DDOS TROJAN: A MALICIOUS CONCEPT THAT CONQUERED THE ELF FORMAT

Peter Kálnai & Jaromír Hořejší
Avast Software, Czech Republic

Email {kalnai, horejsi}@avast.com

ABSTRACT

DDoS threats have been around ever since the Internet took over half of global communications, posing the real problem of denial of access to online service providers. Recently, a new trend has emerged in non-*Windows* DDoS attacks that has been induced by code availability, lack of security and an abundance of resources. The attack infrastructure has undergone significant changes in structure, function and complexity. Malware has evolved into complex and relatively sophisticated pieces of code, employing compression, advanced encryption and even rootkit capabilities. Targeted machines are those running systems supporting the ELF format – anything from desktops and servers to IoT devices such as routers or digital video recorders (DVRs) could be at risk.

In this paper, we will look at the current state of DDoS trojans forming covert botnets on unsuspecting systems. A technical analysis of the most important malware families will be provided, with a specific focus on infection methods, dynamic behaviour, C&C communication, obfuscation techniques, advanced methods of persistence and stealth, and elimination of rivals. We will study cybercriminals' behaviour and introduce their operation tools, including vulnerability scanners, brute-forcers, bot builders and C&C panels. In many cases, it's unnecessary to apply reverse engineering within the analysis – the original source codes are indexed in public search engines and their customization is a subject of monetization. Finally, we will discuss tracking methods and techniques and reveal the targets of these attacks.

1. INTRODUCTION

Today, both desktop users and organizations face various types of cyber attacks including credential theft, spam distribution, click fraud and denial of services (DoS). These are all usually performed via a series of zombie computers in a malicious botnet controlled by cybercriminals. Distributed DoS (DDoS) attacks are carried out using various methods such as volumetric flooding, slow HTTP attacks or TCP protocol misuse. A DNS amplification is an example of volumetric flooding that has become popular recently. Certain trojans for the *Windows* platform with resources containing Chinese symbols have a long tradition of use in this type of attack and lack other spying features that trojans often possess (cf. [1]).

Implementations of flooding procedures have been available for a long time on Chinese source-code sharing services, especially in C/C++. Until recently, the shared sources were usually compiled as DDoS tools exclusively for the *Windows* platform. Crossing to another platform like *Linux* is not a complicated process though, and we have observed that many of these tools have been ported. These tools are not necessarily invasive trojan-type malware. However, they come with an additional framework (often on a different platform from the flooding tools) that provides an attacker with the possibility of forming a malicious botnet. The framework consists of C&C panels, bot builders, installation scripts, port scanners, SSH brute-forcers, etc. In our paper we provide a survey of current *Linux* DDoS bots and we sketch a wider context of usage of the above-mentioned components. This topic has been covered by many independent security researchers, e.g. the *MalwareMustDie!* group summarised their series of blogposts in [2]. The authors presented its early development in [21].

2. COMMON CHARACTERISTICS

2.1 About the ELF format

ELF is short for 'Executable and Linkable Format', which is a common standard format for executables under Unix systems. The instruction set architecture (ISA) of the binary is stored at the offset e_machine, and we refer to this parameter using the prefix 'EM_'. Unix systems power not only desktop and server machines, but are also the main operating systems for embedded devices. Although *Intel 80386* (EM_386) and *AMD64* (EM_x86_64) ISAs are the most common for desktops and servers, the situation for embedded devices is completely different – the devices may run on different ISAs, such as EM_ARM or EM_MIPS. Cross-compilation of malicious binaries to the latter platforms is done with the aim of extending the group of potential victims to target a wider range of devices.

2.2 Static properties

We found that debugging symbols were often not stripped from ELF executables and thus revealed a lot about their functionality, especially the procedure names. The unstripped binaries listed in the Appendix were chosen intentionally so that we could review their features easily. At the same time, plain ELF binaries were occasionally encapsulated with the well-known UPX compression [4], which produced a certain level of polymorphism for the distributors. There were cases when packed executables were altered in such a way that the official tool did not decompress them (e.g. some chosen samples from the Sotdas and MrBlack families referred to in the Appendix). These modifications targeted the UPX header, which is located at the tail of the file. The changed parameters were the UPX magic string, various checksums, file sizes or decompression methods. Dynamic behaviour was not affected.

Infected embedded devices do not often hide the presence of the malware – debug information might be displayed directly on the standard console output.

2.3 Autostart

All programs and tools below are involved in malicious activities, but in a strict sense we consider a DDoS trojan as a DDoS tool with an autostart feature. There are more common methods for bots to handle their persistence. This is done by creating symbolic links or executable scripts in the following directories:

- (AS1) /etc/init.d/ – the central repository of all startup scripts.
- (AS2) /etc/cron.<S>/, where S is a period (hourly, daily, monthly, weekly). Additionally, a service might be added to run in the cron (/etc/crontab).
- (AS3) /etc/rc<N>.d/, where N is the run level indicator (0–6). Alternatively, the desired path can be added to

/etc/rc.local to run it once at boot time before all other scripts.

3. CYBERCRIMINALS' OPERATION TOOLS

Distribution starts with either an automated SSH brute-forcing of various *Windows* and *Linux* servers or vulnerability scanning and exploiting using the hacking tools and password lists mentioned below. If successful, flooding tools are downloaded and installed onto the hacked servers.

The installation process can consist of several steps:

- determining susceptibility to known vulnerabilities of the system
- killing competing time-consuming processes
- establishing autostart
- disabling firewalls.

We found that it was not unusual for flooding and control components to be offered freely as web server stress tests, with a warning that discourages any malicious use. However, authors looking for monetization of their tools offer customization of attack preferences and their control framework in return for a fee.

3.1 Bot builders and C&C panels

According to the type of machines that botmasters use to build their botnets, particular C&C panels and bot builders are chosen. They might be restricted to MIPS bots (e.g. MrBlack) or more heterogeneous with support for *Linux*, *Windows* and *FreeBSD* (e.g. Elknot). A bot builder is designed to set up bot details easily (e.g. the IP address of the C&C, port number, encryption keys, etc.). The builder generates the final binary from a binary template, and after execution it is displayed in the control panel. The control panels are in the form of *Windows* executables.

Figure 1: Example of Chicken Builder for Elknot.

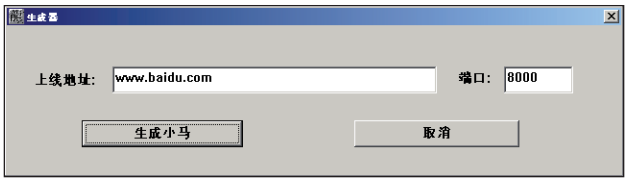

Figure 2: Example of MrBlack Builder.

The acquired builders are written and run under *Windows* OS, although they may produce binaries for both *Windows* and *Linux*. A few examples are as follows:

- Chicken Builder (see Figure 1)
- Text-Box Builder & Manager
- MrBlack Builder (see Figure 2)

The existence of bot-building tools is consistent with the enormous number of existing samples that differ only in settings such as C&C domains.

3.2 HFS listings

A good insight into the distribution chain was provided by file listings displayed as the web content of an HTTP file server (an HFS listing) that was established on a possibly compromised machine. The listings contained various groups of files, starting with port scanners, vulnerability scanners, login/password brute-forcers, password lists, lists of targeted IPs and SSH clients, and ended with the proper flooding tools, bot builders and C&C panels. Sometimes we even noticed text files with IP addresses of servers followed by the default names and passwords. These servers were clearly using the default credentials and were therefore accessible to anyone knowing the name/password combination. Additional crucial information provided by the HTTP file servers was the exact number of downloads of each file, giving us a chance to estimate the size of botnets.

Figure 3: An HFS panel used for distribution of malware – the number of downloads is shown in the right column.

3.3 Brute-forcers and vulnerability scanners

Figure 4 shows an SSH brute-forcer, where attackers can enter a list of IP addresses and a password list, as well as specifying other parameters of the attack.

Figure 5 shows a vulnerability scanner for *Apache Struts*, which was found together with Chinaz samples.

4. DDOS TROJANS IN THE ELF MALWARE SPACE

It should not come as a surprise that ELF malware is not the same as its *Windows* counterpart. This is due to the fact that

Figure 4: An SSH brute-forcer. Attackers enter an IP list and password list and specify other parameters of the attack.

Figure 5: A vulnerability scanner for Apache Struts found together with Chinaz samples.

most *Windows* malware runs on a machine where a user performs personal actions, so info-stealers and banking trojans make sense. ELF malware often targets devices that are not accessed by a real user. Therefore, it makes sense to misuse the device's power for tasks like denial of service, bitcoin mining or proxy networking and anonymization. Figure 6 sketches the total ELF malware space based on *Avast*'s internal data, where the nodes represent particular malware families proportional to the number of unique samples and the edges are present if the corresponding families share a signature. The picture has been manipulated to omit very old viruses, potentially unwanted applications,

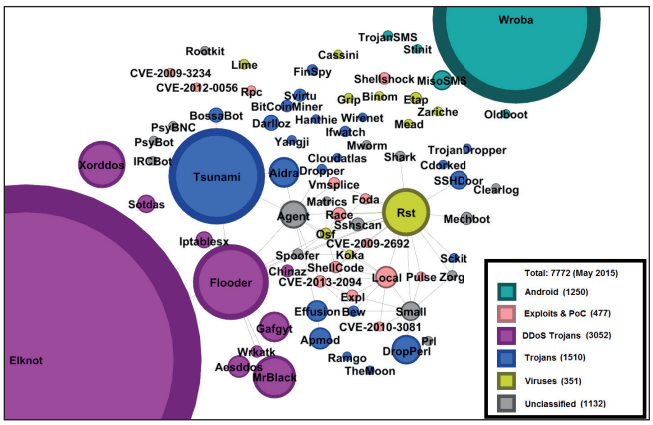

Figure 6: ELF malware space.

and nodes with fewer than 10 unique hashes for the sake of leaving only relevant and interesting data.

Based on the similarity of features we tried to group all the bots into several clusters. Most of them also exist as *Windows* variants, but those will not be discussed.

4.1 Elknot/Setag

With the highest incidence among any ELF malware family (~1400 unique samples), Elknot/Setag is among the most popular malware distributed by attackers. There are two main variants in this family, both of which were written in the modular C++ language. Bots of this type run on *Windows* x86/x64, *Linux* x86/x64 and *FreeBSD* systems. Although the variants differ in complexity, we grouped them into one family because their *Windows* versions have significantly similar debug info. They also have a very similar command grammar. A detailed analysis was provided by *Kaspersky*'s Mikhail Kuzin in [5].

4.1.1 Simple variant

This variant usually comes as an on-the-fly patcher of an embedded DDoS tool. Its purpose is to replace itself with a dropped bot that has the C&C server and the port number carried from the patcher. These data are encrypted with a simple substitution cipher based on adding one from characters on odd positions and subtracting one from characters on even positions.

The tool itself is characterized by the presence of the files fake.cfg or xmit.ini, which contain information such as the attack status, the ranges of outgoing IP addresses and ports, or a domain name for a DNS flood. All network communication is a part of the CNetBase class. The commands supported are ReadTask, StopTask, ReadFake and DestroySocket, and are implemented in the main CManager class.

4.1.2 Variant Setag

This more sophisticated trojan-type bot usually contains the character strings 'Bill' and 'Gates'. It recognizes four types of execution: MainBackdoor, MainSystool, MainMonitor and MainBeikong.

- *Type MainBackdoor:* A bot running this type of execution is called as the /usr/bin/bsd-port/getty file. At first it decrypts a hard-coded configuration using RSA-1024 (the configuration settings are computed as a solution to the equation P ^ Q % N, where P and Q are primes and N is modulus). The SetAutoStart procedure creates the selinux script in (AS1) and multiple links under the name S99selinux in (AS3). The HandleSystools method of the CSystool class replaces the system tools /bin/netstat, /bin/lsof, /bin/ps, /usr/bin/netstat, /usr/bin/lsof, /usr/bin/ps, /usr/sbin/netstat, /usr/sbin/lsof and /usr/sbin/ps with copies of itself and backs up the original files into the newly created /usr/bin/dpkg/ directory. The proper attack is realized in the MainProcess method of the main CManager class. A list of the IP addresses of DNS servers misused for DNS amplification attacks are stored in the /usr/lib/libamplify.so file. The attacking classes are as follows: CAttackIcmp, CAttackSyn, CAttackUdp, CAttackAmp (DNS amplification), CAttackCC, CAttackDns, CAttackPrx, CAttackCompress, CTcpAttack, CAttackCc, CAttackIe and CAttackTns.

- *Type MainSystool:* A bot running this type of execution must be named as one of the systools. This type may be under development because, currently, execution of these replaced commands starts a copy of the trojan but does not ensure its autostart or an alteration of outputs.
- *Type MainMonitor:* A bot running this type of execution is executed as the /usr/bin/.sshd file. It spawns an infinite thread which monitors the existence of an instance of the trojan and handles a reinstallation if needed. The PID of a monitored process is loaded from /tmp/gates.lock. If no process with the given PID exists, then the trojan is installed and started again.
- *Type MainBeikong*: The features of this type of execution are very similar to those of MainBackdoor. This type is run if the bot has a different name from the three mentioned. The autostart is achieved by the DbSecuritySpt script via (AS1) and by multiple links under the name S97DbSecuritySpt via (AS3). In this mode, the trojan checks whether another instance of malware is running in the system (via the presence of *.lock files containing PIDs). If so, then it terminates all of them, deletes all .lock files and restarts. This mode is probably used for updates or reinstallation of the trojan.

The C&C panel is called NFarm and its executable is called xitele, which translates as 'Hitler' in Chinese (see Figure 7).

4.2 MrBlack

This family involves minimalistic DDoS tools characterized by the main thread _ConnectServer with a subroutine DealwithDDoS that handles the attack. The binaries usually contain character strings such as 'Hacker', 'Mr. Black', 'VERS0NEX' or a typo 'connnect'. The support of *Linux* architectures is wide (EM_386, EM_x86_64, EM_ARM and EM_MIPS) and compression with UPX is not exceptional. Additional threads could be spawned for collecting CPU and network statistics and reporting them to a C&C server. A detailed analysis was published by *Prolexic* [6].

4.2.1 Aesddos

Aesddos is a trojan that extends the core functionality provided by MrBlack with C&C communication encrypted with AES. Persistence is realized at the beginning of the execution in the autoboot procedure by stream editing the files /etc/init.d/boot.local (AS1) and /etc/rc.local (AS3). The attack itself is performed via two main threads, backdoorA and backdoorM. The trojan supports the same *Linux* architectures as MrBlack, but the EM_ x86_64 variant was not observed. A detailed analysis of the MIPS variant is provided by M.J. Bohio [7].

4.2.2 WrkAtk

This is another trojan subfamily that strongly resembles Aesddos, with two main threads called DoubleDoMain1 and

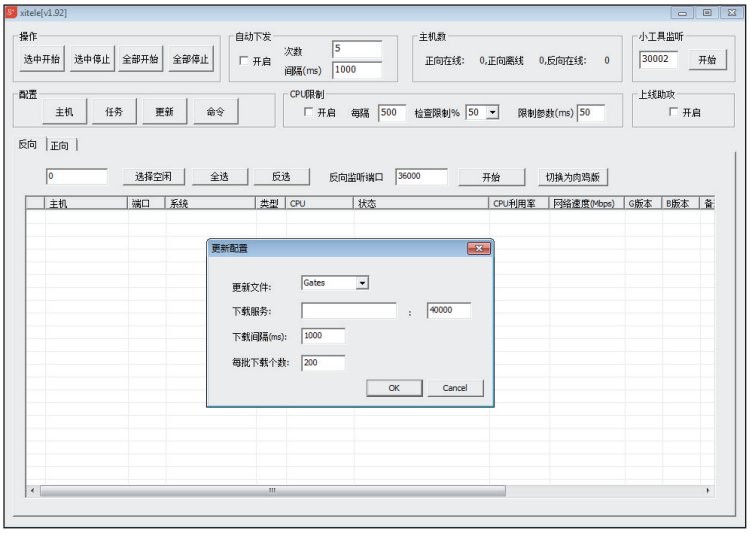

Figure 7: Control panel for Elknot family, called xitele.

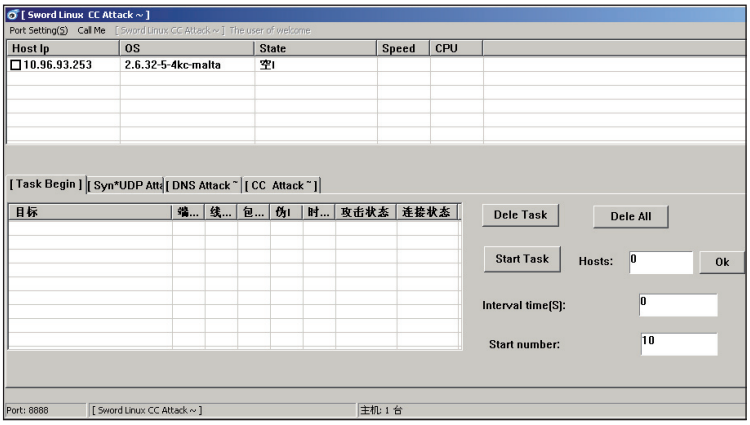

Figure 8: Control panel for MrBlack called sword.exe with one connected bot (MIPS).

DoubleDoMain2. Its name is derived from the thread AttackWorker, which is responsible for calling the DealwithDDoS subroutine with appropriate parameters. Autostart is achieved using the insert_reboot procedure and is based on the same principle as in the case of Aesddos. The already_running procedure checks that the process is the only instance of the trojan, and the test is done by locking the file /var/run/dos32.pid. The configuration file dosset.dtdb is an additional IoC.

4.3 Iptablesx

It was observed that a DDoS trojan has been spread by exploiting vulnerable *Apache Tomcat*, *Struts* or *Elasticsearch* software. The functionality of this piece of malware is divided into two parts: an initial binary is responsible for establishing persistence, eliminating an extensive list of rivals (stored in kill.txt or fuckopen.txt) and downloading an additional bot with implemented attack routines. IoCs list a binary /usr/bin/btdaemon, a process .flush, or S90bluetooth symbolic links.

The proper DDoS bot is written in C and is downloaded as getsetup.rar. Execution parameters are stored in the global variable g_mainsrvinfo of a struct type MAINSRVINFO with entries such as srvver, mainhostip, udpport or Mainisrun. A self-installation is performed into the /boot/ directory with an autostart script called IptabLes/IptabLex in (AS1). The bot then establishes communication with its C&C server. The initial packet contains the memory and CPU statistics of the compromised machine. Once the connection is established and the initial packet has been sent, the bot awaits commands from its control server. The most important DDoS commands are: setlocalip, setrandomip, updatepath, updatesrv, DeleteTask and DeleteAllTask. Observed types of attacks include volumetric SYN_Flood and DNS_Flood. Detailed analyses have been provided by *Prolexic* [8] and *MalwareMustDie!* [9].

4.4 Sotdas

This family is characterized by the presence of the strings 'g_nIsStopDDOS', 'DOSSTAT' or '# chkconfig: 2345 77 37'. The encryption of particular strings is done in a similar manner to that in which it is done in the Elknot family, but the period of substitution is 3 instead of 2. A self-installation takes care of multiple copies as /etc/.zl and /tmp/.lz<digits> files, where digits are derived from actual time. An autostart script is called .zl in (AS1) and symbolic links are called S77.zl in (AS3) (run levels Halt, Single-user mode and Reboot are omitted). The closefirewall procedure clears the environment for the trojan by disabling network filters (by stopping SuSEfirewall2 and disabling ufw) and terminating competing DDoS processes (obviously targeting IoCs of Iptablesx). The related binaries are often compressed with modified UPX. A detailed analysis has been published by *Dr.Web* [10].

4.5 Gafgyt

This is an alternative name for the Lizard Stresser DDoS tool advertised by the Lizard Squad group [11]. It is capable of infecting a variety of devices running on *Linux*. In-the-wild usage of the Bash Bug, a.k.a Shellshock vulnerability (CVE-2014-6271), was observed to install Gafgyt's binaries onto a compromised system [12]. The source code of this IRC bot was leaked in January 2015. The support of architectures is the widest as the installation script downloads cross-compiled bots for EM_386, EM_x86_64, EM_SPARC, EM_PPC, EM_SH, EM_ARM, EM_MIPS and EM_68K. Examples of implemented commands together with an explanation for each are shown in Table 1.

PING	Reply PONG to the server
GETLOCALIP	Get victim's IP
SCANNER	ON | OFF; attempts to login to IPs from a generated list and to install itself if successful
UDP	UDP flooding
TCP	TCP flooding
DNS	DNS amplification
KILLATTK	Stop DDoS attack
LOLNOGTFO	Stop the backdoor

Table 1: Some commands implemented in Gafgyt.

Detailed analyses have been published by *Dr.Web* [13] and *Alienvault* [14].

4.6 Xorddos

This cluster was first discovered in September 2014 [15] and later reported multiple times [16–18]. Its infection vector starts with SSH brute force attacks for the sake of running an installation script under the root user. The script customizes the installation process and contains procedures like main, check, compiler, uncompress, setup, generate, upload, checkbuild, etc. and variables like __host_32__, __host_64__, __kernel__, __remote__, etc. Three requests are issued to hard-coded C&C servers: the initial GET with an MD5-hashed string containing the name of the kernel version; a GET query with the parameters of a customized binary such as rootkit version and a list of the bot's C&Cs; and the final GET request of a compiled binary. The rootkit component with a complicated server-assisted installation is where Xorddos differs from all the other *Linux* trojans. Due to the nature of the *Linux* operating system, knowledge of kernel headers is crucial for loading any kernel module in the victim's machine. In case the hash of the kernel version is unknown to the server, the system's kernel headers are uploaded via a custom uploader and a new rootkit-equipped trojan can be delivered. The rootkit component is based on the open-source Suterusu rootkit, which is also available on *Github* [19].

The binary itself is copied into /boot/ under a random name. The name is then used to autostart the binary via (AS1), another script called udev.sh or cron.sh via (AS2), and symblic links called S90%s (where %s is randomly substituted via (AS3). Moreover, a victim might observe another instance of a randomly named binary in the /boot/ constantly being deleted and created again. This behaviour avoids the need to find a single constant process and to apply the kill command.

The configuration file of the rootkit contains four categories of lists: md5, denyip, filename and rmfile, which mean killing a running process based on its checksum (even though the

Figure 9: Elimination of rivals of Xorddos.

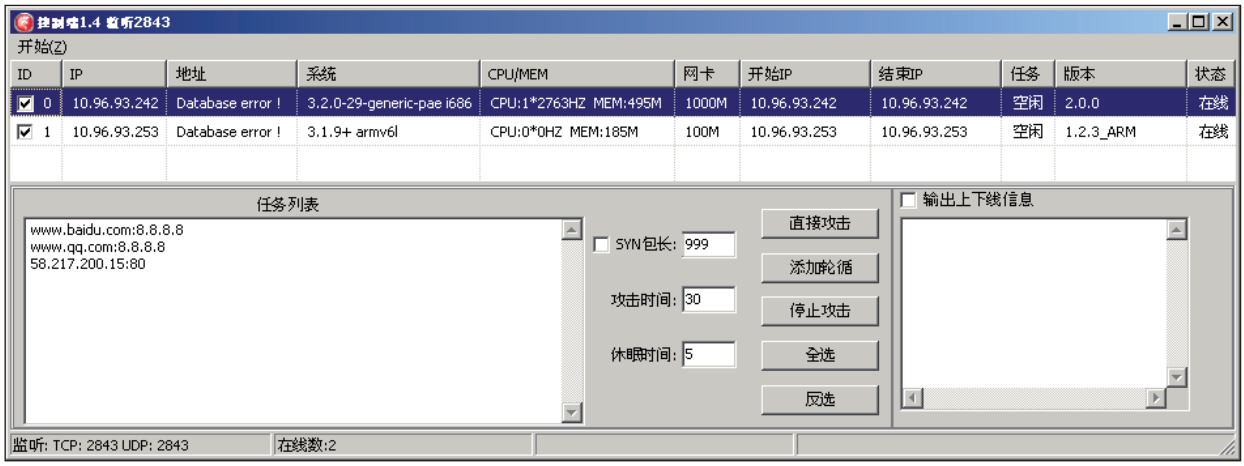

Figure 10: Control Panel for Xorddos with two connected bots, one EM_386, the second EM_ARM.

variable is called md5, the internal implementation is CRC32), on the active communication with an IP from the list, on a filename, and finally removing a file with a specified name. Figure 9 shows part of the config file; marked filenames denote usual competitors such as Elknot, MrBlack, Sotdas or Iptablesx.

The C&C communication is encrypted in both directions with the same hard-coded XOR key (*BB2FA36AAA9541F0*) – which inspired the trojan's name.

4.7 Chinaz

This is a collection of DDoS tools with core flooding functionality borrowed from a simple project, DDosClient, available publicly on *Github* (last update in March 2015). The commands that are supported are at least COMMAND_DDOS_ATTACK and COMMAND_DDOS_STOP, and the attack arsenal lists at least methods including ATTACK_SYN, ATTACK_UDP, ATTACK_ICMP, ATTACK_DNS1 and ATTACK_DNS2. The other character strings are 'DDos ATTACE!', 'SecurtDoor', 'MK32' and 'MK64'. Variants have been observed for multiple platforms: EM_386, EM_x86_64 and EM_MIPS. Samples are often packed by the unmodified UPX tool.

5. STATISTICS AND VICTIM PREFERENCES

The number of bots running on *Linux* platforms was not expected to be high. The size of a botnet could be estimated by data from HFS panels running on different machines. Distributed malicious binaries displayed tens, hundreds, occasionally thousands of downloads. However, that value was just an upper bound as bots might have been reinstalled repeatedly on victims' machines. The situation was similar for botnets of Gafgyt, as [11] revealed.

Regarding the volume of attacks, the reports [8] and [6] state peaks of 30.10 Gbps/6.75 Mpps for Iptablesx respectively, with 70 Gbps/28 Mpps for MrBlack. A DDoS attack performed by one victim of Xorddos reported in [20] lasted three hours and the amount of data averaged 6.63 Mbps/70 Kbps.

By knowing the communication protocol and the command grammar it is possible to harmlessly monitor the C&C activity and to log the targets of DDoS attacks. This was applied by a series of Python scripts for the Elknot/Setag family [21]. We have tried a similar approach for the Xorddos family. Among the observed victims were especially online gaming sites, e-commerce shops, online casinos, etc., all of which belonged to Chinese, American or Canadian IP ranges. These services had both the nature of small or medium-sized local businesses and services with a huge anti-DDoS infrastructure. The success of a flooding attack on the unprotected ones was directly observed in terms of the unreachability of a service shortly after the attack command had been issued.

Targeted IP addresses in the United States or Canada involved infrastructure belonging to *Google Cloud* [20], *CNSERVERS*

LLC, *Global Flag* (hosting game servers like *Counter Strike* or *Day of Defeat*), *CloudFlare*; *Sharktech*; *OVH Hosting*; *Microsoft Hosting*, *Amazon Cloud*; *Akamai Technologies*, etc.

What all these services have in common is that their functionality and profitability depend on their ability to stay online and available for customers' requests. When some of the above-mentioned websites suffer a heavy DDoS attack, they become unreachable to their potential users and/or customers. Any time the service is not available causes a financial loss. Therefore the reason for attacking the above-mentioned type of websites is financial – the potential profit is probably based on the extortion of ransom payments.

6. CONCLUSION

The number of (trojanized) DDoS tools has escalated quickly since the end of 2013. The potential in devices with undeveloped or no security measures is very attractive for attackers to build botnets. We can also see a variety of projects that wrap the core flooding features. We can expect current bots to be enhanced and even more families to arise in the near future, potentially sharing chunks of already known bots as a consequence of a mixture of available source codes. It's hard to predict whether binaries will display an increased level of polymorphism than just a (modified) UPX compression. For the time being, it seems that malware distributors do not care about AV detection. The security community has started to pay significant attention to these threats.

ACKNOWLEDGEMENTS

We would like to thank @benkow_, Christian Rebischke (@Sh1bumi), @threat_inc and Lin Song (University of Iowa) for sharing information and samples. Moreover, we thank the following researchers for openly sharing their threat intelligence: @MalwareMustDie, @TekDefense and @da_667.

REFERENCES

[1] Edwards, Nazario. J. A Survey of Contemporary Chinese DDoS Malware, Virus Bulletin 2011 (Barcelona).

[2] MalwareMustDie! Tango down report of OP China ELF DDoS'er. September 2014. http://blog.malwaremustdie.org/2014/09/tango-down-report-of-op-china-elf-ddoser.html.

[3] Kálnai P.; Hořejší J. Chinese Chicken: Multiplatform DDoS botnets, https://www.botconf.eu/wp-content/uploads/2014/12/2014-2.10-Chinese-Chicken-Multiplatform-DDoS-Botnets.pdf.

[4] UPX: the Ultimate Packer for eXecutables. http://upx.sourceforge.net/.

[5] Kuzin, M. Versatile DDoS Trojan for Linux. July 2014. https://securelist.com/analysis/publications/64361/versatile-ddos-trojan-for-linux/.

[6] Prolexic. Spike DDoS Toolkit. October 2014. http://www.prolexic.com/kcresources/prolexic-threat-advisories/prolexic-threat-advisory-spike-ddos-toolkit-botnet/spike-ddos-toolkit-cybersecurity-A4-092414.pdf.

[7] Bohio, M. J. Analysis of a MIPS malware. GIAC, SANS Institute. https://www.giac.org/paper/grem/2573/analyzing-backdoor-bot-mips-platform/124977.

[8] Prolexic. IptabLes/IptabLex DDoS Bots. September 2014. http://www.prolexic.com/kcresources/prolexic-threat-advisories/prolexic-ddos-threat-advisory-iptables-iptablex-linux-bots-botnet-cybersecurity/TA-DDos-Binary-Bot-IptabLes-v6-A4.pdf.

[9] MalwareMustDie! ITW Infection of ELF .IptabLex & .IptabLes China #DDoS bots malware. June 2014. http://blog.malwaremustdie.org/2014/06/mmd-0025-2014-itw-infection-of-elf.html.

[10] Dr.Web. Linux.Myk.5. http://vms.drweb.com/virus/?i=4365599.

[11] Krebs, B. Lizard Stresser Runs on Hacked Home Routers. January 2015. http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/.

[12] Security Affairs. Hackers target Bash Bug vulnerability in the wild. September 2014. http://securityaffairs.co/wordpress/28626/cyber-crime/hackers-use-bash-bug.html.

[13] Dr.Web. Linux.BackDoor.Fgt.1. November 2014. http://vms.drweb.com/virus/?i=4242198.

[14] Blasco, J. Attackers exploiting Shellshock (CVE-2014-6721) in the wild. September 2014. https://www.alienvault.com/open-threat-exchange/blog/attackers-exploiting-shell-shock-cve-2014-6721-in-the-wild#sthash.bzInLhax.dpuf;.

[15] MalwareMustDie! Fuzzy reversing a new China ELF Linux/XOR.DDoS. September 2014. http://blog.malwaremustdie.org/2014/09/mmd-0028-2014-fuzzy-reversing-new-china.html.

[16] Kálnai, P. Linux DDoS Trojan hiding itself with an embedded rootkit. January 2015. https://blog.avast.com/2015/01/06/linux-ddos-trojan-hiding-itself-with-an-embedded-rootkit/.

[17] FireEye. Anatomy of a Brute Force Campaign: The Story of Hee Thai Limited. February 2015. https://www.fireeye.com/blog/threat-research/2015/02/anatomy_of_a_brutef.html.

[18] Level3. It Takes a Village – Collaborative Steps to Breaking Botnets: How Level 3 and Cisco Worked Together to Improve the Internet's Security and Stop SSHPsychos. April 2015. http://blog.level3.com/security/breaking-botnets-how-level-3-and-cisco-worked-together-to-improve-the-internets-security-and-stop-sshpsychos/.

[19] Github. An LKM rootkit targeting Linux 2.6/3.x on x86(_64), and ARM. https://github.com/mncoppola/suterusu.

[20] Evidence of a DDoS attack. February 2015. https://gist.github.com/Manawyrm/74e687b2a01527725d33.

[21] ValdikSS. Some tools to monitor BillGates CnC servers. February 2014. https://github.com/ValdikSS/billgates-botnet-tracker.

APPENDIX

Elknot Builder	F126C3F8530587F7CADEB8B969BC04AB114B468922171A953211345AD5A8F380
Elknot Xitele Kit	48183D0DD8DA484639ADDA9F60E5FEA340D7C6B4C77458384EE98CB21972ADE5
Elknot EM_386	D1F922A762BBD4E0725D4625BE4A39CEBFA03D1875339E9F01F825A2DCDC9E65
Elknot EM_386 (Setag)	568A52AA9A9AC2698BA7C49FE4A3AEE34D96FE0F25ECCB31FC726D941BB135EA
Elknot (Setag) UPX	3E89F0D71671DB79506050E0823D121EA5A19457308AF3E379AC45A0338B1B33
Elknot Script	21ACDA48CAD399B049D03A51A64C9E4BB2DC96C1916BC4EECD6FC828E8036083
Chinaz EM_386	A86B1899821C2833B989A736E928A4137FA6D0954C9816747F6AFEF536F757F9
Chinaz EM_x86_64	1EB72C76F79FA01CE39198C91AF5C7A4E36897E9A9A8F5D29CA68BA7371A2361
Chinaz EM_MIPS	87934D993BB5262FB2826DA05CB4657EC6B20849A65C5D00D260BBF58878F45E
Chinaz Sources	992ED01DEF5ECE5B90CE242820D2BFDD580FDDDE12DBC10CE5A395A7923922C9
MrBlack C&C	1828AC46C67E120274688A562D04E9E9A629C39090A848956FB7DB45B6551B74
MrBlack Builder	E83F69052FC240DC43FC2B32F77408B2B3488E67B29B04041E7C6B8622CE8602
MrBlack EM_x86_64	F2DF127535902E6390CE2EC198C12A5BD9A361901C2D8008A064DF96EFD10E29
MrBlack EM_x86_64 UPX	6DD946E821DF59705DCFEB79FAB810336D0EE497FD715FB5B6711E05C0428F4D
MrBlack EM_386	8766317F20B05C792514ADDD8BB4904021049ACD86E8D70E9FFFD1D12FAD51CE
MrBlack EM_ARM	26FCBDC7EBE2750B4008D8C67186A9DA03D34B994662BD93E49D7C572AADBAE0
MrBlack EM_MIPS	736C08988602155954C02CBEF0B4ED3DD916C7EB659032202F15081620058988
MrBlack Sources	8499E6727253FA98DACC3D753CC08CB207C64A290D9521E94A65C2BDA34F405E
Aesddos EM_ARM	AF765C0F87846E6E1A184B64A4DA8E51588F0F6A7048FEFDD60B53058373C6B6
Aesddos EM_386	D6E77D8F2FFDF61981241022E8D7034014927BFFA23793739051CAD34867F766
Wrkatk EM_386	288D91AF1B5F3A57C0B3D66330F56BBCD38604948B3154CD4842D277FA86F664
Wrkatk EM_x86_64	0940E4A72DBA133838CCD0992914C5FB2BF106D5A018F289B9C5896C0E237CC6
Wrkatk EM_MIPS	8A1CE3302E896CD695528EB0CC744EC6E18C1D708C944BE7C8AFFB3B4D44BD5D
Wrkatk EM_ARM	4ED6E5CFA9D7006E021BBD099AAFD4F2ADAAE3307DC25262E240D9E8829B960D
Iptablesx Dropper	F41C4C9EE0FBAEFF5397F27531A91135C1D98C54A9E0BDC6CA52315E3E208537
Iptablesx EM_386	9F89CA6F4580F6EBE021D2C2E2C528B93E4492C4B6E6BD5F339361E86F8585D8
Sotdas EM_386	E75E49AC157DADC8C4E7230D531BE0DB6FBC339B5D75B7AB8FA6202CE0EC8E2A
Sotdas EM_x86_64 UPX	59D53A8DFB2B646293E422743EAF8C6F3AB576BACCDF36BB133C4F458AAF60A3
Xorddos C&C	496F413E6C8B6F258C238AF6EAF61C2B524DC0DC985E4E659627ADAE1ED31517
Xorddos Script	BA84C056FB4541FE26CB0E10BC6A075585990F3CE3CDE2B49475022AD5254E5B
Xorddos Uploader	44153031700A019E8F9E434107E4706A705F032898D3A9819C4909B2AF634F18
Xorddos EM_386	AD26ABC8CD8770CA4ECC7ED20F37B510E827E7521733ECAEB3981BF2E4A96FBF
Xorddos EM_x86_64	859A952FF05806C9E0652A9BA18D521E57090D4E3ED3BEF07442E42CA1DF04B6
Xorddos EM_ARM	49963D925701FE5C7797A728A044F09562CA19EDD157733BC10A6EFD43356EA0
Xorddos Rootkit	6BE322CD81EBC60CFEEAC2896B26EF015D975AD3DDA95AE63C4C7A28B7809029
Gafgyt Server	2A04C216FCE75D19E5162081EB747B8A77C205F6DD933B0864C08FB086C929C5
Gafgyt EM_x86_64	BAABCECAC23775FDD3E52CD1FB0E4C46777A6747E854074ECE751767D13F6DD7
Gafgyt EM_386	28EA6EE1080B4D436685D0D0C87EEF492EA2A376917437E865D0D1513114B8D7
Gafgyt EM_ARM	67FF5F3F10AD86ED0A9F90244E7B5BE839AFB0AAEB49E22130551A09A0F08FF8

Gafgyt EM_MIPS	04BEF883E7098FDA9148A75C43165D45AC5FBB8B6032848E9C5D9A5E3897DF52
Gafgyt EM_PPC	7F13A4C911AB0682D9A7F5988DA9C7BE0AE781CE15945E4C0AA76A78E22CBF2F
Gafgyt EM_SH	D59C7CF8D9EFBD93F0B907C12BB4C18CC5CE7D800B234DB219D2D919C0B0AFDC
Gafgyt EM_SPARC	277D2D00E27BCF4536BB492CAC16001E8832DC9BBED384A8C523B49A199790E6
Gafgyt EM_x86_64	BAABCECAC23775FDD3E52CD1FB0E4C46777A6747E854074ECE751767D13F6DD7
Gafgyt EM_68K	4E611FB1466920885D1216AB7D9B4F16A3F31D52CF7B39FFC21FC6CA41534738
Gafgyt Script	8D0B152A91202356B3B5470C5C017B4E9595C5325D8C14DA1DEBBE1782225A14
Gafgyt Sources	1AF299A269FFDB4461E181CA774FC307A592288AD4B3F6B93226C955EB9B8084