

cryptographique. Ainsi les défauts de conception du protocole SSH rendent nécessaire un patch d'obfuscation supplémentaire [9] pour effectivement protéger l'intégrité avec TCP Stealth et SSH.

13 Résumé

Des solutions techniques telles que TCP Stealth fournissent une possibilité aux administateurs de durcir leurs systèmes en protégeant des services TCP internes contre des attaques par des criminels, qu'ils soient des particuliers, motivés par des mobiles commerciaux ou des services d'état. Mais comme Linus Neumeier du CCC (Chaos Computer Club) a remarqué récemment dans un OpenEd pour Heise, il peut bien s'avérer impossible de gagner la course aux armes dans le long terme uniquement par des moyens techniques. Sans la volonté politique nécessaire de protéger par la loi, de promouvoir et de financer des systèmes de communication sûrs, cette bataille inégale continuera — et les utilisateurs perdront. Neumann a souligné que des systèmes de communication sûrs seraient possibles, mais que les gouvernements ont bien plus peur d'une perte de contrôle qu'ils ne soutiennent des réseaux durcis (et moins contrôlables). Beaucoup de travail politique est devant nous ; mais déjà aujourd'hui, les fabriquants de systèmes d' exploitation et les administateurs peuvent améliorer la situation en mettant en place des outils de sécurité modernes.

Références

[1] Belgacom attack : Britain's GCHQ hacked Belgian telecoms firm. `http://www.spiegel.de/international/europe/british-spy-agency-gchq-hacked-belgian-telecoms-firm-a-923406.html`, September 2013.

[2] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Zmap : The internet scanner. `https://zmap.io/`, August 2013.

[3] Barton Gellman and Ashkan Soltani. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say. The Washington Post, October 2013.

[4] Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, and Hideyuki Tokuda. Is it still possible to extend TCP ? In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 181–194, New York, NY, USA, 2011. ACM.

[5] Jesper Juhl. Applying patches to the Linux kernel, August 2005. `https://www.kernel.org/doc/Documentation/applying-patches.txt`, visited July 1st, 2014.

[6] Julian Kirsch. Knock. `https://gnunet.org/knock`, August 2014.

[7] Julian Kirsch, Christian Grothoff, Jacob Appelbaum, and Holger Kenn. TCP Stealth, August 2014. IETF draft.

[8] M. Krzywinski. Port knocking : Network authentication across closed ports. *SysAdmin Magazine*, 12 :12–17, 2003.

[9] Bruce Leidl. Obfuscated openssh. `https://github.com/brl/obfuscated-openssh`, April 2010.

[10] Moxie Marlinspike. *knockknock*, December 2009. `http://www.thoughtcrime.org/software/knockknock/`, visited May 5th, 2014.

[11] Laura Poitras, Marcel Rosenbach, and Holger Stark. A wie Angela. `http://www.spiegel.de/spiegel/print/d-126267965.html`, March 2014.