

Hitpop DDoS Malware Analysis

PUBLIC VERSION

Status: Private version
Distribution: Unlimited

Date: 15 May 2008
ID: b518e4acc853630bc2c2d6ff0121c052

Overview

Beginning in March 2008, Arbor Networks began seeing a new distributed denial of service (DDoS) bot in the wild. This malcode uses an HTTP server to receive commands. This DDoS bot appears to be Chinese in origin, and appears to target Chinese users. At present we are aware of 32 active command and control (C&C) servers, usually located in ChinaNet AS4134 or China169 AS4837. The analysis shared here remains incomplete, with a number of questions presented at the end of the document.



This is a public version of the analysis, most of the URLs have been obfuscated and sensitive portions have been removed.

Introduction

Starting in March 2008, Arbor Networks began to see a distinct new DDoS bot codebase active via captured malware samples. Inspection of these samples and their contacted URLs suggest that this bot has originated, and is controlled by, Chinese-language attackers. All of the C&Cs we have discovered are located in Chinese IP address space.

We have dubbed this malware “Hitpop” based on some of the AV names and the strings present in the bot’s configuration.

Based on AV analysis reports and forum discussions, we think that this tool is relatively new and may have emerged in winter, 2008.

Command and Control

The Hitpop bot uses HTTP to communicate between the infected client who is to participate in the DDoS and the server where the bot master relays their commands. Commands are passed over an unencrypted HTTP channel on port 80.

Client Infection and Query

Infected clients are usually Windows 2000 and XP systems. The DDoS malware can be loaded onto the system through a Trojan horse download as a first stage payload or, more often, as a second stage payload. The Hitpop Trojan is often bundled with other malware, including information stealers such as Greybird. These botnets do not appear to be dedicated solely to DDoS attacks, and it is unclear if this is their primary purpose. However, these are actively used DDoS networks.

In several instances, we have seen the bot malware executable loaded onto a system via a first stage loader that receives download commands via a text file. This text file lists multiple executables, including the link to the bot file. This style of configurable loader appears to be popular in Chinese-origin malware. We have seen several tools that can be configured to load possibly dozens of executables onto a box through a first stage downloader, with names such as “Pangu” and “InfeWeb”. The malware in these situations are usually related to “infostealer”-class malware that steals the user’s gaming credentials, although we sometimes see DDoS bots loaded onto the infected system. This is in contrast to other regions that usually perform more targeted installations of only one or two pieces of malcode, and usually with a direct financial motivation or theft capability.

Here is one example file, taken from late March 2008, from the URL <http://x.xxx.cn/a/ax.txt> (URL obfuscated):

```
http://x.xxx.cn/a/8.exe  
http://x.xxx.cn/a/servstr.exe  
http://x.xxx.cn/a/myself.exe
```

The bot malware is present in the “myself.exe” binary. This is a common name for the first stage Hitpop bot binary based on our observations. It is unclear if this bot is due to one group’s activities or simply a default name for the binary; it may be the latter, as the results of “whois” lookups on the bot distribution domains do not show any obvious connections. The first stage downloader file fetches these files and launches them, further compromising the host. This multi-stage, flexible infection vector is not uncommon in malware delivery, except for the differences noted above.

Once a system is infected, information about the bot’s installation and state appears to be stored in an INI file.

```
cc16.ini [mydown] old_exe =  
cc16.ini [mydown] old_dll =  
cc16.ini [mydown] old_dll32 =  
cc16.ini [mydown] fn_exe =  
C:\WINDOWS\system32\mycc080328.exe  
cc16.ini [mydown] ver = 080328  
cc16.ini [mydown] fn_dll_start =  
C:\WINDOWS\system32\mycc080328.dll
```

Once infection is complete, the host will contact the command and control server with a simple GET request, usually to a script named “active.asp”. The script may be called with arguments to specify which file was installed:

```
http://60.xxx.xxx.219/active.asp?tgid=myself
```

This contact appears to be nothing more than registration and does not appear to unlock any access for the bot.

Infection of the client also creates new files: an executable, a DLL, and a DOS batch file used to remove the malware following some “trigger” event. In some cases this can be because the bot is unable to contact the server for commands, and so the malware simply uninstalls itself. The malware runs as a DLL called from “rundll32.exe” with a specific function as an entry point:

```
rundll32.exe C:\WINDOWS\system32\mycc080328.dll mymain
```

This DLL will then launch the executable created earlier. One of the steps this executable takes is to ensure that the DLL runs (via rundll32) at every system startup by altering the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\run "" = rundll32.exe  
C:\WINDOWS\system32\mycc080328.dll mymain
```

Clients query the server using the standard Internet Explorer binary located at C:\program files\internet explorer\iexplore.exe, which is executed by the previous executable. Commands are fetched by the client through a simple HTTP “GET” request to a URL

hardcoded in the binary, usually with a filename of “cc.txt”. A list of monitored URLs is presented later in this document.

Unlike other HTTP bots, there is no unique ID reported back to the server, and no password is needed to retrieve the command file. This has allowed us to analyze the attacks commanded by the botnet.

Server Characterization and Response

Initial analysis indicates that the servers run on Windows with IIS, unlike other HTTP botnets we have been tracking that use a server running Apache and PHP. The web front-end appears to be coded in ASP pages with VBScript. The command configuration appears to be written out as a static text file. It is unclear from our analysis at this point if this command server is backed by a database or not.

The server responds with a plaintext series of commands as a simple text file. The client handles dissecting the fields to discover what commands to act on and what targets to launch the attack against. Unlike other HTTP bots such as BlackEnergy or Barracuda, this bot uses an unencoded, plain text format to pass the data back to the host. The

```
www.wowbeez.com|80|@@|2000|y|150|zh-cn
www.omg-wow.com|80|@@|2000|y|150|zh-cn
www.woweternal.info|80|@@|2000|y|150|zh-cn
www.wowbeez.com|80|@@|2000|y|150|zh-cn
www.omg-wow.com|80|@@|1000|y|150|zh-cn
www.woweternal.info|80|@@|1000|y|150|zh-cn
www.toxic-wow.net|80|@@|1000|y|150|zh-cn
www.omg-wow.com|80|@@|1000|y|150|zh-cn
www.toxic-wow.net|80|@@|1000|y|150|zh-cn
www.woweternal.info|80|@@|2000|y|150|zh-cn
www.toxic-wow.net|80|@@|2000|y|150|zh-cn
www.omg-wow.com|80|@@|1000|y|150|zh-cn
www.woweternal.info|80|@@|1000|y|150|zh-cn
www.toxic-wow.net|80|@@|1000|y|150|zh-cn
www.toxic-wow.net|80|@@|1000|y|150|zh-cn
www.wowbeez.com|80|@@|1000|y|150|zh-cn
0.0.0.0|80|@@|1000|y|150|zh-cn
0.0.0.0|80|@@|1000|y|150|zh-cn
0.0.0.0|80|@@|1000|y|150|zh-cn
0.0.0.0|80|@@|1000|y|150|zh-cn
180|1
070930|||
GET%20/%20HTTP/1.1%0D%0AAccept%3A%20*/%0D%0AAccept-
Language%3Azh-cn%20%0D%0AAccept-
Encoding%3A%20gzip%2C%20deflate%0D%0AUser-
Agent%3A%20Mozilla/4.0%20%28compatible%3B%20MSIE%206.0%3B%2
0Windows%20NT%205.1%3B%20SV1%29%0D%0AHost%3A%20%0D%0ACon
nection%3A%20Keep-Alive%0D%0A%0D%0A|
Èç' ûÄúµÄÒ³ ÄæÄ»ÓÐ ×Ô¶¯Ìø×ª£-Çë|
```

```
meta http-equiv="refresh" content="0;URL=|
window.location =|
window.location=|
end|
end1|
```

Access to the server appears to be controlled through a minimal username and password login over the web. The form also includes a simple 4-digit CAPTCHA approach to prevent others from launching brute-force attacks on the login credentials. A sample of the login window – which is similar across all of the C&C servers we are tracking – is shown below.



Interestingly, the CAPTCHA mechanism appears broken. Inspection of the HTML and VBScript source of the page contains this snippet:

```
end if
loginform.systempassword.value=trim(loginform.systempassword.value)
if len(loginform.systempassword.value)<>4 then
    msgbox "验证码输入不正确!"
    exit sub
end if
loginform.submit
end sub
```

This field – `loginform.systempassword` – is the 4-digit CAPTCHA code that changes with every page reload. This code appears to only see if the code is 4 digits long, not the

presented value. This CAPTCHA provides no additional security against an automated brute force attack on the login credentials.

We have not been able to inspect the administrative UI for this botnet, either in isolation or of an active botnet.

Searching the web for unique strings within this page's HTML source using Google shows no unique matches. In fact, this code appears most commonly in Chinese language ASP tutorial pages, suggesting this code may have been lifted from such a page.

Dissecting the Received Commands

The command file (usually "cc.txt") passed back to the bot contains 5 major sections. The file is entirely unencoded (ie no Base64 encoding, and certainly no significant encryption applied).

The first section lists the targets by IP or hostname, port, and URI. A valid target may look like the following:

```
www.wowbeez.com|80|@@|1000|y|150|zh-cn
```

When the bot is directed to target a specific URL other than "/", the third field in the above command is replaced with the path to the resource on the server:

```
bbs.61229.com|80|/bbs/reg.cgi|2000|y|1|zh-cn
```

A newline character separates specific targets. If the target host is simply "0.0.0.0", no attack will be launched.

The meanings of the remainder of the command, including the "1000", "y", "150" and "zh-cn", are unclear at this time. See "Unresolved questions" below.

The second section contains two fields:

```
180|1
```

This appears to be a time delay for the bot to return to the server for new commands. The purpose of the second field is unclear, but may simply be a Boolean flag to indicate that the bot should return for new commands.

The third section is a simple number and is often with empty fields:

```
070930|||
```

This value – 070930 – may be the bot's version ID. At times the second field of this line may contain a URL to an executable. When we download this binary we find new Hitpop

bot malware. This may be a simple update mechanism for the bot, meaning that if the version of the bot differs from the command file's version then the executable is downloaded and installed.

The fourth section appears to be an HTTP request, complete with headers.

```
GET%20/%20HTTP/1.1%0D%0AAccept%3A%20*/%0D%0AAccept-  
Language%3Azh-cn%20%0D%0AAccept-  
Encoding%3A%20gzip%2C%20deflate%0D%0AUser-  
Agent%3A%20Mozilla/4.0%20%28compatible%3B%20MSIE%206.0%3B%2  
0Windows%20NT%205.1%3B%20SV1%29%20%0D%0AHost%3A%20%0D%0ACon  
nection%3A%20Keep-Alive%0D%0A%0D%0A|
```

The fifth and final section of the commands given back to the bot appear to be an HTML template for malicious website redirection. When the text is decoded, it roughly translates to "If the page does not automatically change, please ..." This may be part of a Trojan horse website creation that the bot creates.

```
Èç' ûÄúµÄÒ³ ÆæÄ»ÓÐ ×Ô¶¯ìø×ª£-Çë|  
meta http-equiv="refresh" content="0;URL=|  
window.location =|  
window.location=|  
end|  
endl|
```

This section appears to contain a path to an executable to be downloaded, which can yield additional infections. The EXE URL shown here appears to be a new version of the bot; the current bot can compare its own version ID to the advertised version ID here and update as needed.

```
meta http-equiv="refresh" content="0;URL=|  
window.location =|  
end|  
http://202.xxx.xxx.96/jnhack.exe|213|108300|0|17  
endl|
```

To date we have not found any sites specifically modified by the Hitpop malware.

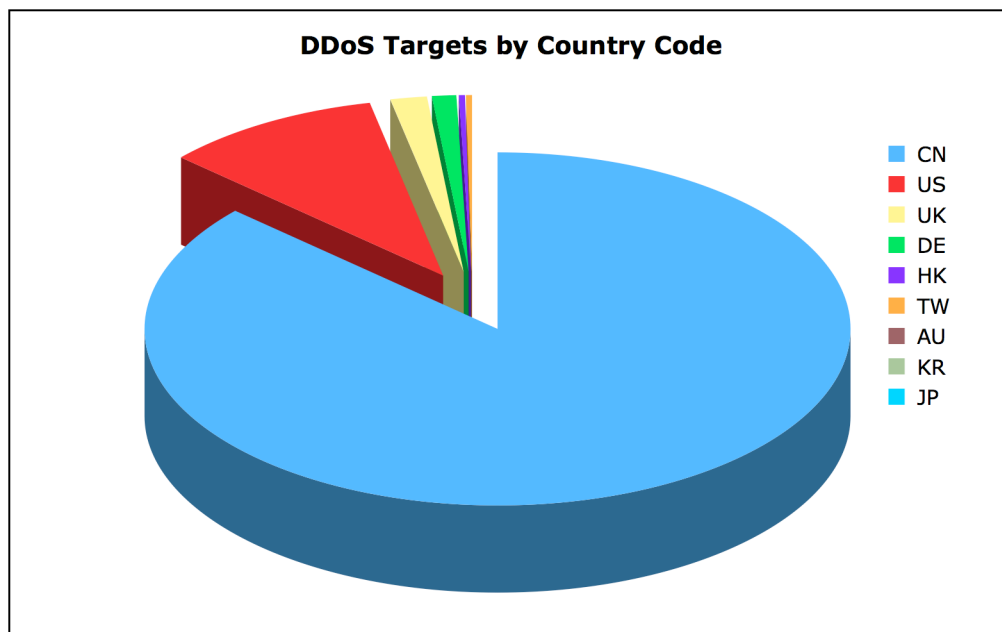
These last two sections mix an HTTP GET request with headers together with an HTML response template, are confusing. They do not appear to be directly related, as they are opposite ends of an HTTP communication.

Importantly, Hitpop appears to be limited to performing HTTP floods only. While the port and URL are configurable, this is markedly different from other bots that can perform ICMP, UDP or SYN floods.

DDoS Activity

Since the initial analysis and characterization of this malware, we have been actively analyzing the C&C servers' commands to discover the targets of the attacks.

The bulk of the attacks we have logged target Chinese sites, usually web boards or community sites. We have not logged any significant attacks against high-profile sites to date. The breakdown of the countries targeted by the Hitpop botnets we are actively monitoring are shown below.



Many targets of these attacks have been online gaming sites and forums, especially World of Warcraft-related sites. We have seen a significant amount of Chinese-language malware targeting online games and WoW in particular as of late. C&C Activity has not been distributed evenly, with some of the C&C hosts being far more active than the others.

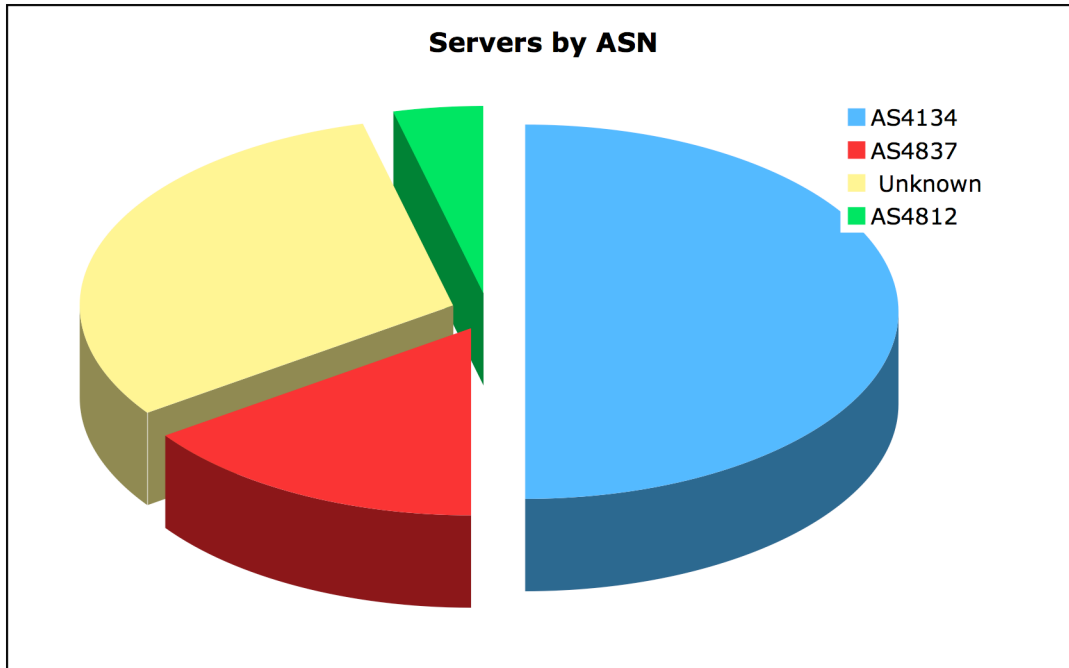
AV Detection

This malware appears poorly characterized by the AV community. AV names, when present, are ambiguous in many cases, such as “Delf”, “Suspicious”, “Dropper” or “Win32.Agent”. Major AV tools such as Symantec do not recognize several samples that we have analyzed. The following table represents a compendium of AV names gathered from several samples.

Microsoft	Variants of Hitpop, Pophot, Meredrop
Kaspersky	Variants of Trojan.Win32.Delf
McAfee	Variants of New Malware
Sophos	Mal/Behav-151

C&C Sites

Most, if not all, of the active Hitpop C&Cs that we have discovered are located in Chinese IP address space. The following shows the breakdown of the C&C location by ASN.



Both the character sets used by the malware and the whois information suggest Chinese-language authors behind the malware, and Chinese-language users of the malware.

Open Questions and Requests

This analysis is incomplete at this time and is missing several key elements for completeness. Available resources and language barriers, as well as our visibility into Chinese-specific threats limit us from performing additional analysis. Open questions at this time include:

1. Is the source of this malware, or a kit, available? How widely?
2. What is the purpose of the additional fields in the command reply from the server? It is likely that they specify the number of threads or the type of attack.
3. How big are these botnets?

Acknowledgements

A number of individuals and teams have helped share some information about this malware but will remain anonymous at their request. Their contributions to helping in this analysis through review, samples, and insights have been appreciated.