Tracking DDoS Attacks: Insights into the Business of Disrupting the Web

Armin Büscher[§] and Thorsten Holz[‡]

[§]Websense Security Labs
abuescher@websense.com

[‡] Ruhr-University Bochum
thorsten.holz@rub.de

Abstract

Known for a long time, *Distributed Denial-of-Service* (DDoS) attacks are still prevalent today and cause harm on the Internet on a daily basis. The main mechanism behind this kind of attacks is the use of so called *botnets*, i.e., networks of compromised machines under the control of an attacker. There are several different botnet families that focus on DDoS attacks and are even used to sell such attacks as a service on Underground markets.

In this paper, we present an empirical study of modern DDoS botnets and analyze one particular family of botnets in detail. We identified 35 Command and Control (C&C) servers related to *DirtJumper* (also called *Ruskill*), one of the popular DDoS botnets in operation at this point in time. We monitored these C&C servers for a period of several months, during which we observed almost two thousand different DDoS attacks carried out by the botmasters behind the botnets. Based on this empirical data, we performed an analysis of the characteristics of DDoS attacks. To complement this C&C-centric point of view, we briefly analyzed the information logged at two different victims of *DirtJumper* DDoS attacks to study how such attacks are perceived at an endhost. Our results provide insights into modern DDoS attacks and help us to understand how such attacks are carried out nowadays.

1 Introduction

Many different kinds of *Denial-of-Service* (DoS) attacks are known and they constitute a recurring pattern in the area of computer security. Especially *Distributed Denial-of-Service* (DDoS) attacks, in which a large number of systems concurrently perform a DoS attack against a victim, play an important role [14]. The first well-documented DDoS attacks were observed back in the year 1999, when a tool called *trin00* was studied by Dittrich [7]. He detected a network of more than 220 compromised systems that was used to flood a single server located at the University of Minnesota starting on August 17, 1999. This machine was offline for over two days and at least sixteen other systems were also attacked by the same DDoS network. In the following months, many similar attacks were observed and these incidents even received attention from the mainstream media when sites such as for example *Yahoo!*, *Amazon*, and *eBay* were targeted in February 2000.

Since then, this kind of attacks has evolved and DDoS attacks are still an important attack vector to consider. Due to their proliferation and practical importance, DDoS attacks have also received a lot of attention from the research community (e.g., [4, 8, 11, 12, 16, 21, 25]). Many different kinds of detection and mitigation approaches were developed over time. Nevertheless, these attacks are still prevalent and cause harm on the Internet.

Botnets as Root-Cause. These days, DDoS attacks are typically performed by so-called *botnets*, i.e., networks of compromised machines under the control of an attacker (often called *botmaster*) [6,8,19]. A typical DDoS attack is carried out as follows: the botmaster instructs the infected machines to perform an attack against a specific victim, and then all these machines (called *bots*) overwhelm the victim with requests by for example sending SYN packets (leading to resource starvation at the victim) or large UDP packets (leading to congestion of the network link). This is the same basic mechanism that was used by *trin00*, with the difference that the malicious code used in modern attacks and the flooding techniques have evolved over time.

In this paper, we perform an empirical study of modern DDoS botnets and analyze how such attacks are carried out nowadays. On the one hand, we analyze more than 450 binaries classified by anti-virus vendors as *DirtJumper* (also called *Ruskill*), one of the popular families of DDoS-related malware. We attempt to detect the Command & Control (C&C) servers belonging to these bots in an automated fashion by executing the sam-

ples in a controlled environment. This analysis enables us to detect 35 live C&C servers for our study, which we observed for a period of four months between October 2011 and January 2012. In total, we observed 1,968 different DDoS attacks performed by these botnets. We focus our evaluation on the commands that were sent by the botmasters to infected machines. This enables us to obtain insights into typical attacks such as for example the attacked server port (e.g., 85.7% of the attacks targeted port 80 / HTTP) and the temporal distribution of attacks (e.g., about one third of the DDoS attacks only last for up to one hour). On the other hand, we complement this C&C-centric view with a brief analysis of server-side victim logs related to two specific attacks: by contacting the victims of two DDoS attacks detected by our monitoring system, we were able to obtain webserver access logs of actual attacks. An analysis of these traces allowed us to study how such attacks are perceived at an endhost, e.g., to determine the countries bots are coming from and the impact of such attacks.

Related Work. This paper continues a line of work in which empirical studies were performed to understand modern botnets. More specifically, previous studies analyzed botnets such as *Storm Worm* [10], *Conficker* [17], *Torpig* [22], *Waledac* [15], *MegaD* [5], and *Cutwail* [23]. These studies introduced methods and techniques to observe botnets and helped to obtain insights into the mechanisms behind such networks of compromised machines. As a result, there is a toolkit of methods that can be used to track botnets; in our case we used these techniques to study different *DirtJumper* C&C servers in detail.

In addition, several empirical studies were performed to understand aspects of misuse on the Internet such as for example the spam campaign trail [13], fake anti-virus campaigns [18], or pay-per-install schemes [3]. We focus our study on DDoS attacks and provide insights into how such attacks are carried out. This complements previous studies in this area and presents further insights into the mechanisms behind current attacks.

Outline. In summary, we make the following three contributions in this paper:

- We provide an overview of DDoS-related bots with a specific focus on *DirtJumper*, one of the prolific malware families related to DDoS attacks.
- We perform an empirical study of *DirtJumper* botnets, in which we studied 35 C&C servers over a period of several months and thereby observed almost two thousand DDoS attacks.
- To add to this C&C-centric point of view, we also briefly study how modern DDoS attacks affect victims by analyzing victim logs of two particular *DirtJumper* incidents.

2 Empirical Overview of DDoS Ecosystem

In this section, we provide an empirical overview of the current market for DDoS-related malware and study popular ready-to-use kits sold and exchanged in the cybercrime Underground. It is based on manual investigative work in related forums and the manual analysis of DDoS tools and malware acquired during the investigation.

2.1 DDoS in the Underground Ecosystem

The DDoS malware families we analyzed for this study all had a similar architecture which can also be found in malware kits for other purposes (e.g., banking Trojans or password stealers like *ZeuS* and *SpyEye*). The command and control part is provided as a so-called *panel*, a web-frontend that can be installed on common webserver configurations using the server-side scripting language *PHP* and a database (typically *MySQL*). The installation of the C&C panel is similar to the installation of a web application and typically includes aspects such as automated installation scripts to create database tables and a basic configuration. Aside from providing an interface to issue commands to the botnet, the panels also provide statistical information for the botmaster. The other half of the malware kits is a so-called *builder*, a program that can be used to create new malware samples with the chosen configuration for C&C URL and other parameters. After creating a new instance of the bot, the botmaster will try to distribute the malware to computers, for example by sending spam emails with a malicious link or attachment, or performing drive-by download attacks. Eventually, this enables the botmaster to infect more machines (i.e., create new bots) that connect back to the C&C panel.

Popular DDoS malware families we discovered in Underground forums include *DirtJumper*, *Darkness*, and *Gbot*. The prices for the DDoS malware kits ranged from just under one thousand US dollars for the *Darkness* kit to free pirated downloads of the *DirtJumper* kit. The cybercrime ecosystem also provides so-called *DDoS-for-rent* services saving the customer the hurdle of maintaining the necessary infrastructure. The pricing in the advertisement posts we found started at 5 US dollars per hour and promised incremental rebates, e.g., one day worth of DDoS attacks for just under 50 US dollars.

At the time of starting our analysis, the malware *DirtJumper* had the highest number of samples found in-the-wild through *VirusTotal* of all DDoS tools we came across. The kit appears to be especially popular in Russian Underground forums. We found three versions of free *DirtJumper* downloads in different Underground forums, which enabled us to dissect the *panel* and *builder* components. We thus decided to analyze this malware family in greater detail.

2.2 Analysis of *DirtJumper*

We now provide an overview of the abilities of the *DirtJumper* bot with a focus on the C&C communication. The URL of a C&C server is embedded into the executable file at the time a new sample is generated by a botmaster using the *builder* component. After infection of a system, the malware generates a random bot identification number and contacts the designated C&C server with a HTTP request to obtain the current attack commands. The bot ID is stored in the file `keys.ini` located in the Windows system directory and used to identify the bot during communication with the server and subsequently generate statistics to display on the *panel* website. The server answers the bot request with the attack command entered and saved by the botmaster in the *panel* configuration form. The *DirtJumper* bot does not use any form of encryption for the HTTP communication with a C&C server. Listing 1 shows an example of a bot querying for new commands and the C&C server replying with an attack command. Domain names have been obfuscated to avoid harming victims.

Listing 1: Example of DirtJumper C&C communication.

```
POST /index.php HTTP/1.0
Host: ***C&C***.com
Keep-Alive: 300
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1)
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

k=807789926667168
------------------------------------------------------
HTTP/1.1 200 OK
Date: Thu, 02 Feb 2012 21:33:48 GMT
Server: Apache
X-Powered-By: PHP/5.2.17
Vary: Accept-Encoding,User-Agent
Content-Length: 30
Connection: close
Content-Type: text/html

01|50|60http://***VICTIM***.com
```

When phoning home to the C&C server, the infected machine performs a HTTP POST request in which the bot ID is included. We can easily emulate this kind of requests by performing a HTTP POST request with an arbitrary bot ID, for example with the following command using the command-line tool *cURL*:

```
curl -X POST -d "k=12345" http://***.com/index.php
```

The attack command returned by the C&C server consists of four parts: attack mode, number of threads, timeout until the next command query, and the list of URLs to attack. In the example listing, the attack mode is `01`, which translates to HTTP flood. The supported attack modes for DirtJumper are:

- `01` – *HTTP flood*: Performs conventional HTTP GET requests to the target webserver and disconnects as soon as the first response packet is received. This attack mode is typically used to overpower webservers by exhausting system and application resources.
- `02` – *Synchronous flood*: Sends a TCP SYN packet to the target server/port and disconnects after the TCP handshake is completed. This attack can target arbitrary TCP-based services by overpowering the victims' network stack with connection requests.
- `03` – *Downloading flood*: A HTTP GET request to the webserver is performed and the full response is downloaded. This attack is carried out to exhaust the available bandwidth of the victim.
- `04` – *POST flood*: The bot sends HTTP POST requests to the victim's webserver. This attack is efficient against websites that expect data sent from the client, e.g. authorization forms in login pages. The requests generate a high load on the server with relatively few requests compared to the other attack modes.

The number of threads in the example attack is 50 and the timeout for requesting a new command is 60 seconds. Each thread carries out the attack specified by the attack mode. This multithreading is important for a successful HTTP attack when the attacked victim server gets slower with its responses due to the overwhelming number of concurrent requests.

During HTTP-based attacks, the bot uses a random HTTP user-agent header selected from a list of 65 user-agent strings with various combinations of device, operating system, and web browser. This feature is meant to impede the differentiation between an attacking bot and a legitimate client. The list includes exotic user-agents that are unusual to produce a lot of requests (e.g., *Nintendo Wii*, *Nokia Symbian* smartphones, or old software like *Microsoft Windows 95* in combination with *Internet Explorer 4*). With the knowledge of this specific list of user-agents and the noisy request frequency carried out by the bot during attacks, it is straightforward to identify attacking clients in access logs.

3 *DirtJumper* Botnet Monitoring

Based on this overview of *DirtJumper*, we now first describe the infrastructure we used to automatically monitor the commands sent by DDoS botnets of this particular malware. We then analyze the recorded C&C command data set and study server-side log files we were able to obtain by contacting victims of DDoS attacks.

3.1 Automated Monitoring of HTTP C&C

Figure 1 provides an overview of the system architecture and workflow of the processes used in capturing botnet

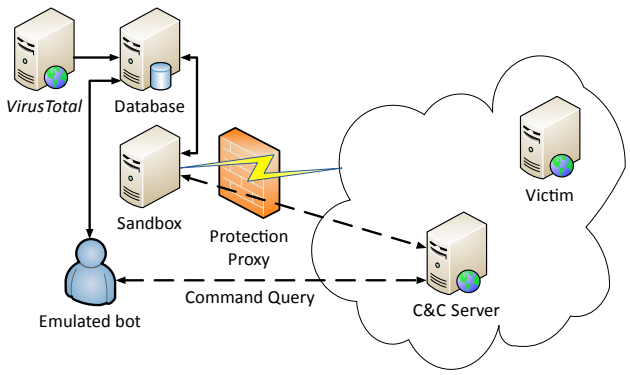

Figure 1: Architecture diagram

commands. As a starting point to automatically identify C&C servers of DDoS-related botnets, we implemented an interface to the *VirusTotal Malware Intelligence Service* [9] to regularly obtain recent malware samples based on a search for an AV detection name. A challenge trying to identify samples of a malware family is the difference and inconsistency between the AV detection names of the various vendors [1]. To solve this, we select a test set of malware samples that are manually verified to belong to a malware family and compare the detection rates and names of all vendors in the *VirusTotal* scanning service. The AV engine with the best identification results is selected as the reference detection for the malware download process.

The downloaded malware samples are queued into a behavior-based analysis system. This malware analysis system uses virtual machines with a clean *Microsoft Windows XP* operating system patched to *Service Pack 2*. The sandbox automatically attempts a deliberate infection from a clean reverted system state and runs for a timeout of 60 seconds. The virtual machines are connected to the Internet through a modified HTTP proxy server and resulting network traffic is captured in PCAP format. The proxy server only allows a single HTTP connection to each unique URL during the selected timeout and blocks subsequent connection attempts to prevent the infected virtual machine from participating in a DDoS attack. Malware analysis frameworks such as *Anubis* [2] or *BitBlaze* [20] could also be used, but we opted for this simple approach since we are mainly interested in the network traffic observed during the analysis.

The next step in the analysis phase is carried out by a script parsing the PCAP data. It automatically analyzes all HTTP connections from the network captures saved by the sandbox system. The script subsequently identifies connections to C&C servers based on the features learned in the previous, manual analysis of the botnet communication. The suspected URLs of C&C servers are then stored in the database. To automatically monitor the botnets found in the network analysis phase, we implemented an emulated bot client that performs HTTP POST requests every 5 minutes to communicate with the C&C servers and stores the commands it receives in the database to enable a later analysis.

Month	# Samples
Jan-2011	2
Feb-2011	12
Mar-2011	8
Apr-2011	17
May-2011	5
Jun-2011	5
Jul-2011	3
Aug-2011	7
Sep-2011	6
Oct-2011	35
Nov-2011	55
Dec-2011	180
Jan-2012	130

Table 1: *DirtJumper* samples first seen per month at *VirusTotal*.

3.2 Empirical Data Sets

Using our monitoring infrastructure, we conducted an experiment targeting the malware family *DirtJumper* (also known as *Ruskill*). Between October 2011 and January 2012, the system downloaded a total of 465 malware samples from *VirusTotal* [9] following the reference detection of *Microsoft* with their virus name *Dishigy.B*. Table 1 shows the number of samples with the according detection first seen at *VirusTotal* per month. We started our analysis with a set of 65 samples that were uploaded from January to September 2011. Starting in October, the automated sample download script regularly acquired new samples to feed the analysis stage.

The dynamic analysis stage identified 274 samples with network connections detected as *DirtJumper* C&C traffic from the traffic captures created by running the malware in our analysis system. A total of 68 unique C&C URLs were identified to be queried for commands according to the bot's communication scheme.

To gather the attack data set, we used an emulated bot script according to the C&C communication schema described in Section 2.2. The emulated DDoS bot queried the C&C servers every five minutes. Of the 68 suspected botnet C&C URLs, 35 servers responded at least once with a valid DirtJumper attack command to the requests of the emulated bot. During the analysis period, we observed commands to attack a total of 1,968 unique URL targets. Figure 2 shows the observed lifetime of the servers after initial discovery up to 24 days. Three C&C servers were online for a longer time and operated over

TLD	Country/Denotation	# Attacks
.ru	Russia	517
.com	*Commercial*	438
.net	*Network*	91
.org	*Organization*	35
.ua	Ukraine	24
.su	Soviet Union	23
.au	Australia	22
.biz	*Business*	20
.ai	Anguilla	20
.info	*Informational*	17

Table 2: Top ten top-level domains of DDoS victims.

the full time span of our monitoring evaluation. The average monitoring time of a *DirtJumper* C&C server in the data set was 406.52 hours (i.e., almost 17 days).

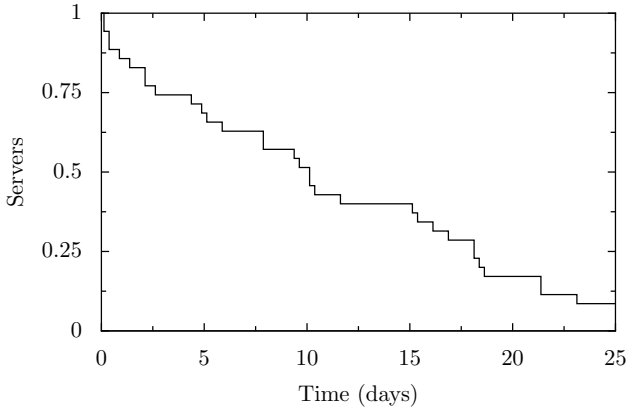

Figure 2: Observed lifetime for 35 C&C servers.

3.3 Analysis of Botnet Commands

Table 2 lists the top ten top-level domains of victim URLs in the obtained command set. The majority of victims are Russian domains, that relates to the popularity of the *DirtJumper* malware kit in Russian Underground forums as implied in Section 2.1.

To determine what kinds of services were targeted, we look up the categorization of attacked HTTP URLs using *Websense*'s classification technology. Out of 1707 attacks on web servers, 1011 URLs (59.2%) could be successfully categorized. Out of the 696 uncategorized URLs, 386 were attacks on IP addresses. Since reverse DNS lookups are unreliable and often do not contain valuable information to classify a given domain, we opted to not study these IP addresses further. Table 2 provides an overview of the category of DDoS victims.

Figure 3 shows how long the emulated bots received commands to attack each URL in percentages. Over one third of the DDoS attacks (33.5%) only last for up to one

Category	# Attacks
Shopping	215
Adult Material	111
Hacking	85
Business and Economy	78
Infrastructure	78
Gambling	75
Message Boards and Forums	75
Games	65
News and Media	64
Malicious Websites	40
Education	25
Entertainment	25
Government	20
Internet Communication	16
Advertisements	10
Freeware and Software Download	7
Blog and Personal Sites	7
Search Engines and Portals	6
Bot Networks	4
Social Networking	3
Computer Security	2
Unknown	696

Table 3: General website categories of DDoS victims.

hour. A total of 34 attacks lasted longer than ten days, with the longest DDoS at over 45 days and still ongoing at the end of the analyzed data set. A closer analysis revealed drops in the number of attacks after certain time spans (e.g., after 72 and 100 hours). This hints at *DDoS-for-rent* services described in Section 2.1 offering an attack with a specified length.

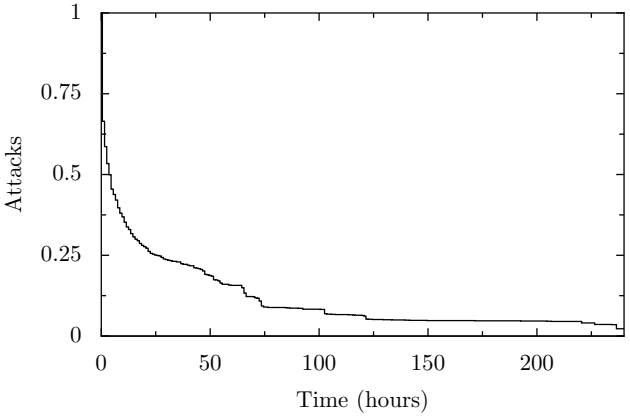

Figure 3: Distribution of attack length in hours.

Since a *DirtJumper* botnet can be used for DDoS attacks on arbitrary services relying on the network protocol TCP, we parsed the victim URLs to extract destination ports. Similar to a web browser, the *DirtJumper* bot

Port	Service	# Attacks	Attacks %
80	HTTP	1686	85.7%
3306	MySQL	49	2.5%
22	SSH	26	1.3%
8080	Alternate HTTP	22	1.1%
443	HTTPS	21	1.1%
21	FTP	21	1.1%
1723	PPTP	21	1.1%
25	SMTP	4	0.2%
Other	*Unknown*	118	6.0%

Table 4: Overview of attacked server port.

Command	Name	Rate (%)
01	HTTP flood	48.7%
02	Synchronous flood	14.8%
03	Downloading flood	2.6%
04	POST flood	34.0%

Table 5: Overview of observed attack type.

interprets the absence of a port in the URL as destination port 80, the standard port for HTTP-based services. Table 4 shows the distribution of the attacked ports and well-known services associated with these ports. While most attacks are targeting web servers, the attackers also try to disrupt database, remote access, *Virtual Private Network* (VPN), and mail services. Furthermore 108 attacks target custom port numbers that could not be mapped to services using well-known standard ports.

As described in Section 2.2, the *DirtJumper* bot has four different attack modes. Table 5 shows the distribution of the mode picked by the botmasters for attack commands. The percentage of *Synchronous flood* attack commands certainly is consistent with the percentage of attacks on non-HTTP services shown in Table 4.

The two remaining arguments sent with each command to a bot are a timeout to the next command query and the number of threads used in the attack. The average number of threads found in the command data set was 185.0, meaning the attackers make extensible use of the multithreading implemented by the *DirtJumper* bot. The average C&C query timeout value was 497.1 seconds (∼8.3 minutes) and the maximum timeout value was 6095 seconds (∼101.6 minutes).

3.4 Analysis of Server-side Victim Logs

During the analysis of the attack data set we noticed attacks on two popular webpages with an IT security background: `krebsonsecurity.com` and `virustotal.com`, the same malware scanning service we used to identify and acquire our malware sample set. By contacting the victims, we were able to obtain the webserver access logs of the attacked servers that we subsequently analyzed to identify attacking *DirtJumper* bots. This provides another vantage point at DDoS attacks and enables us to learn how such attacks are perceived at the victim's endhost. We were able to identify the IP addresses of bots by filtering clients with many requests and a HTTP user-agent used by the bot (see Section 2.2 for details).

Country	# Bot IPs	Attacks %
India	4152	19.5%
Thailand	3026	14.2%
Indonesia	2237	10.5%
Vietnam	1660	7.8%
Pakistan	1580	7.4%
Kazakhstan	973	4.6%
Malaysia	830	3.9%
Mexico	732	3.4%
Philippines	614	2.9%
Egypt	503	2.4%

Table 6: Top ten bot origin countries participating in observed attack on *krebsonsecurity.com* (KOS)

Country	# Bot IPs	Attacks %
Russia	11361	89.6%
Belarus	240	1.9%
Ukraine	211	1.7%
Czech Republic	183	1.4%
Spain	182	1.4%
Germany	111	0.9%
Kazakhstan	62	0.5%
USA	47	0.4%
France	37	0.3%
Poland	29	0.2%

Table 7: Top ten bot origin countries participating in observed attack on *virustotal.com* (VT)

The first data set of an attack was sent to us by the independent journalist Brian Krebs, who is specializing in investigative reports of cybercrime activities all over the world. The webserver hosting his blog `krebsonsecurity.com` (KOS) was attacked in late November 2011 by a *DirtJumper* botnet. The webserver logs show 21,293 unique IP addresses participating in the attack. Table 6 depicts the top ten countries the bot's IP addresses were mapped to.

The second data set was sent to us by contacts at the malware scanning service *VirusTotal* (VT). The service was flooded with requests for over 14 days in late December 2011 by a total of 12,686 unique IP addresses and Table 7 provides the country distribution of the attacking botnet.

Although we analyzed only two data sets of victim logs, we can obtain some first, preliminary result. In both attacks, several thousand infected machines sent concurrent requests to the victim's server and caused connectivity problems with this flood of requests. We observe that the country distribution is far from homogeneous in both attacks. In fact, most bots related to the attack against VT were based in Russia, while the attack against KOS was mainly coming from South Asia (and being more evenly distributed).

4 Discussion

Next we briefly discuss the results of our analysis of the DDoS botnet data sets and speculate on the common motives attackers have to perform DDoS attacks.

4.1 Estimations of DDoS Impact.

The size of the botnets found and the attack command analysis in Section 3.3 give an idea how powerful the DDoS attacks can strike targets. The average number of 185 DoS threads used by the attackers can lead to a multithreaded attack with near the full available bandwidth between the bot and the victim. If several thousand bots simultaneously send such a large number of requests, small- or even medium-sized servers often cannot cope with this number of concurrent requests.

While setting up the DDoS botnet monitoring system, we considered adding a component to query the attacked website to find out whether or not an attack was successful. We decided against implementing and using such a component for several reasons. First and foremost, the success of a *Denial-of-service* is subjective to the client requesting the service. The large number of uncontrollable factors impacting whether a client can still get a response from an attacked server makes the results of sampling unreliable. An example of these factors is the load-balancing technology used for popular websites. A website that uses several webservers all around the world to serve the local demand could be overwhelmed by a botnet DDoS attack on one continent where the main number of bots originates from, while still being able to fully operate for users on other continents. Second, the requests from a monitoring component would aggravate the situation for the victim. Even though a single query might appear insignificant compared to the simultaneous botnet attack, it still adds to the overall server overload.

4.2 Motives for DDoS Attacks.

While investigating Underground forums offering DDoS tools or services, we found five general motives for DDoS attacks:

- blackmail service operators
- disrupting the competition
- disrupting adversaries
- manipulating services
- political protests

Taking into account the categories of attacked sites listed in Section 3.3, we can make an educated guess regarding the likely motives for the attacks on the respective services. For example, websites in the categories *Shopping* and *Adult* entertainment are presumably targeted to extort money from the underlying business. Likewise examples can be found in the *Business and Economy* category. In December 2011, the Australian online brokerage website `etrade.com.au` was reportedly attacked and successfully disrupted by a *DirtJumper* botnet to blackmail the operators [24]. After discovering the story, we searched the attack data set analyzed in Section 3.3 and located the corresponding commands sent from a C&C server over five days in late December 2011. These extortion attacks certainly take advantage of the fact that many companies nowadays rely and focus on web services for their business and even brief downtimes can cause severe financial disadvantages.

The attacks targeting adversaries of the Underground show how criminals use *DDoS* when they fear for their business, as seen in the attacks against *KrebsOnSecurity* and *VirusTotal*. Krebs reported in several blog posts on *DDoS* attacks and *DDoS for Rent* services by criminals, which supposedly courted their resentment. The *VirusTotal* service is a free and publicly available online virus scanner that causes problems for malware authors by closely working with the anti-virus industry and sharing uploaded malware samples to improve detection rates. Every successful DDoS against the service prevents users from uploading suspicious files and, therefore, might help attackers buy time before anti-virus vendors catch their creations.

Another interesting find are the DDoS attacks used to disrupt direct competition on the Underground market. The victim category analysis shows a considerable number of attacks against *Hacking* forums, as well as, direct attacks against the malware distribution (*Malicious Websites*) and infrastructure (*Botnets*). Observing the commands of DDoS botnets might thus also help to uncover instances of other malicious activities on the Internet.

The categorization shows attacks on 20 websites classified as *Government*, which equals just over one percent of the victim URLs. The motivation behind these attacks is likely protest, but we could not find any claims of so-called hacktivist groups to target the involved victims. This leads to the assumption that *DirtJumper* botnets are not substantially involved in politically motivated attacks.

5 Conclusion

In this paper, we presented an empirical study of DDoS botnets. More specifically, we studied 35 C&C servers related to the *DirtJumper* family of malware. We observed almost two thousand DDoS attacks and performed an analysis regarding various aspects that enables us to obtain insights into such attacks. Furthermore, we complemented this C&C-centric vantage point with a brief analysis of two DDoS attacks from the victim's side.

Acknowledgements: This work was in part supported by the German Federal Ministry of Education and Research under BMBF Grant 01BY1111 (*MoBE*).

References

[1] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario. Automated Classification and Analysis of Internet Malware. In *Symposium on Recent Advances in Intrusion Detection (RAID)*, 2007.

[2] U. Bayer, A. Moser, C. Kruegel, and E. Kirda. Dynamic Analysis of Malicious Code. *Journal in Computer Virology*, 2(1), 2006.

[3] J. Caballero, C. Grier, C. Kreibich, and V. Paxson. Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *USENIX Security Symposium*, 2011.

[4] R. K. C. Chang. Defending Against Flooding-based Distributed Denial-of-Service Attacks: A Tutorial. *IEEE Communications Magazine*, 40, 2002.

[5] C. Y. Cho, J. Caballero, C. Grier, V. Paxson, and D. Song. Insights from the Inside: A View of Botnet Management from Infiltration. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.

[6] E. Cooke, F. Jahanian, and D. McPherson. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In *USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2005.

[7] D. Dittrich. The DoS Project's 'trinoo' distributed denial of service attack tool. `http://staff.washington.edu/dittrich/misc/trinoo.analysis`, Oct. 1999.

[8] F. C. Freiling, T. Holz, and G. Wicherski. Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks. In *European Symposium on Research in Computer Security (ESORICS)*, 2005.

[9] Hispasec Sistemas S.L. VirusTotal Malware Intelligence Service. `https://www.virustotal.com`, 2011.

[10] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. C. Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[11] C. Jin, H. Wang, and K. G. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *ACM Conference on Computer and Communications Security (CCS)*, 2003.

[12] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds. In *USENIX Symposium on Networked Systems Design and Implementation*, 2005.

[13] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. On the Spam Campaign Trail. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[14] J. Mirkovic and P. L. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *Computer Communication Review*, 34(2), 2004.

[15] C. Nunnery, G. Sinclair, and B. B. Kang. Tumbling Down the Rabbit Hole: Exploring the Idiosyncrasies of Botmaster Systems in a Multi-Tier Botnet Infrastructure. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.

[16] V. Paxson. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. *Computer Communication Review*, 31, 2001.

[17] P. Porras, H. Saïdi, and V. Yegneswaran. A Foray Into Conficker's Logic and Rendezvous Points. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2009.

[18] M. A. Rajab, L. Ballard, P. Mavrommatis, N. Provos, and X. Zhao. The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.

[19] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Internet Measurement Conference*, 2006.

[20] D. Song, D. Brumley, H. Yin, J. Caballero, I. Jager, M. G. Kang, Z. Liang, N. James, P. Poosankam, and P. Saxena. BitBlaze: A New Approach to Computer Security via Binary Analysis. In *International Conference on Information Systems Security (ICISS)*, 2008.

[21] D. X. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. In *IEEE Conference on Computer Communications*, 2001.

[22] B. Stone-Gross, M. Cova, L. Cavallaro, R. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[23] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2011.

[24] The Sydney Morning Herald. Dirty dealings and DirtJumper. `http://bit.ly/A0CVYP`, 2012.

[25] A. Yaar, A. Perrig, and D. Song. Pi: A Path Identification Mechanism to Defend Against DDoS Attacks. In *IEEE Symposium on Security and Privacy*, 2003.