Il programma HACIENDA

Julian Kirsch (Technische Universität München),
Christian Grothoff (Technische Universität München),
Monika Ermert (Heise),
Jacob Appelbaum,
Laura Poitras,
Henrik Moltke
Traduzione italiana di Luca Saiu

15 agosto 2014

1 Introduzione

Fin dagli albori di TCP, il port scanning viene usato dai criminali informatici per individuare sistemi vulnerabili. Un nuovo gruppo di documenti top secret verificato da Heise rivela che nel 2009 l'agenzia di spionaggio inglese GCHQ ha reso i port scanners uno "strumento standard" da applicare contro intere nazioni (Figura 1). Ventisette paesi sono elencati come obiettivi del programma HACIENDA nella presentazione (Figura 2), che è fornita con un'offerta promozionale: i lettori che vogliano fare ricognizione nei confronti di un altro paese devono semplicemente inviare una e-mail (Figura 3). I documenti non specificano i dettagli di alcun sistema di controlli e contrappesi, né giustificano in alcun modo una tale azione. È da notare anche come la possibilità di fare port-scanning su un paese intero non sia un'ipotesi fantastica; nel 2013 è stato implementato un port scanner chiamato Zmap, in grado di fare scanning sull'intero spazio degli indirizzi IPv4 in meno di un'ora, usando un solo PC. [2] L'uso massiccio di questa tecnologia può quindi rendere qualsiasi server ovunque, grande o piccolo, un obiettivo per sabotatori criminali che lavorino per qualche stato.

La lista dei servizi presi di mira include servizi ubiqui come HTTP e FTP, e anche dei protocolli amministrativi diffusi come SSH (Secure SHell protocol – usato per l'accesso remoto ai sistemi) e SNMP (Simple Network Management

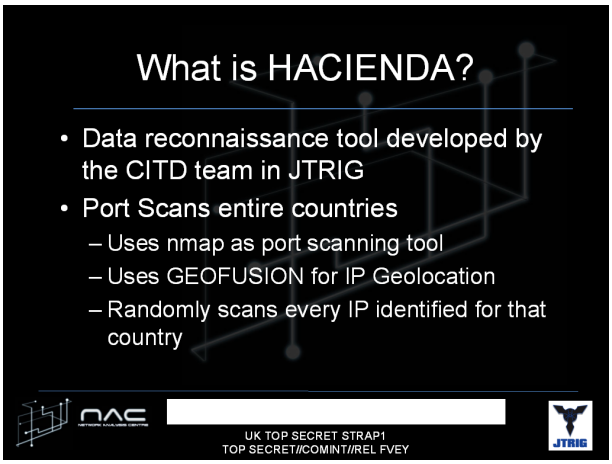


Figura 1

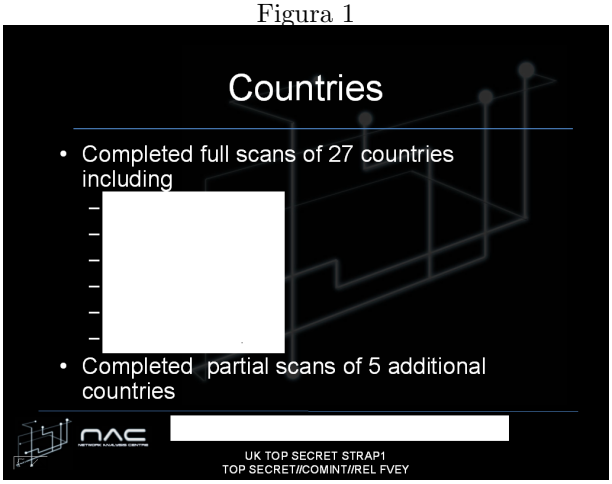


Figura 2

Protocol – usato per l'amministrazione di rete) (Figura 4). Poiché nel frattempo sono stati sviluppati strumenti di port scanning come Zmap che permettono a chiunque di fare scanning massiccio, non è tanto la tecnologia usata ad essere sorprendente in questo caso, quanto la scala enorme e la pervasività dell'operazione.

La prossima sezione fornisce del background sul funzionamento degli strumenti di port mapping e su quali informazioni si possano ottenere per mezzo di essi, in modo da rendere chiaro cosa diventi possibile nel momento in cui un attore-stato li utilizzi su vasta scala.

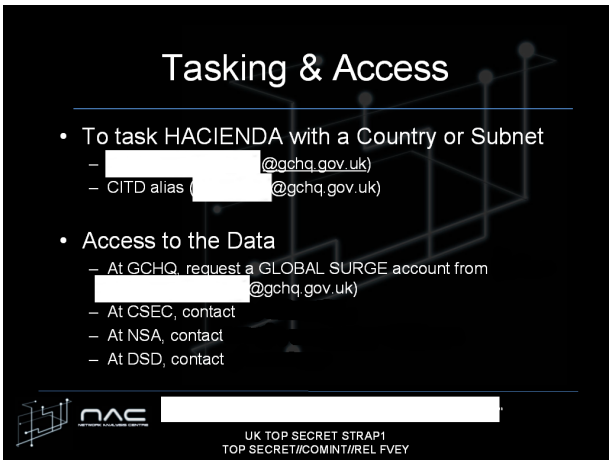


Figura 3

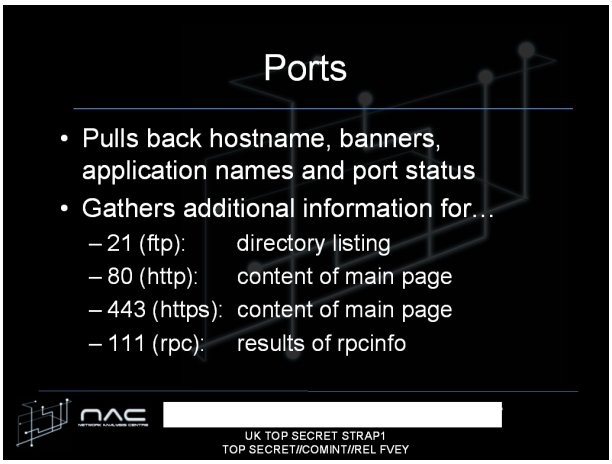


Figura 4

2 Background: la stretta di mano in tre tempi di TCP

Il protocollo usato più comunemente su Internet è il TCP – Transmission Control Protocol. Ogni volta che si invia un'email o si svoglia una pagina web, TCP è il protocollo usato per trasferire dati in modo affidabile tra clients e servers. Gli strumenti di port-mapping sfruttano un problema strutturale in TCP per determinare quali servizi sono attivi su un sistema; fin dagli albori di TCP il port scanning viene usato da chi voglia attaccare per localizzare i sistemi vulnerabili. Ogni volta che un client TCP vuole comunicare con un server TCP, le due parti eseguono quella che viene chiamata la "stretta di mano in tre tempi" di TCP. Il fondamento degli strumenti di port mapping è un difetto di concezione di questa stretta di mano: durante la stretta di mano il server rivela informazioni sulla disponibilità di un servizio senza controllare alcuna autorizzazione del cliente.

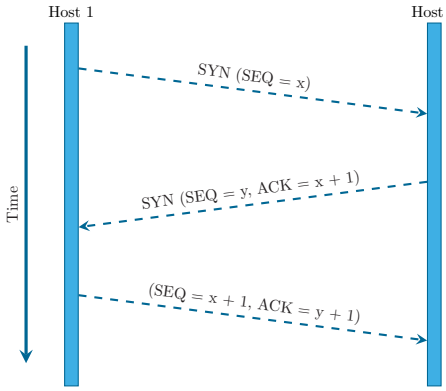


Figura 5: Flusso di pacchetti di una stretta di mano in tre tempi di TCP, eseguita con successo.

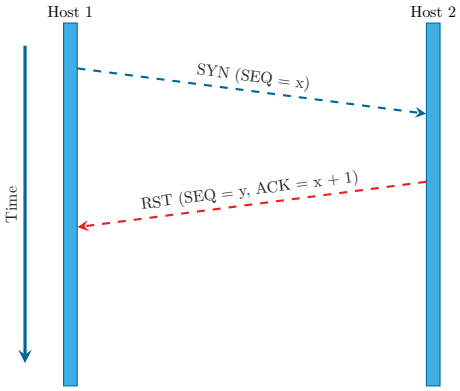


Figura 6: Flusso di pacchetti per un tentativo di connessione ad una porta TCP chiusa.

La Figura 5 mostra la sequenza di pacchetti TCP inviati per stabilire una connessione. Una connessione viene stabilita come segue: l'host che intende iniziare la connessione prima invia un pacchetto TCP SYN ("synchronize"). Se l'host destinatario accetta la richiesta di connessione, invia un pacchetto SYN/ACK ("synchronize/acknowledge"). Dopo aver ricevuto una risposta positiva, l'host che ha iniziato la connessione invia un paccketto ACK ("acknowledge"), che termina la stretta di mano in tre tempi di TCP. La stretta di mano in tre tempi permette a un avversario di determinare facilmente se a una data porta su un host su Internet viene offerto un qualche servizio TCP: se la porta TCP è chiusa, il server reagisce diversamente al pacchetto TCP SYN (Figura 6), inviando un pacchetto RST ("reset") invece del pacchetto SYN/ACK che invierebbe in caso di porta aperta. Quindi un avversario può facilmente mappare i servizi Internet

considerando le differenti risposte dei servers in relazione ai flussi di pacchetti illustrati rispettivamente nelle Figure 5 e 6.

3 Il nemico online

Oltre ad eseguire semplici port scans il GCHQ scarica anche i cosiddetti banners e altre informazioni facilmente disponibili (Figura 4). Un banner consiste di dati, tipicamente in formato testo, inviati da un'applicazione quando si connette a una porta associata; questo spesso mostra informazioni sul sistema e sull'applicazione, compresi il numero di versione e altre informazioni utili per chi è alla ricerca di sistemi vulnerabili. Una ricognizione eseguita sulla scala massiccia rivelata nei documenti indica come l'obiettivo sia una raccolta dati attiva per mappare *tutti* i servizi vulnerabili, e non limitata a obiettivi definiti. Diversamente dal cliché secondo cui si "ascolta tutto"[1], i documenti presentati mostrano un'interazione attiva con reti e sistemi.

Preparandosi ad attacchi contro i servizi offerti attraverso SSH e SNMP, l'agenzia di spionaggio mira a infrastrutture critiche come i sistemi usati per la manutenzione di rete.

Come mostra l'esperienza delle intrusioni in Belgacom [1] e Stellar [11], quando il sistema informatico o le credenziali di rete di un impiegato possono essere utili, i sistemi e le persone vengono presi di mira e attaccati.

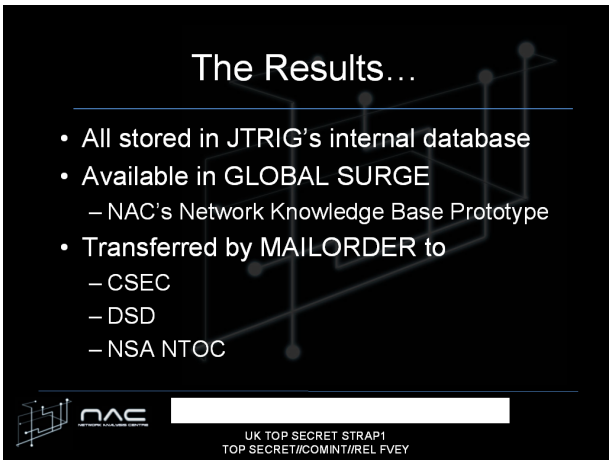


Figura 7

[1]Nota di Luca Saiu del 22 Agosto 2014: Fabio Chiusi critica la mia traduzione, questo passaggio in particolare (`https://twitter.com/fabiochiusi/status/501305936112521216/photo/1`). Non gli do torto: senza dubbio non si tratta di grande letteratura. Ma almeno questa versione ha il pregio della fedeltà all'originale "Rather than the cliche of merely listening to everything". Si tratta di temi delicati e per questo è particolarmente importante non travisare il senso dell'originale: aggiungere un condizionale non è una decisione da poco quando si muovono accuse pesanti. Con più tempo a disposizione si sarebbe potuta migliorare la forma.

Il database risultate dalle scansioni viene poi condiviso con altre agenzie di spionaggio dell'alleanza "Cinque Occhi" (Figura 7), che include gli Stati Uniti, il Canada, il Regno Unito, l'Australia e la Nuova Zelanda. MAILORDER è descritto nei documenti come un protocollo di trasporto sicuro usato tra le agenzie di spionaggio dei Cinque Occhi per scambiarsi i dati raccolti.

4 Ogni apparecchio è un bersaglio

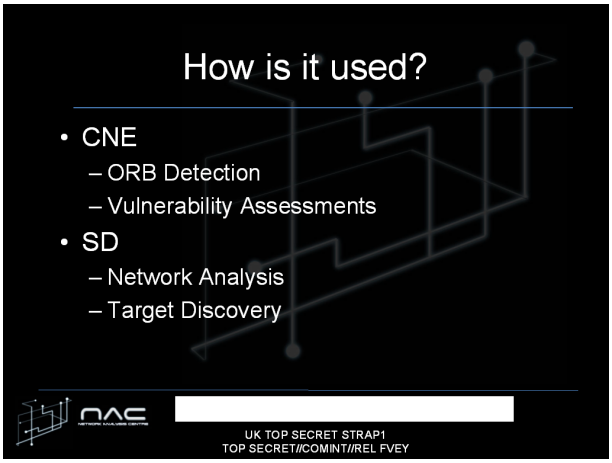


Figura 8: CNE sta per "Computer Network Exploitation".

Il processo di fare scanning su interi paesi e ricercare infrastrutture di rete vulnerabili per prenderne il controllo è consistente con il meta-obiettivo di "domininare Internet" — e Mastering the Internet è anche il nome di un programma di intercettazioni del GCHQ. Queste agenzie di spionaggio tentano di attaccare qualsiasi sistema possano, presumibilmente allo scopo di usarlo come ponte per ottenere accesso ad ulteriori sistemi; un sistema può essere attaccato semplicemente in quanto potenziale strumento per ottenere un cammino di accesso verso un obiettivo di spionaggio di valore — anche in assenza di qualsiasi indizio che una tale situazione si possa prima o poi verificare. Usando questa logica ogni apparecchio diventa un obiettivo da colonizzare, poiché ogni apparecchio controllato con successo è teoricamente utile come mezzo di infiltrazione e monitoraggio, o come una risorsa operativa contro un ulteriore possibile obiettivo.

Il port scanning e il download dei banners al fine di identificare quale software operi sul sistema preso di mira non sono che il primo passo dell'attacco (Figura 8). I documenti top secret di CSEC, NSA e GCHQ verificati da Heise mostrano che le agenzie di spionaggio coinvolte seguono la metodologia comune del crimine organizzato online (Figura 9): la ricognizione (Figura 10) è seguita dall'infezione (Figura 11), dalla messa sotto controllo (Figure 12), e infine dall'esfiltrazione (Figura 13). La presentazione della NSA rende chiaro come l'agenzia adotti la mentalità dei criminali; le slides discutono le tecniche in questione e mostrano delle screenshots dei loro strumenti che supportano questo processo criminale (Figure 14, 15 e 16).

Figura 9

Figura 10

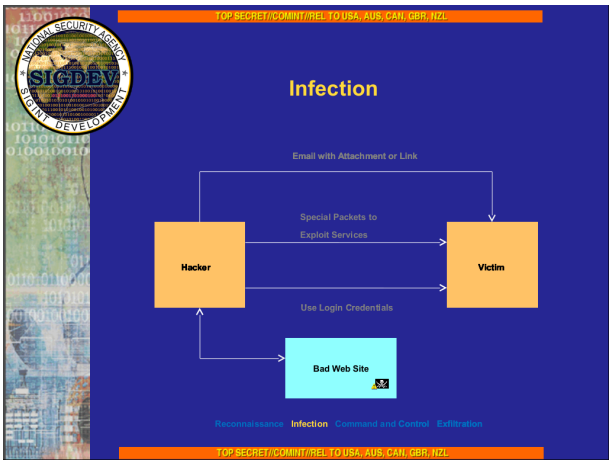


Figura 11

Figura 12

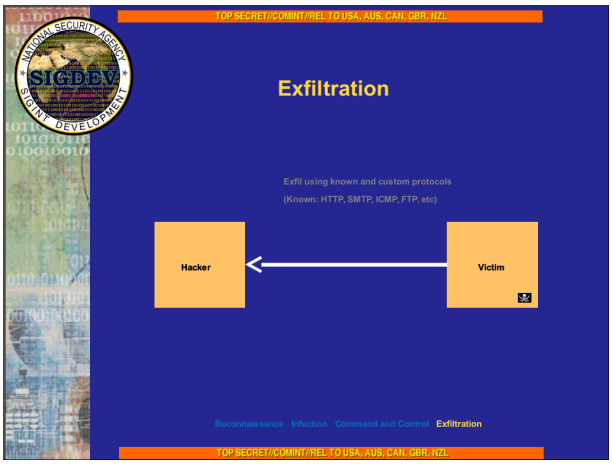


Figura 13

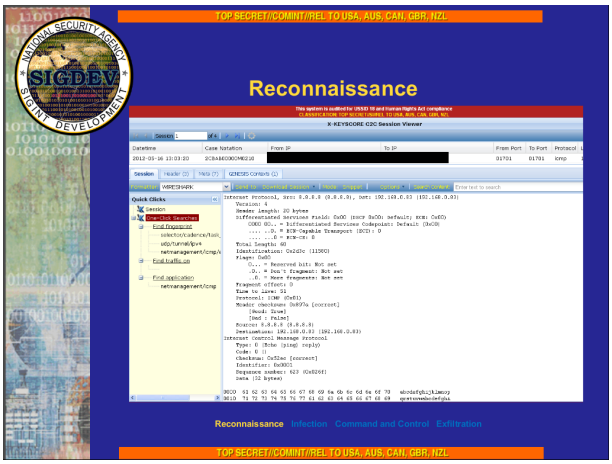


Figura 14

Figura 15

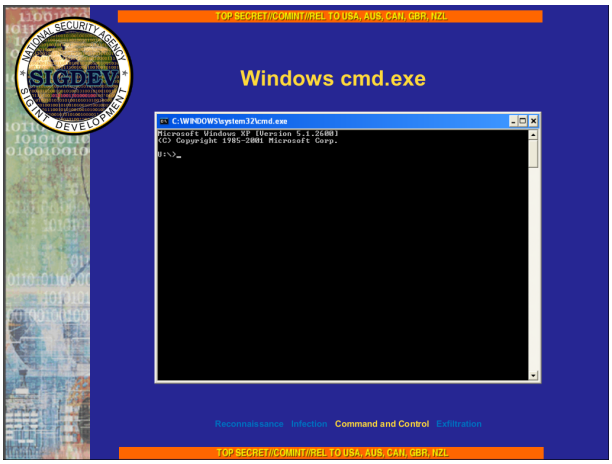


Figura 16

5 La colonizzazione di Internet

La NSA è nota per essere interessata agli attacchi 0-day, ossia attacchi basati su vulnerabilità largamente sconosciute, per cui non esistano patches. Una volta che un avversario armato con attacchi 0-day abbia scoperto un servizio vulnerabile che gira su un sistema, la difesa diviene praticamente impossibile. È improbabile che le firewalls offrano una protezione sufficiente, a causa degli amministratori di rete che possono necessitare comunque di accesso remoto, o delle agenzie di spionaggio che possono essersi già infiltrate nella rete locale [3]; inoltre aggiungere componenti addizionali in una rete interna, come firewalls amministrate via SNMP, potrebbe addirittura aprire nuove vulnerabilità.

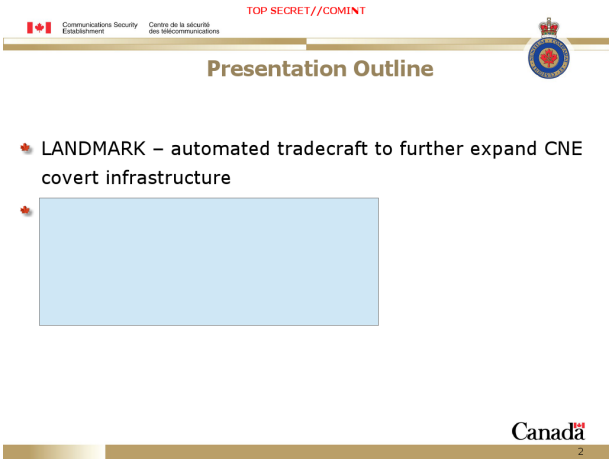


Figura 17

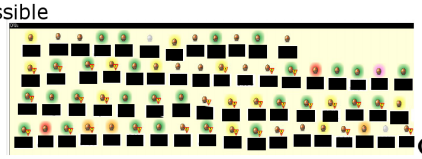
La Figura 8 punta a un ruolo particolare di HACIENDA nell'infrastruttura dei paesi membri del gruppo di spionaggio, particolarmente l'estensione della loro infrastruttura coperta. I documenti top secret verificati da Heise descrivono il programma LANDMARK, un programma dell'agenzia di spionaggio canadese CSEC per espandere l'infrastruttura coperta (Figura 17).

L'infrastruttura coperta include i cosiddetti Operational Relay Boxes (ORBs), degli hosts usati per nascondere la posizione dell'attaccante quando i Cinque Occhi lanciano attacchi contro obiettivi per rubare dati (Figure 18). Diverse volte all'anno il gruppo di spionaggio tenta di prendere controllo *del maggior numero di macchine possibile*, purché si trovino in altri paesi. Per esempio nel febbraio del 2010 ventiquattro spie hanno individuato oltre 3000 ORBs potenziali in un solo giorno di lavoro (Figura 19). In ogni caso, poiché leggere i risultati del port scanning fornito da HACIENDA era considerato troppo laborioso (Figura 20), il processo è stato automatizzato tramite il sistema automatico OLYMPIA (Figura 21). In conseguenza di questo le spie si vantano di poter localizzare apparecchi vulnerabili in una sottorete in meno di cinque minuti (Figura 22).

Communications Security Establishment | Centre de la sécurité des télécommunications

LANDMARK

- CSEC's Operational Relay Box (ORB) covert infrastructure used to provide an additional level of non-attribution; subsequently used for exploits and exfiltration
- 2-3 times/year, 1 day focused effort to acquire as many new ORBs as possible in as many non 5-Eyes countries as possible



Canada

Figura 18

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

Figura 19

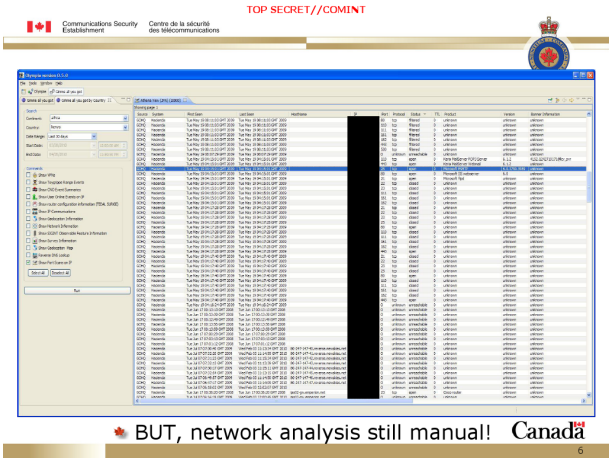


Figura 20

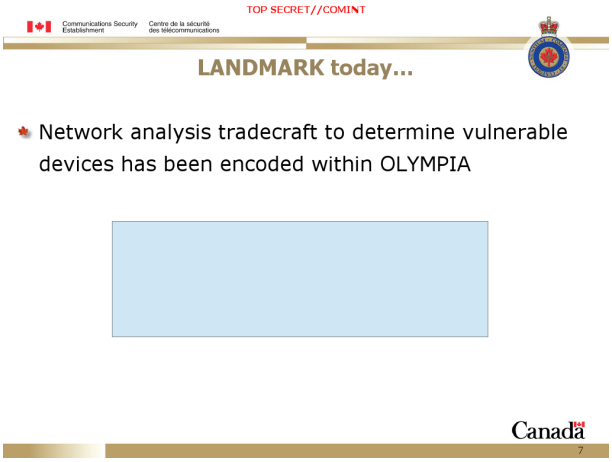


Figura 21

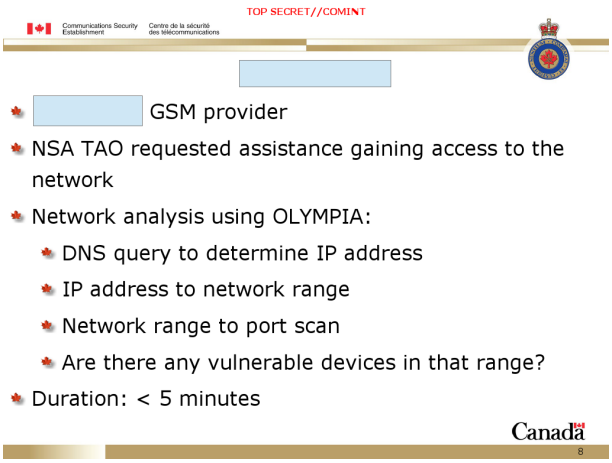

TOP SECRET//COMINT
Communications Security Establishment
Centre de la sécurité des télécommunications
GSM provider
NSA TAO requested assistance gaining access to the network
Network analysis using OLYMPIA:
DNS query to determine IP address
IP address to network range
Network range to port scan
Are there any vulnerable devices in that range?
Duration: < 5 minutes
Canada
8

Figura 22

I canadesi non sono i soli a usare HACIENDA per localizzare macchine da attaccare e trasformare in ORBs. Nel GCHQ la caccia agli ORBs è organizzata come parte del programma MUGSHOT (Figura 23). Il GCHQ ha anch'esso automatizzato il processo, e a causa di questo dichiara un'accuratezza significativamente migliorata (Figura 24). Ancora, l'informazione ottenuta da HACIENDA ha un ruolo di primo piano (Figura 25). Un punto centrale è che con MUGSHOT il GCHQ integra i risultati di scansioni attive (HACIENDA) e il monitoring passivo (Figura 26), per "capire tutto ciò che è importante di tutte le macchine su Internet".

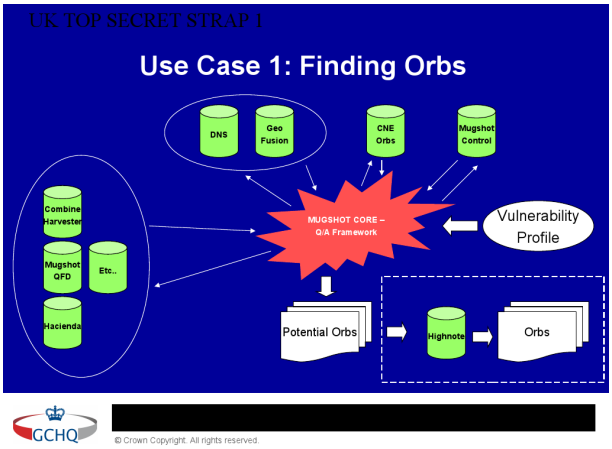


Figura 23

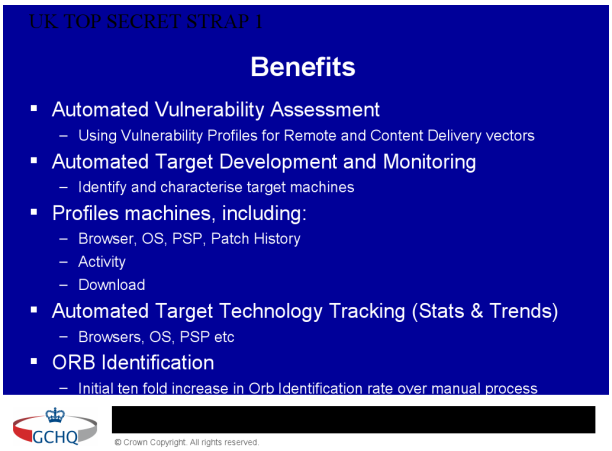


Figura 24

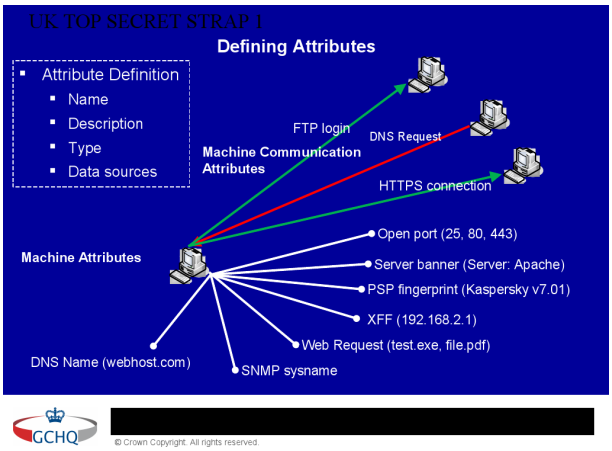


Figura 25

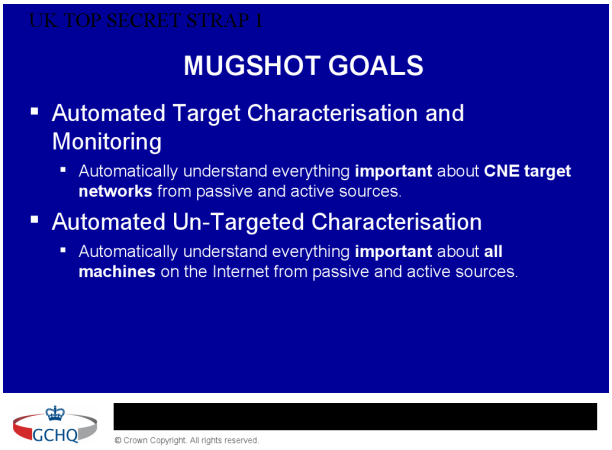


Figura 26

Gli amministratori di sistema e di rete adesso sono esposti alle minacce di spionaggio industriale, sabotaggio e violazioni di diritti umani create da avversari che sono stati-nazione, che attaccano indiscriminatamente l'infrastruttura di rete e forzano i servizi. Un simile avversario ha bisogno di ben poche ragioni per un attacco al di là della semplice possibilità di ottenere accesso, ed è supportato da un budget di diversi miliardi di dollari, immunità dalla giustizia, e collaborazione forzata da parte delle compagnie dei paesi dei Cinque Occhi. Di conseguenza ogni amministratore di sistema o di rete deve preoccuparsi di proteggere il suo sistema da una minaccia di un livello senza precedenti. In particolare i cittadini dei paesi

fuori dai Cinque Occhi, per effetto di questo programma, si ritrovano un grave peggioramento dello stato di sicurezza, privacy e resilienza dell'infrastruttura.

6 Riassunto

Le agenzie di spionaggio stanno usando il loro potere per ottenere controllo dei sistemi in Internet, e da lì proiettare potere. Le loro azioni seguono il modello tipico di comportamento dei cyber-criminali, facendo ricognizione tramite port scanning attivo e passivo per identificare vittime potenziali. Data questa grave minaccia, gli amministratori di sistema devono migliorare le loro misure difensive e, in particolare, ridurre la visibilità dei servizi non pubblici. Le patches dei servizi non proteggono dagli attacchi 0-day, e le firewalls possono essere inapplicabili o insufficienti. Nella seconda parte del nostro articolo introdurremo un'ulteriore opzione perché gli amministratori di sistema possano rendere i servizi di amministrazione non pubblici meno visibili dalle operazioni di ricognizione. Standardizzando queste tecniche la comunità può contrastare il tentativo dei servizi di sicurezza di dominare Internet.

Abbattere HACIENDA

In questo articolo descriveremo una nuova variante di port knocking basata sul modello avversario di uno stato-nazione che faccia scanning attivo, e che offre quindi qualche protezione dal programma HACIENDA, potenzialmente bloccando le agenzie di spionaggio nella fase di ricognizione.

7 Introduzione

Mentre difendersi da vulnerabilità sconosciute nei servizi pubblici è piuttosto difficile è molto più semplice minimizzare la propria impronta visibile, e quindi la propria superficie d'attacco per i servizi amministrativi. Il port knocking [8] è un metodo ben noto per rendere i servers TCP meno visibili in Internet. L'idea di base è che un server TCP non risponda (positivamente) a una richiesta TCP SYN a meno che prima non sia stato ricevuto un pacchetto particolare "knock". Questo può essere d'aiuto per la sicurezza, poiché un attaccante che non possa stabilire una connessione TCP non potrà arrivare al server TCP.

Comunque le tecniche di port knocking tradizionali [10] in genere non considerano un avversario come uno stato-nazione moderno. Specificamente un port scan non è l'unico metodo per apprendere l'esistenza di un servizio; se l'accesso al servizio ha luogo attraverso una rete ove l'avversario può ascoltare il traffico, l'avversario può osservare la connessione e da questa dedurre l'esistenza del servizio. Un attaccante stato-nazione può anche essere in grado di osservare l'intero traffico del client TCP, ed eseguire attacchi man-in-the-middle sul traffico che origina dal client. In particolare, con routers compromessi nell'infrastruttura, è possibile eseguire un attacco man-in-the-middle per appropriarsi di una connessione TCP immediatamente dopo il completamento della stretta di mano TCP iniziale. Un attaccante avanzato in controllo dei routers può anche tentare di identificare l'uso di port knocks insufficientemente nascosti individuando dei patterns inusuali nel traffico di rete. Comunque, può essere ancora sicuro fare l'ipotesi che questo avversario non consideri sospetta una stretta di mano TCP standard, visto che è estremamente comune.

8 TCP Stealth

TCP Stealth ("Discrezione in TCP") è una bozza della IETF [7] che descrive una variante di port knocking facile da mettere in opera e nascosta. TCP Stealth incorpora il token di autorizzazione nel TCP ISN, e permette alle applicazioni di aggiungere delle protezioni sul payload. Di conseguenza TCP Stealth è difficile da individuare in rete visto che il traffico è indistinguibile da una normale stretta di mano a tre fasi TCP, e gli attacchi man-in-the-middle e replay sono mitigati dalle protezioni sul payload. TCP Stealth funziona sia con IPv4 che con IPv6.

TCP Stealth è utile per qualsiasi servizio con un gruppo di utenti di dimensione abbastanza limitata da rendere pratica una condivisione di passphrase con tutti i membri. Gli esempi includono accessi amministrativi SSH o FTP ai

servers, i ponti Tor, servers personali POP3 o IMAP(S) e reti overlay friend-to-friend e peer-to-peer. La maniera più semplice di utilizzare TCP Stealth è attraverso il supporto di un sistema operativo.

TCP Stealth è disponibile per sistemi basati sul kernel Linux attraverso la patch *Knock* [6]. Se i kernels includono questa patch, il supporto TCP Stealth si può aggiungere alle applicazioni attraverso una semplice chiamata `setsockopt()`, o pre-caricando la libreria `libknockify` e impostando le relative variabili d'ambiente.

9 Installazione

Poiché il kernel Linux ufficiale non supporta ancora *Knock* è necessario applicare una patch al kernel della macchina ove si intenda utilizzarlo. Applicare la patch al kernel è molto semplice:

1. Prima di tutto, scaricare i sorgenti del kernel desiderato da `https://www.kernel.org`, se si intende usare un kernel ufficiale. Molte distribuzioni adattano il kernel e quindi forniscono dei sorgenti modificati, che alcuni utenti preferiscono.

2. Una volta che i sorgenti sono dispinibili, scaricare la patch *Knock* appropriata da `https://gnunet.org/knock`. Si noti che, in caso si voglia provare una versione del kernel non tra quelle elencate nel sito web di *Knock*, la migliore opzione è usare le patch per le versioni più vicine.

3. Entrare nella directory dei sorgenti del kernel (si rimpiazzi la parte `<your-version>` a seconda delle versioni del kernel e della patch), e applicare le patches (ulteriori informazioni su come applicare ed annullare una patch sono disponibili sugli archivi di *kernel.org* [5]):

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

4. Ottenere la configurazione del kernel in esecuzione. Esistono diversi metodi:

   (a) Le distribuzioni *Debianoidi* mantengono una copia dei parametri di configurazione in `/boot`; si può copiare il file di configurazione nella directory corrente tramite il comando seguente:

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

   (b) Molte altre distribuzioni compilano il kernel abilitando la possibilità di leggere la configurazione corrente dal filesystem `/proc/`:

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

   (c) Se nessuno dei due casi sopra si applica si può tentare di usare la configurazione di default:

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

Questo comunque tende a non produrre una configurazione perfetta in termini di prestazioni e stabilità.

5. Scegliere il default per tutti i parametri di configurazione che non sono nella configurazione corrente, visto che una versione nuova del kernel può introdurre opzioni di configurazione nuove:

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

6. Attivare *Knock* nella configurazione scegliendo `Networking Support > Networking Options > TCP/IP networking > TCP: Stealth TCP socket support` nel menu interattivo:

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

7. A questo punto il kernel è pronto per la compilazione. Per compilare kernel e moduli:

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

La compilazione può richiedere del tempo. Su macchine multicore si può scegliere il numero di thread di compilazione passando l'opzione `-j` a entrambi i comandi `make` sopra.

8. Se la compilazione ha successo installare il nuovo kernel e tutti i moduli. Successivamente, creare un nuovo *initramdisk* per il kernel. Se `sudo` è installato:

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

Altrimenti i comandi senza la stringa `sudo` si possono digitare ad un prompt di root.

9. Riavviare la macchina, e avviare il nuovo kernel attraverso il boot manager. La macchina adesso supporta *Knock*.

10 Attivare *Knock* con `LD_PRELOAD`

Knock può essere utilizzato senza modificare i sorgenti del programma. Questo può essere utile nei casi in cui il codice sorgente non sia disponibile o quando inserire le chiamate *libc* necessarie non sia fattibile, per esempio a causa di restrizioni imposte dalla logica dell'applicazione.

Per usare *Knock* in applicazioni esistenti viene fornita una libreria dinamica *libknockify*. L'utilizzazione di base di *libknockify* per abilitare *Knock* in un programma `example_program` è la seguente:

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

Dopo di che, se l'applicazione `example_program` comunica via TCP, `libknockify` imposterà le opzioni di sockets opportune per abilitare l'uso di *Knock* nel kernel. Nell'esempio la passphrase condivisa viene dalla variabile con valore `shared secret`, e la protezione dell'integrità del contenuto è limitata ai primi 42 bytes del payload dello stream TCP. Se la variabile `KNOCK_INTLEN` non è impostata la protezione dell'integrità del contenuto è disattivata.

11 Usare TCP Stealth con `setsockopt()`

Gli sviluppatori di applicazioni possono integrate il supporto per TCP Stealth direttamente nel loro codice. Questo ha il vantaggio di permettere di controllare quali connessioni TCP hanno TCP Stealth attivato, il che potrebbe migliorare ulteriormente l'usabilità. Per abilitare il port knocking di base (ovviamente con un kernel che supporti *Knock*) l'applicazione deve semplicemente eseguire una singola chiamata `setsockopt()` dopo aver creato il socket TCP:

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

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

Per la protezione d'integrità del contenuto, i clienti TCP devono inoltre specificare i primi bytes del payload che saranno tramsmessi in una seconda chiamata `setsockopt()`, prima di invocare `connect()`:

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

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

I servers che si aspettano la protezione d'integrità del contenuto necessitano semplicemente di una seconda chiamata `setsockopt()`, per specificare il numero di bytes che ci si aspetta che TCP Stealth protegga:

```
int payload_len = 4;

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

12 Limitazioni

Oggigiorno la maggior parte degli apparecchi per utilizzatori finali accede a Internet attraverso un router gateway che esegue una *traduzione di indirizzi di rete* ("Network Address Translation", NAT). Anche se TCP Stealth è stato progettato per evitare l'uso di informazioni tipicamente alterate in NAT, alcuni

	Porta TCP		
Comportamento	34343	80	443
Invariato	126 (93%)	116 (82%)	128 (90%)
Modificato in uscita	5 (4%)	5 (4%)	6 (4%)
Modificato in entrata	0 (0%)	1 (1%)	1 (1%)
Modificato (entrambe)	4 (3%)	13 (9%)	7 (5%)
Proxy (probabilmente mod. entr.)	0 (0%)	7 (5%)	0 (0%)
Totale	135 (100%)	142 (100%)	142 (100%)

Tabella 1: Cambiamenti in ISN eseguiti da apparecchi per utilizzatori finali, a seconda della porta di destinazione secondo le misure di Honda e altri. [4]

apparecchi NAT modificano le timestamps TCP e gli ISNs, interferendo così con il meccanismo di port knocking. La Tabella 1 riassume gli esperimenti di Honda e altri, che mostrano quanto comuni le modifiche di ISN siano in pratica negli apparecchi NAT.

In termini di sicurezza TCP Stealth è limitato ai 32 bits del campo TCP ISN, e un avversario persistente potrebbe quindi avere successo per fortuna, o con un attacco a forza bruta. In ogni caso riteniamo che TCP Stealth fornirà protezione adeguata contro attacchi indiscriminati senza obiettivi precisi, come HACIENDA. Spostare i servizi amministrativi dalle porte standard può ulteriormente diminuire la probabilità di scoperta accidentale da parte di port scanners attivi.

Benché l'uso di protezione di integrità con TCP Stealth sia tecnicamente opzionale, il port knocking senza protezione d'integrità offre un grado di sicurezza limitato contro un avversario che osservi il traffico di rete e si impossessi delle connessioni dopo la stretta di mano TCP iniziale.

Di conseguenza i protocolli di rete futuri dovrebbero essere concepiti per scambiare informazioni importanti all'inizio del primo pacchetto TCP. Sfortunatamente questo non è il caso per SSH, che invece espone un banner con informazioni sulla versione a un avversario, ben prima della stretta di mano crittografica. A causa di questi difetti di concezione nel protocollo, è attualmente necessaria una patch di offuscazione addizionale [9] per fare uso delle protezioni di integrità di TCP Stealth con SSH.

13 Riassunto

Le soluzioni tecniche come TCP Stealth sono un mezzo per gli amministratori di rafforzare i loro sistemi proteggendo i servizi TCP interni dagli attacchi dei criminali, siano essi privati, motivati commercialmente o stati. Comunque, come recentemente affermato da Linus Neumann della CCC in OpEd for Heise, può darsi che non sia possibile ottenere una vittoria di lungo termine con mezzi puramente tecnici. Senza la necessaria volontà politica di proteggere legalmente, promuovere e finanziare dei sistemi di comunicazione sicuri, questa battaglia

asimmetrica continuerà — e gli utenti perderanno. Neumann ha sottolineato come dei sistemi di comunicazione sicura siano possibili, ma i governi siano molto più interessati alla perdita di controllo che a reti robuste, e quindi meno controllabili. Moltissimo lavoro politico rimane da fare; ma i fornitori di sistemi operativi e gli amministratori possono già oggi migliorare la situazione mettendo in opera delle soluzioni di sicurezza moderne.

Riferimenti bibliografici

[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.