Story of the Cutwail/Pushdo hidden C&C server

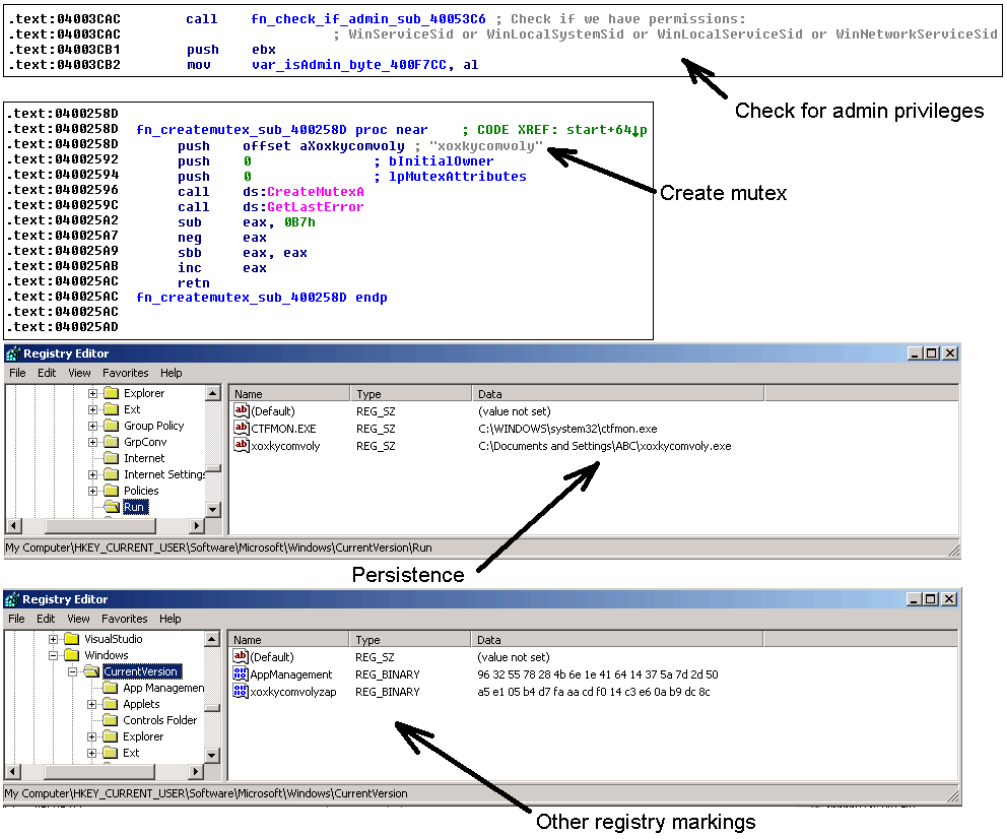
By Threat Intelligence Team 25 Jun 2013

Story of the Cutwail/Pushdo hidden C&C server

This is a loose sequel to the Cutwail botnet analysis blogpost published on the malwaremustdie.blogspot.com. In this blogpost I will primarily focus on the downloaded PE executable itself (SHA256: 5F8FCC9C56BF959041B28E97BFB5DB9659B20A6E6076CFBA8CB2D591184C9164) and the network traffic that it generates. I will also reveal a hidden C&C server.

But first let's quickly go through the things it does at the beginning:
- It registers an exception handler that will only start the process again using CreateProcess().
- It performs a check whether it has admin privileges.
- It checks or creates a mutex named "xoxkycomvoly" (hardcoded identifier used on multiple occasions).
- It checks or creates couple of registry entries under HKCU\Software\Microsoft\Windows\CurrentVersion.
- It checks if the process image filename is "xoxkycomvoly.exe" (it restarts for the first time).
- It nests into the system by creating autorun entry in registry under HKCU\Software\Microsoft\Windows\CurrentVersion\Run.
- It copies itself to the user's profile directory named as "xoxkycomvoly.exe".

Then on the first time an exception occurs and the sample is restarted from the user's profile location named as "xoxkycomvoly.exe".



Initial startup activities

After these initial steps, the sample starts communicating heavily over the network.

The sample contains number of hardcoded hostnames in a plain text form, making a little bit of an impression that it is a simple program, but those are rather only decoys. Let's go chronologicaly and see what it does exactly.

The first group of hardcoded hostnames that takes place is the following list of 6 SMTP servers:

`smtp.live.com`
`smtp.mail.yahoo.com`
`smtp.sbcglobal.yahoo.com`
`smtp.directcon.net`
`mail.airmail.net`
`smtp.compuserve.com`

These SMTP servers are only used to try outgoing TCP connection on port 25, to see if it is filtered in any way (to see if it's gonna be able to send spam).

Then there is another list of plain text hostnames, which look somewhat suspiciously:

`4darabians.nl`
`4dbenelux.be`
`accords-bilateraux.ch`
`4e-energiezentrale.de`
`4effect.ca`
`4egolifestyle.de`
`4elementos.cl`

```
4-elements.ch
4elements.de
4elements.hu
4-elements.se
4emails.de
8wellesley.ca
8zsmost.cz
4enerchi.nl
4entertainmentgroup.tv
4ernila.de
4e-solutions.ch
accounting.ee
0daymusic.biz
0handicap.at
4dbabamozi.hu
accords-bilateraux.ch
0kommanix.de
4effect.ca
4egolifestyle.de
4elementos.cl
4-elements.ch
4elements.gr
4elements.pl
```

Almost all of these hostnames listen on port 25, giving a premature idea that they may be the SMTP relays through which it sends spam, but the sample doesn't really do anything much with these, they are only supposed to distract us from the real business!

The interesting thing that comes next is that the sample also contains several enxored data blobs, each one enxored with different xor key, and these blobs contain some very interesting things, hidden from the plain sight.

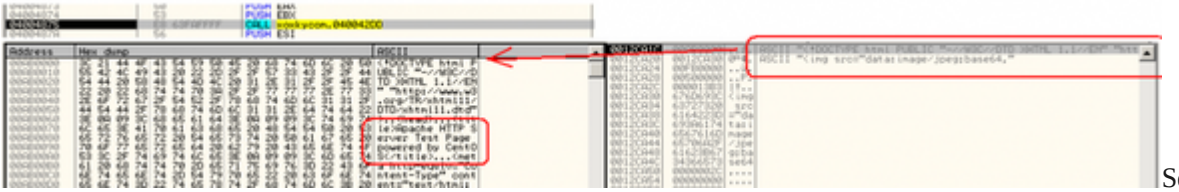
The first data blob that gets dexored is this list of hostnames:

1st group of hidden hostnames

```
4dmobil.at
4eternity.ch
4everandever.de
4everflashlight.de
4ever-hosting.de
4evernet.de
4evernails.nl
4every1.cc
4everything.pl
9online.fr
9welten.de
4everweb.nl
accountingtechs.biz
```

These dexored hostnames look like a good candidates for C&C servers to me. Indeed as the next step the sample is trying to connect to one of these hostnames on port 443 (HTTPS) possibly to obtain a command (Cutwail/Pushdo uses custom binary executable modules). Usage of the HTTPS protocol prevents the actual content to be inspectable in the network capture records, but through reverse engineering we can see that it requests GET /. If the server responds with a reasonable reply (at least 1024 bytes long), it proceeds to check whether the response contains the familiar HTML mark (which is also enxored btw): `<img src="data:image/jpeg;base64`

If that's indeed found in the server response, the data after the mark (obviously not a jpeg) are decrypted with some interesting decryption routines and we get a binary executable (Cutwail/Pushdo module). Then a new svchost.exe process is started in a suspended state and the binary is injected into it and executed.
On the other hand, if the obtained server response does not contain the special HTML mark, no other C&C server from this list is tried. It is simply happy with a web server test page, having no orders from the C&C server. Seems a bit fishy!


Search for special "<img src" mark

Anyway, next enxored blob that comes to the program flow turns out to be this large list of domains:

2nd group of hidden hostnames

Pretty impressive list of 200 hostnames, actually only 176 unique:

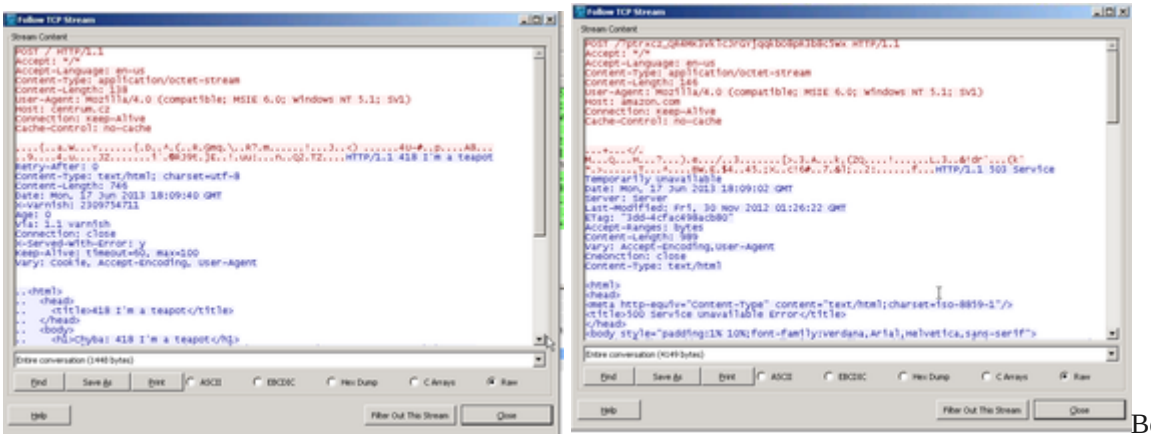
```
accessus.net, accountant.com, actuslendlease.com, agilent.com, allstream.net, amazon.com,
anetsbuys.com, aon.at, arcor.de, aruba.it, atkearney.com, aussiestockforums.com, axelero.hu,
backaviation.com, badactor.us, bassettfurniture.com, beeone.de, bendcable.com, blackplanet.com,
bluewin.ch, bluewin.com, bmw.com, boardermail.com, brettlarson.com, cablelan.net, caixa.gov.br,
canada.com, cbunited.com, centrum.cz, cfl.rr.com, charter.com, chataddict.com, chickensys.com,
claranet.fr, clarksville.com, clear.net.nz, cnet.com, col.com, colorado.edu, comcast.net, cox.com,
cox.net, creighton.edu, cwnet.com, cytanet.com.cy, diamondcpu.com, doctor.com, dr.com, earthlink.com,
earthlink.net, emailmsn.com, embarqmail.com, erzt.com, expn.com, feton.net, floodcity.net, free.fr,
freenet.de, frisurf.no, frostburg.edu, gallatinriver.net, gatespeed.com, gcsu.edu, gmx.net, grar.com,
gravityboard.com, happyhippo.com, hotmale.com, hoymail.com, ia.telecom.net, idea.com,
idealcollectables.com, ig.com.br, imaginet.com, in.com, indosat.com, intelnet.net.gt, intuit.com,
ipeg.com, itexas.net, iupui.edu, iw.com, iwon.com, juno.com, jwu.edu, kazza.com, kcrr.com, kw.com,
lansdownecollege.com, machlink.com, mania.com, marchmail.com, maui.net, mediom.com, metallica.com,
metrocast.net, mexico.com, microtek.com, midway.edu, migente.com, mindspring.com, ministryofsound.net,
msu.edu, mts.net, music.com, mville.edu, mvts.com, mynet.com, mzsg.at, nccn.net, netsync.net,
number1.net, o2.pl, oakland.edu, oakwood.org, optonline.net, orange.pl, oregonstate.edu,
otakumail.com, pandora.be, parrotcay.como.bz, passagen.se, pchome.com.tw, penn.com, pga.com,
picsnet.com, posten.se, potamkinmitsubishi.com, primeline.com, rcn.com, reactionsearch.com,
ricochet.com, rockford.edu, rowdee.com, sexstories.com, shmais.com, south.net, springsips.com,
sscomputing.com, stargate.net, stc.com.sa, surfglobal.net, suscom.net, t-mobel.com, talstar.com,
tampabay.rr.com, tellmeimcute.com, the-beach.net, tiscali.co.uk, uakron.edu, ufl.edu, uga.edu,
ukr.net, uplink.net, uymail.com, vail.com, vampirefreaks.com, verizonwireless.com, vodafone.com,
vodafone.nl, voicestream.com, wcsu.edu, willinet.net, windermere.com, windstream.net, worldnetatt.net,
worldonline.co.uk, www.aol.com, xtra.co.nz, yahoo.com.cn, yahoo.com.hk, yahoo.com.tw, yatroo.com,
zdnetmail.com, zoomnet.net, zoomtown.com
```

The next thing is that it starts a pair of 8 threads. One pair will connect randomly to these hosts on port 80 and

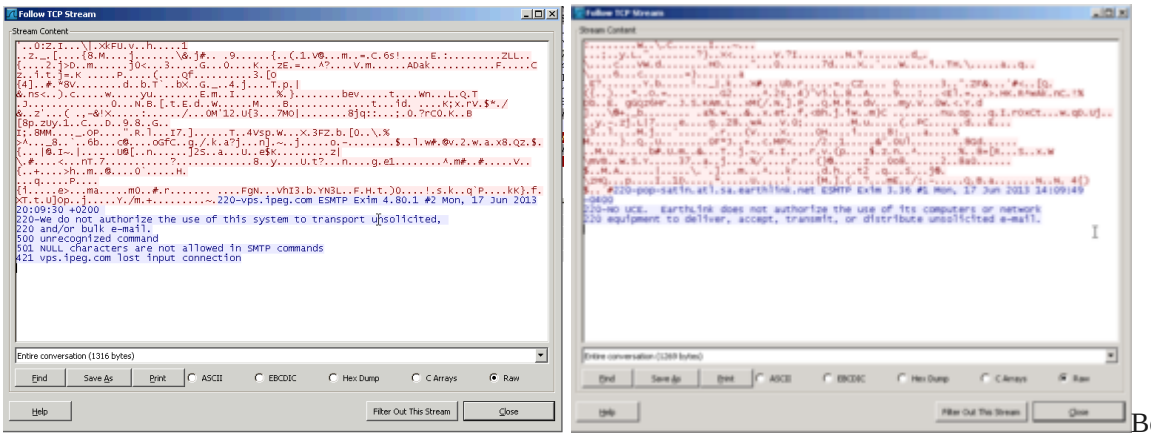
send bogus HTTP requests (like those infamous ptrxcz_ POST requests) and the other pair will connect randomly to those hosts on port 25 and just send some random data to it, not even SMTP traffic, not even waiting for the server banner. This goes on in an endless loop, in the background, 16 noisy threads sending bogus traffic, apparently trying to mask some activity, which is about to start soon.

```
01FB16CE    CC              INT3
01FB16CF    CC              INT3
01FB16D0    55              PUSH EBP
01FB16D1    8BEC            MOV EBP,ESP
01FB16D3    83EC 0C         SUB ESP,0C
01FB16D6    E8 95000000     CALL 01FB1770
01FB16DB    8945 FC         MOV DWORD PTR SS:[EBP-4],EAX
01FB16DE    8B45 FC         MOV EAX,DWORD PTR SS:[EBP-4]
01FB16E1    50              PUSH EAX
01FB16E2    E8 49010000     CALL 01FB1830
01FB16E7    83C4 04         ADD ESP,4
01FB16EA    833D 444FFB01   CMP DWORD PTR DS:[1FB4F44],0
01FB16F1    7E 6B           JLE SHORT 01FB175E
01FB16F3    C745 F8 0A0000  MOV DWORD PTR SS:[EBP-8],0A
01FB16FA    0FB60D 484FFB0  MOVZX ECX,BYTE PTR DS:[1FB4F48]
01FB1701    85C9            TEST ECX,ECX
01FB1703    74 07           JE SHORT 01FB170C
01FB1705    C745 F8 080000  MOV DWORD PTR SS:[EBP-8],8          8 times
01FB170C    C745 F4 000000  MOV DWORD PTR SS:[EBP-C],0
01FB1713    EB 09           JMP SHORT 01FB171E
01FB1715    8B55 F4         MOV EDX,DWORD PTR SS:[EBP-C]
01FB1718    83C2 01         ADD EDX,1
01FB171B    8955 F4         MOV DWORD PTR SS:[EBP-C],EDX
01FB171E    8B45 F4         MOV EAX,DWORD PTR SS:[EBP-C]
01FB1721    3B45 F8         CMP EAX,DWORD PTR SS:[EBP-8]
01FB1724    7D 30           JGE SHORT 01FB1756
01FB1726    6A 00           PUSH 0
01FB1728    6A 00           PUSH 0
01FB172A    8B4D F4         MOV ECX,DWORD PTR SS:[EBP-C]
01FB172D    51              PUSH ECX
01FB172E    68 E014FB01     PUSH 1FB14E0                        send bogus HTTP traffic
01FB1733    6A 00           PUSH 0
01FB1735    6A 00           PUSH 0
01FB1737    FF15 0420FB01   CALL DWORD PTR DS:[1FB2004]         kernel32.CreateThread
01FB173D    6A 00           PUSH 0
01FB173F    6A 00           PUSH 0
01FB1741    8B55 F4         MOV EDX,DWORD PTR SS:[EBP-C]
01FB1744    52              PUSH EDX
01FB1745    68 F011FB01     PUSH 1FB11F0                        send bogus SMTP traffic
01FB174A    6A 00           PUSH 0
01FB174C    6A 00           PUSH 0
01FB174E    FF15 0420FB01   CALL DWORD PTR DS:[1FB2004]         kernel32.CreateThread
01FB1754    EB BF           JMP SHORT 01FB1715
01FB1756    6A FF           PUSH -1
01FB1758    FF15 1420FB01   CALL DWORD PTR DS:[1FB2014]         kernel32.Sleep
01FB175E    33C0            XOR EAX,EAX
01FB1760    8BE5            MOV ESP,EBP
01FB1762    5D              POP EBP
01FB1763    C3              RET
01FB1764    CC              INT3
01FB1765    CC              INT3
```

Start 16 threads generating a bogus traffic noise

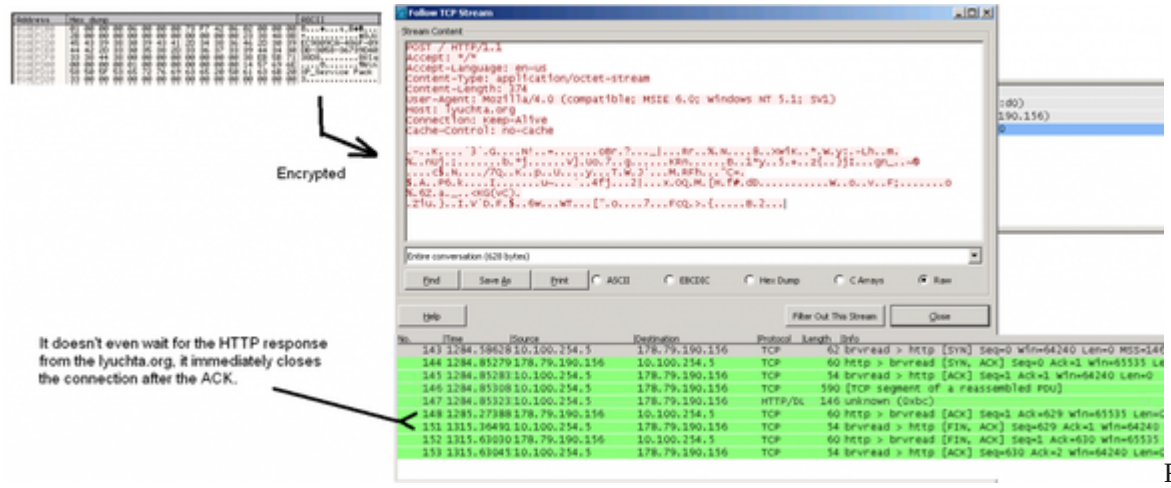


Bogus HTTP traffic examples



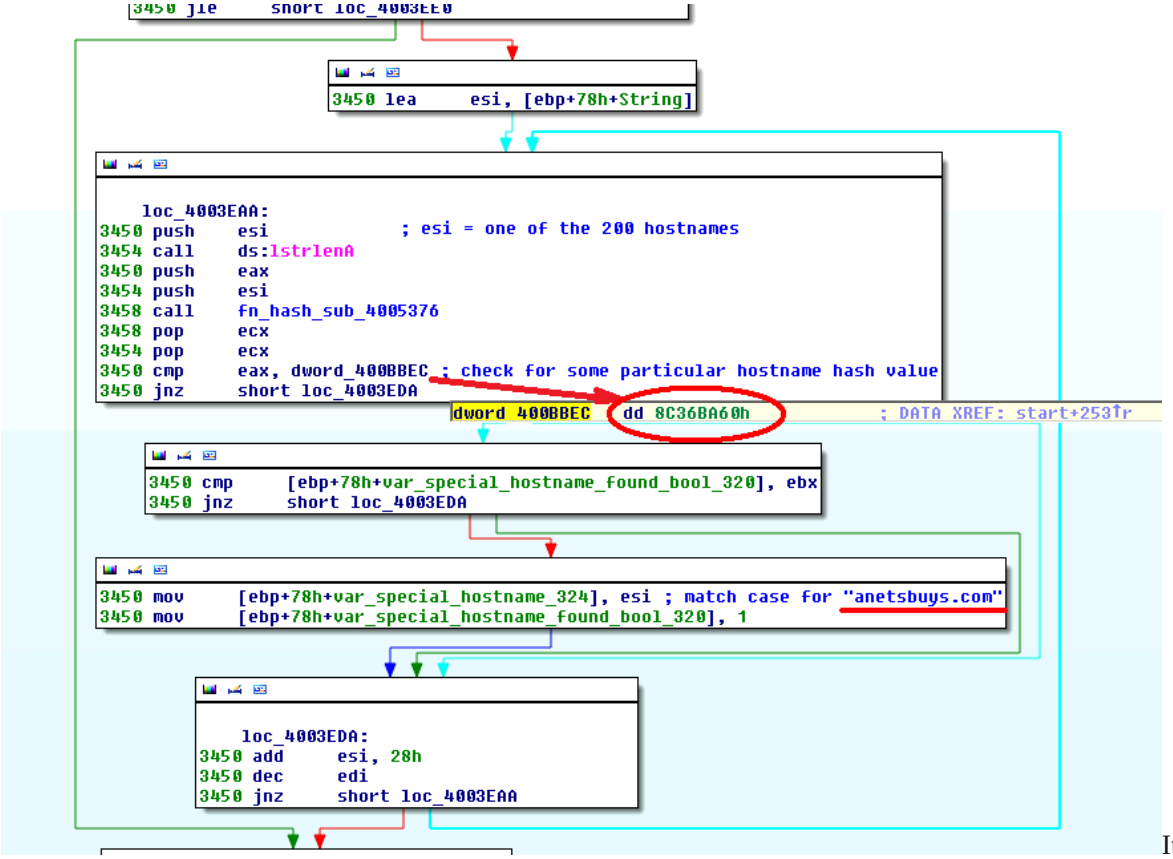
Bogus SMTP traffic examples

OT: Then the sample collects some informations about our client machine such as disk, network adapter info, operating system version etc, encrypts it and sends it to lyuchta.org over HTTP in the background:



Reporting client machine info to lyuchta.org

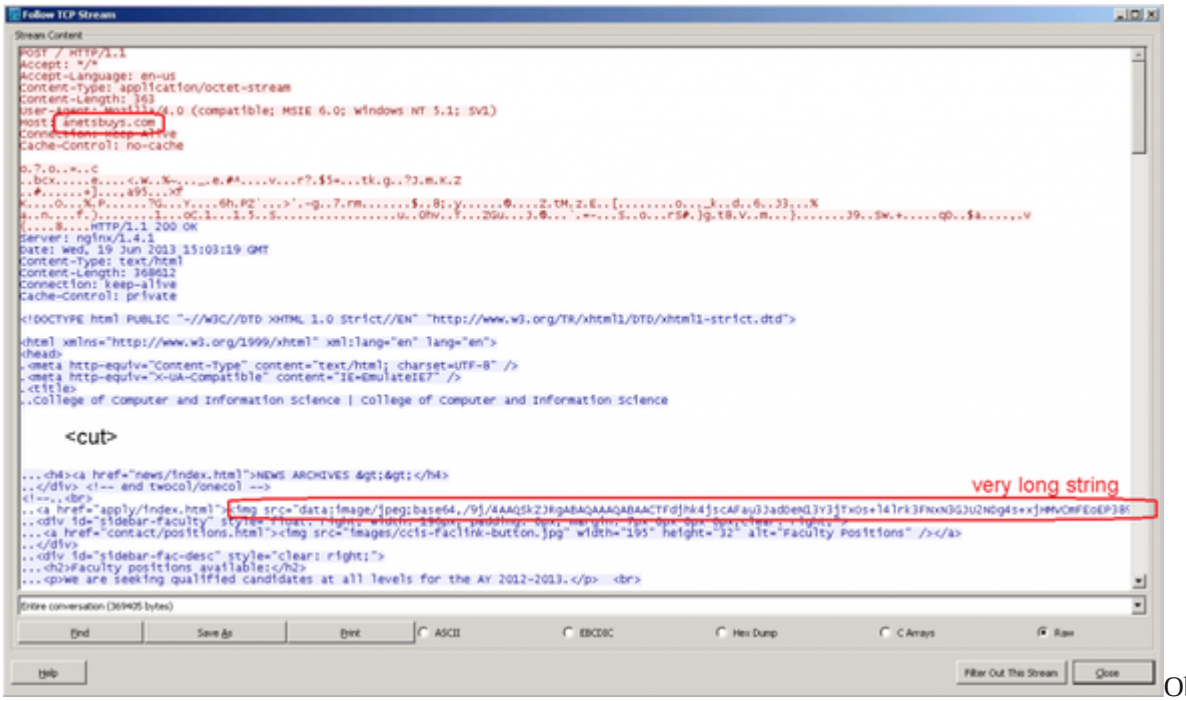
Now there is an interesting code block in the program, where it iterates through all those 200 dexored hostnames (victims of bogus traffic). It calculates some hash value from each one and then compares it to some strange hardcoded value, which only now started to make sense. The comparison condition triggers when comparing "anetsbuys.com" hostname, which happens to match the hash. This is the special C&C server, whose identity is primarily being masked. It's not only hidden in the traffic noise from those 16 annoying threads in the background, but also being supposedly one of the victims.



Iterate through all the hosts from 2nd group and find the special one

This special C&C server is then used to obtain commands (modules) from in the final stage of the malware, where it enters an endless loop and contacts the special C&C server for orders, modules to load. If it is unable to obtain an order from it, then it resorts to generate those characteristic KZ domain names with the DGAs and try to obtain an order from them. Personally I have seen orders to come only from the special C&C "anetsbuys.com". When an order comes and everything goes fine, the sample sleeps for 12 hours, otherwise it sleeps for 2 minutes to repeat the loop and get some command.

Communication with the C&C server:



Obtained 4 modules from this one

Cutwail/Pushdo module decrypted in the memory:

Looks like a typical executable file

The modules then perform actual spamming and other malicious activities.

Source: https://blog.avast.com/2013/06/25/15507/