

Dirt Jumper: A New and Fast Evolving Botnet-for-DDoS

M. Marquez Andrade, N. Vljajic
Department of Computer Science and Engineering
York University
Toronto, Canada

Abstract— In July 2011, a fairly new and rather aggressive strain of botnet-for-DDoS malware, named **Dirt Jumper**, was identified by Arbor Networks. Since then, numerous incidents of DDoS attacks involving this strain of malware have been reported. In this paper, we first give a general overview of **Dirt Jumper**'s history, structure and operation as it has been documented on the Internet. Subsequently, we present the results of our own analysis of **Dirt Jumper**, conducted using the GFI Sandbox environment. We also provide an overview of **Pandora DDoS toolkit** – the latest offspring coming out of the **Dirt Jumper** family, which appeared on the black botnet marked in the early 2012. We conclude the paper by outlining some areas of continuing and future work.

Keywords: DDoS; botnet; *Dirt Jumper*; sandbox analysis;

I. INTRODUCTION

Since its advent back in 1990, WWW (the Web) has revolutionized almost every aspect of our lives. For many people, the Web has become the primary medium through which they retrieve information, conduct business or establish and maintain social relationships. Our ever-increasing reliance on the Web, however, comes at the price of an ever-increasing vulnerability to different forms of intrusions and attacks on the computer networks and hosts comprising the Web/Internet. Distributed Denial of Service (DDoS) is generally recognized as one of the most dangerous threats to the normal operation and availability of the Web. As such, DDoS has been a subject of great interest – both for those who work on defending various aspects of the Web infrastructure, as well as for those who aim to exploit the Web's weaknesses in order to achieve financial or political gain.

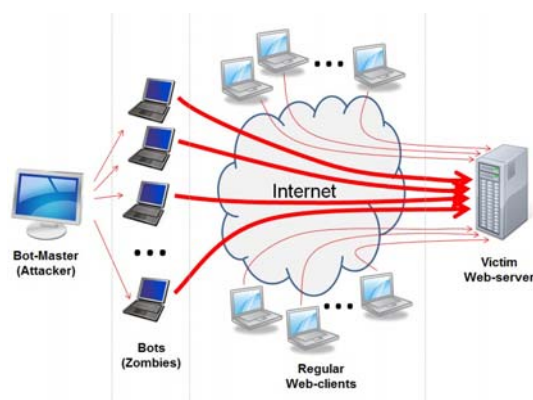


Figure 1. A botnet executing a DDoS attack.

Although the motives and targets of DDoS attacks can greatly vary, the commonality of all DDoS attacks is that they involve concerted efforts to saturate the victim machine (often a web-server) with a large volume of traffic, leaving the server unable to respond to legitimate user requests. The most common way of executing a DDoS involves the use of a system of compromised/infected machines, the so-called botnet (see Fig. 1). Most botnets discovered so far operate in a centralized manner, with the master machine (owned and operated by the actual cybercriminal) remotely controlling the compromised third-party computers, also known as bots or zombies, via the so-called command and control (C&C) center. The execution of a DDoS is accomplished with the master instructing the zombies to send large amounts of attack traffic to the victim machine, either directly or through some form of route reflection.

Nowadays, easy-to-use botnet building toolkits are readily available on the underground online marked, at an average cost of only several hundred US\$. Already formed and fully functional botnets can be purchased for several thousand US\$, or rented for under US\$100 a day [4]. In a recent report by a researcher from Arbor Networks [5], the most prevalent types of botnets currently being offered (and operating) on the Internet are identified. Among those, **Dirt Jumper** is described as 'one of the most popular', 'fairly new' and 'rather aggressive' botnet strains. The goal of our work has been to look into the history of **Dirt Jumper**'s evolution, as well so to gain a better understanding of the actual structure and operation of this particular type of botnet.

In the first part of this report (Sections II, III, IV), we provide a general historic and technical overview of **Dirt Jumper**, as it has been documented on the Internet. We also outline the key characteristic of **Pandora** - the latest progeny in the line of botnet-for-DDoS toolkits derived from the original **Dirt Jumper** platform. In the second part of the report (Section V), we present some of our own findings obtained by experimenting with **Dirt Jumper** and **Pandora**. We conclude the report with an outline of future research directions.

II. HISTORY OF DIRT JUMPER

According to our findings, the earliest documented occurrence of **Dirt Jumper**, which originally appeared under the name **Russkill**, dates back to January 2009. In Table 1, we provide a chronological list of key events related to **Dirt Jumper**'s evolution, detection and reported incidents of DDoS attacks. Although not exhaustive, the list reflect the overall

level of activity since 2009 to date, as performed by both – the hackers and cybercriminals that engineer and deploy Dirt Jumper, as well as numerous anti-malware companies that try

to keep this and other forms of malware under reasonable control.

TABLE I. CHRONOLOGICAL LIST OF KEY EVENTS RELATED TO DIRT JUMPER TABLE TYPE STYLES

January 12, 2009	– Detection of RussKill bot exe file, wihpg.exe, by Comodo antivirus. http://camas.comodo.com/cgi-bin/submit?file=ca63f9b726cd923620c6179edd24c16b9bd2340e54b22a44666f4ce762e594c2&iiframe
December 15, 2009	– Appearance of first article, by Malware Intelligence, describing RussKill. http://malwareint.blogspot.com/2009/12/russkill-application-to-perform-denial.html
December 29, 2009	– Earliest Virus Total analysis spotting presence of RussKill infection, specifying wihpg.exe as the bot. https://www.virustotal.com/file/ca63f9b726cd923620c6179edd24c16b9bd2340e54b22a44666f4ce762e594c2/analysis/1262027702/
February 10, 2010	– Appearance of another article, by NoVirusThanks, on RussKill revealing that one of its Command and Control centers was located at akakalat.com [4]. http://blog.novirusthanks.org/2010/02/a-new-ddos-bot-named-russkill-is-in-the-wild/
August 4, 2010	– Nartv.org publishes a more detailed discussion/analysis of RussKill. http://www.nartv.org/2010/08/04/the-ambler-botnet/
May 8, 2011	– Symantec provides signatures for Dirt Jumper. http://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=24603
July 4, 2011	– Appearance of DDoS-for-hire business add in underground forum, offering Dirt Jumper version 3 as one of its instruments. http://ddos.arbornetworks.com/2011/08/dirt-jumper-caught/
July 15, 2011	– etp.roseltorg.ru is attacked by Dirt Jumper. http://ddos.arbornetworks.com/2011/08/dirt-jumper-caught/
September, 2011	– Appearance of new Dirt Jumper, version 3 with MD5: f29b1089b3f5e076d4d4bd2a3a02d3cb . http://www.deependresearch.org/2011/10/dirt-jumper-ddos-bot-new-versions-new.html
November 17, 2011	– Krebsonsecurity.com is attacked by RussKill with Command and Control at noteye.biz. http://www.zimbio.com/Hacking/articles/mua8Jd7nmsg/DDoS+Attack+KrebsOnSecurity+com+using+Russkill
November 30, 2011	– Several small Financial institutions are disturbed by Trojan and DDoS attacks. Dirt Jumper is the suspected perpetrator. http://krebsonsecurity.com/2011/11/ddos-attacks-spell-gameover-for-banks-victims-in-cyber-heists/
December 29, 2011	– Prolexic issues Dirt Jumper Threat advisory with high risk factor and releases free security scanner. http://www.darkreading.com/authentication/167901072/security/news/232301120/prolexic-issues-dirt-jumper-threat-advisory-and-releases-free-security-scanner.html
February 6, 2012	– MoneyManagement.com.au is hit by a DDoS attack, which will later be identified to have been carried out by a Dirt Jumper botnet. http://digitaljournal.com/article/319366
February 10, 2012	– tradingroom.com.au is hit by a DDoS attack, which is allegedly similar to the one Dirt Jumper performed over MoneyManagement.com.au on February 6 th . http://www.smh.com.au/it-pro/security-it/dirty-dealings-and-dirtjumper--financial-websites-fend-off-extortion-attacks-20120210-1sity.html
February 10 & 14, 2012	– Multiple non-government sites in Russia are hit by DDoS attacks conducted by Dirt Jumper-type of botnet(s). The attacks are suspect to be politically motivated, leading up to the Russian elections. http://www.dataprotectioncenter.com/security/ddos-attacks-in-russia-added-to-protests/
February 22, 2012	– Virus Total publishes analysis of Dirt Jumper version 5. https://www.virustotal.com/file/b3ed2acb025ba5624d61056433ea9d119d031622dcd7de25c723a83e598e0419/analysis/1329931549/
February 23, 2012	– Onthar.in leaks the binaries of Dirt Jumper version 5. http://onthar.in/articles/dirt-jumper-ddos-bot-analysis-version-5/

April 3, 2012	– Onthar.in analyzes the newest version of Dirt Jumper, Pandora. http://onthar.in/articles/pandora-ddos-bot-analysis/
July 27, 2012	– Krebsonsecurity.com is attacked by Pandora. http://krebsonsecurity.com/2012/08/triple-ddos-vs-krebsonsecurity/
August 8, 2012	– Prolexic releases a thread advisory for Pandora with medium risk factor. http://www.prolexic.com/e/9892/TkNCX4/7fz6v/107547155

III. STRUCTURE AND OPERATION OF DIRT JUMPER

A. Toolkit

The construction toolkit for Dirt Jumper botnet retails for as little as \$150 on various underground forums [1]. The toolkit consists of: 1) a PHP/MySQL application to build the botmaster's administrative-directory (to be used on top of a running SQL engine), 2) an executable that builds the actual malware binary (BuilderDJ3.exe), and 3) an executable that acts as the builder template (d3v3.exe) [1]. Once the builder generates the malware(bot) binary, the spreading of the malware and the formation of the botnet can begin. Those who do not want to go through the burden of building their own botnet might instead choose to use the services of a DDoS provider. According to [1], a botnet with Dirt Jumper capability/features can be rented for US \$10 per hour, or US \$45 per day.

B. Spreading

The Dirt Jumper bot binaries (i.e., the actual malware) is usually spread via spam, exploit kits, fake downloads (fake video codec, backdoored pirated software), or can be pushed out to machines already infected with other forms of malware, such as Zeus or Spyeeye [1].¹

C. Malware Installation & System Changes

Depending on the version/variant of the malware, there are two ways how Dirt Jumper gets installed on a system [2]:

1) *As a Windows Service.* Bots with this type of installation correspond to MD5=f29b1089b3f5e076d4d4bd2a3a02d3cb. Changes that may signal the presence of this type of bot on a system include:

a) Presence of the following files:

<system folder>\drivers\svgtook.exe (or svflooje.exe)

<Windir>\keys.ini (only contains the 15 digit bot ID)

b) Presence of the registry modifications such as (file name may vary):

HKLM\SYSTEM\CurrentControlSet\Services\svgtook

HKLM\SYSTEM\...\Services\svgtook\Security

HKLM\SYSTEM\...\Services\svgtook\Enum

¹ Variants of Zeus and Spyeeye steal passwords and give attackers direct access to the infected computers, which enable subsequent installation of other forms of malware (including the installation of Dirt Jumper) on the given machines.

2) *As a binary executed with Winlogon.* Bots of this kind correspond to MD5=f7c0314fb0fbd52af9d4d721b2c897a2. System changes that may signal the presence of this version of Dirt Jumper include:

a) Presence of the following files:

<system folder>\svdhalp.exe

<system folder>\svdhalp.exe.ini

<Windir>\syskey2i.drv (only contains the 15 digit bot ID)

b) Presence of DATA "explorer.exe, svdhalp.exe" under TYPE "Shell" in the following registry subkey:

HKCU\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon.

D. Command&Control (C&C) Communication

Dirt Jumper belongs to the group of HTTP-based command and control (C&C) botnets ([1], [2], [3]). As mentioned in Section III.A, the Dirt Jumper C&C center operates as a PHP/MySQL server, and can be accessed by its respective bots via standard HTTP protocol (i.e. through a URL of the form: <http://domain-naime/index.php>). Most of its domain names resolve into IP addresses located in Russia, Latvia and US. For a detailed list see [2].

The domain name of a Dirt Jumper C&C server is hardcoded into the binaries of all its respective bots. Hence, once a bot/binary has been successfully activated on the host machine, it first proceeds to resolve the provided domain name into a valid IP address (by contacting the local DNS server) [2]. Subsequently, the bot sends an HTTP-POST request to the obtained C&C IP address, in order to: 1) authenticate itself with its respective C&C server, and 2) obtain the list of targets to attack [1]. (The authentication process is performed by means of 15-digit ID placed in the payload of the initially sent HTTP-POST request. The HTTP response from the C&C server contains three pipe-delimited values followed by the URL of the sites to attack (e.g., 01|300|150<http://www.victim.com>) [3], where:

1) The 1st delimited value represents a 'command code' and determines the mode and type of attack to be executed by the bot [3].

2) The 2nd delimited value specifies the number of threads to be deployed during the attack (see Section III.E).

3) The 3rd delimited value defines the time delay (i.e. periodicity) to contact the C&C server [1].

In cases when there is no ongoing attack, the response from the C&C server contains only the three pipe-delimited values.

The entire communication lifecycle of a Dirt Jumper bot is illustrated in Fig.2.

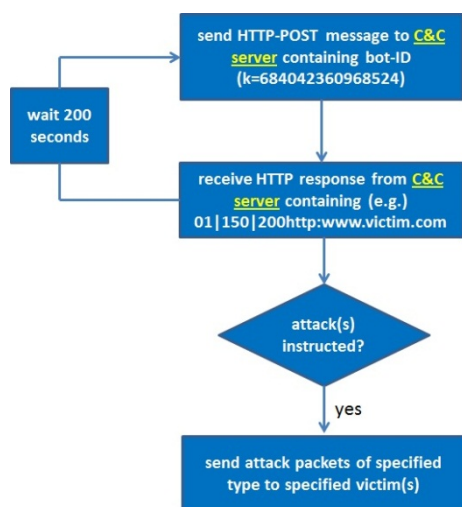


Figure 2. Communication lifecycle of Dirt Jumper bot.

E. Types of DDoS Attacks

According to [1], Dirt Jumper v.3 enables the bot-master to perform the following four types of DDoS attacks:

1) *HTTP Flood* – This type of attack causes server overload by means of repeated, conventional HTTP requests. As soon as the bot receives the response from the server, the bot breaks the underlying TCP connection, and sends a new request.

2) *Synchronous Flood* – In this type of attack, each bot sends 150 (or more) simultaneous requests to the server. Once the server has responded to all the requests, the bot repeats the procedure. According to the author of Dirt Jumper, this type of attack is generally more powerful and, thus, gives more instantaneous results than the standard HTTP flood.

3) *Downloading Flood* – This type of attack aims to cause the bandwidth saturation of the victim's site, by instructing the bots to download/request larger (more bandwidth consuming) files.

4) *POST Flood* – In contrast to 3), this type of attack aims to create a processing overload on the victim server. To achieve the given goal, the bots are instructed to send random usernames and passwords (embedded in HTTP-POST packets) to web-site forms hosted on the victim server.

The September 2011 version of the bot introduces two additional forms of attack ([1], [2]):

5) *Multipurpose Flood (Light)* – This type of attack aims to decrease the probability of (early) detection by performing a dynamic change in the following: 1) the size, content, timeout and sending rate of attack packets; 2) user agent and referrer value in HTTP requests; etc.

6) *Multipurpose Flood (Full)* – This attack is very similar to 5), except that in addition to HTTP-GET it also employs HTTP-POST requests, which (as in 4)) are aimed at increasing the processing load on the victim server.

In the HTTP packets of all above mentioned types of attacks the Connection field is set to 'Keep-Alive', with the obvious intention of forcing the connection to stay open, and thus occupy more of the server's memory and processing resources [1].

IV. STRUCTURE AND OPERATION OF PANDORA TOOLKIT

In February 2012, a prepackaged toolkit named Pandora - developed by the same individual known as the author of Dirt Jumper ('sokol') - appeared on various malware forums [15]. The toolkit is especially prized for its attack potency. In particular, the toolkit is advertised as requiring only 10 bots to take down weak sites, 30 bots to bring down medium-sized sites with little protection, and 1000 bots to bring down web-site with more sophisticated protection [15].

In terms of its structure and operation, Pandora shares many commonalities with Dirt Jumper. However, Pandora also comes with a few noteworthy changes, including [15]:

1) Pandora has evolved to send 4 pipe delimited values preceding the target URL: type of attack, duration of attack, connect-back intervals and timeouts.

2) The bot authentication ID has been changed from a 15-digit number into a 32-byte long alphanumeric string.

3) The Pandora C&C offers five types of attacks:

- *HTTP Min* – In case of this attack, the bot sends a TCP-SYN request and immediately closes the connection, without waiting for the response from the server. The incomplete TCP 3-way handshake procedure leaves the server in the state of wasteful waiting.
- *HTTP Download* – In this type of attack, the bot leaves the server in the state of 'waiting' after establishing a full TCP connection.
- *HTTP Combo* – This attack alternates between *HTTP Min* and *HTTP Download*.
- *Socket Connect* – The purpose of this attack is to concentrate on a particular socket on the victim server.
- *Max Flood* – In this type of attack the bot will issue POST requests which are greater than 1,000,000 bytes in length to the victim server, with the purpose of congesting the server's upload channel.

V. MITIGATION STRATEGIES FOR DIRT JUMPER

Recently, Prolexic has proposed several strategies for mitigation of attacks generated by Dirt Jumper botnet(s) [17]. They involve blocking suspicious packets, restoring the registry and removing the executables and services installed by the bots in the zombie machines. These techniques are generally applicable to Pandora botnet(s). However, being a faulty version of Dirt Jumper, there are additional ways to mitigate Pandora attacks, including:

- Blocking packets with ET requests. Namely, when “Socket Connect” attack is selected, the infected machines send improper ET requests (instead of proper GET), due to typographical error in the payload itself.
- Blocking POST requests with payloads greater than 1,000,000 bytes - an unusual size which correspond to attack 4 or “Max Flood” [15].

VI. EXPERIMENTAL RESULTS

Our experimentation involving Dirt Jumper has been conducted using the GFI Sandbox environment [6]. In Section IV.A, we give a general overview of the GFI Sandbox environment, while in Section IV.B we outline some of our most important findings to date.

A. GFI Sandbox Environment

GFI Sandbox - formerly CWSandbox - is a sophisticated industry-leading tool for quick and safe analysis of malware behaviour. In particular, the tool is capable of analyzing the behavior of any suspected Windows application, including infected Microsoft Office documents, malicious URLs, and scripts in Flash ads. The analysis is conducted by executing the malware of interest in a secure and monitored environment. The final analysis reports include: the details of how the application was executed on the desktop, what system changes were made, the network traffic generated, and the severity level of the threat. GFI Sandbox is implemented as a client-server system. The server component handles: client management, automation, analysis warehousing, sample storage, and reporting. The client component(s) are the sandboxed Windows machines where the actual sample execution and analysis occurs.

B. Dirt Jumper Experimentation

1) Experimentation with Bot Binaries

For the purposes of our research, we have obtained Dirt Jumper bot binaries from four reputable sources: DeepEnd Research, GFI ThreatTrack Feeds, Telus Security Labs, and Virus Total. The GFI Sandbox analysis² has determined that two of these binaries corresponded to MD5=f29b1089b3f5e076d4d4bd2a3a02d3cb, and the other two to MD5=f7c0314fb0fbd52af9d4d721b2c897a2 (see Section III.C). The binaries, when executed, were attempting to connect to C&C centers at [7] (which, at the time, mapped to IP= 216.218.158.19) and [8] (IP=178,79.172.145) respectively, by periodically sending HTTP-POST request (see Section III.D). The user-agent field in the headers of all generated HTTP-POST packets was falsely set to: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US). Unfortunately, at the time of our experimentation with the given binaries, there were no active C&C servers running on the designated addresses, hence the bots never received any returning packets/information. In terms of the number and nature of

system changes performed during the installation of these two versions of Dirt Jumper, the results of our GFI Sandbox analysis were pretty much in line with the earlier findings, as outlined in Section III.C.

In order to obtain a copy of a ‘live bot’³, we have additionally looked at various public repositories on the Internet. We have been fortunate to land upon three new (i.e, very recent) versions of Dirt Jumper at [9] which is a file-sharing site. Using GFI Sandbox, it has been determined that the first of the binaries obtained from this site corresponds to MD5=ee560cc68c01615f8b864fda5fdce9a7, and is programmed to attempt to connect to a C&C center at [10] (IP=31.170.162.183). As in the earlier cases, at the time of our experimentation, there were no active C&C servers running on this particular URL/address. On the other hand, our GFI Sandbox analysis has revealed that in terms of its installation and number/nature of system changes, this version of Dirt Jumper is somewhat different from the ones previously discussed and analyzed. Specifically, this version of Dirt Jumper is of ‘As a Windows Service’ type, but the number of files that get stored in <system folder> and the number of registry modifications are different from those discussed in III.C.1).

The second binary obtained from [9] is shown to correspond to MD5= d482027b31abd7f081ec80e9d5ee4c75, and it attempts to communicate with a C&C center at [11] (IP=209.190.85.14). At the time of our experimentation, which spanned the course of a few days, this particular bot turned out to be ‘live’, and it managed to successfully connect to the designated C&C server. The responses returned by the server contained the following commands: 11|200|120http://... and 11|200|120http://..., specifying [12] and [13] as the victim websites.

The third binary we obtained from [9] corresponds to the MD5= 1179b7f18dcbb64fecf4ddb7c3ea8ce7, and appears to be a version of Pandora. For now we haven’t been able to document the communication between this bot and its C&C since once the bot starts running it does not appear to initiate any communication. However, an unusual characteristic of this bot sample is that it infects sysWOW64 and not system32 folder as expected, even though decompilation of the binary shows that the system32 path is the one hardcoded.

Decompilation of the Dirt Jumper v3 binaries shows a hard coded list of User-Agents, Referrers, and the C&C domain. The Pandora version only shows a list of User-Agents, while the Referrers seem to be randomly generated strings and the domain of the C&C is encrypted. On the other hand, in none of the versions can the source code be extracted.

Once the binaries are executed on the victim machines, we are able to use Task Manager to view the list of bot processes. In the case of Dirt Jumper v3 most of its process file names start with “sv...”, such as “svflooje” shown in Fig. 3.

² The majority of our experiments described in this document were conducted in February and March 2012.

³ By ‘live bot’, we refer to a bot that attempts (and manages) to connect to an active/running C&C centre.

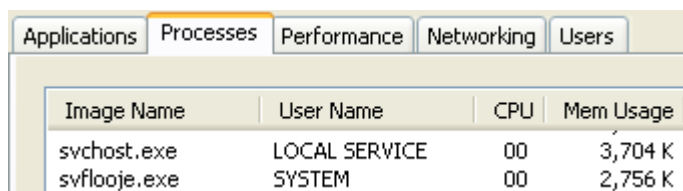


Image Name	User Name	CPU	Mem Usage
svchost.exe	LOCAL SERVICE	00	3,704 K
svflooje.exe	SYSTEM	00	2,756 K

Figure 3. Screenshot of bot process in task manager

The Pandora version of the bot doesn't show any running process, however it can be spotted running as a service in the services tab, with the name "ServerNabs4".

2) Experimentation with Botnet Set-up Toolkits

In the March 2012 report released by Prolexic [1], one site hosting a copy of the Dirt Jumper set-up toolkit was identified (see Section III.A). After downloading these set-up toolkit, we have been able to do the following:

a) Build copies of our own Dirt Jumper binary, with any arbitrary C&C server URL embedded in them. Consequently, this has allowed us to create test-bots for the purpose of probing any .php address suspected of hosting a Dirt Jumper C&C center. One such bot, built to connect to [11], has successfully confirmed the existence of the 'live' C&C center mentioned in the previous section.

b) Build a test C&C server, in order obtain a better understanding of the server's built-in functionalities as well as the ways it governs its respective bots. We have managed to successfully mount the PHP/MySQL application found in the Dirt Jumper set-up toolkit on top of the WampServer environment. Subsequently, we have been able to confirm that the version of the toolkit that we currently possess includes provisions for four different types of DDoS attacks, as illustrated in Fig. 3.

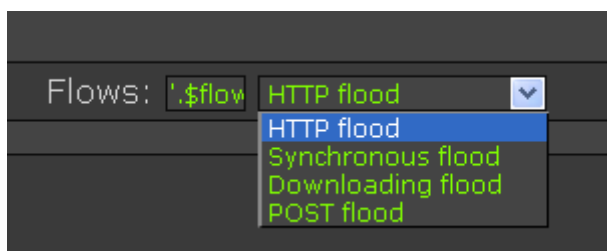


Figure 4. Snapshot of GUI window of our own test C&C server

The database setup required to run the server was simple – in particular, an install.php file in the toolkit was run to create the appropriate sql tables in the database. The tables were subsequently populated by the bots (i.e., their respective IDs) attempting to communicate with the C&C through the index.php page. Even though the index.php page accesses the database to register the bots it doesn't do any sanitizing of the parameters passed to it.

Once we were able to instruct the bots to attack, it was possible to observe that the flows value refers to the number of threads the bot will create to attack. This is also confirmed by

the fact that in the newest version (Pandora) the flows value was replaced by "threads".

We were also able to understand the meaning of each of the 3 pipe delimited values in the response packets sent from the C&C to its bots. Namely, once we instructed the bots to attempt different kinds of attacks we could observe that the second digit of the first pipe delimited value shifted from 1 to 4. Accordingly, we were able to conclude that the second digit of the first pipe delimited value refers to the type of attack to be executed. Subsequently we instructed the bots to attack and observed the first digit of the first pipe delimited value shift from 0 to 1. Thus, this digit refers to the attack/don't attack command. The second and third pipe delimited values allude to the number of threads and delay before next communication with C&C respectively.

In addition to the toolkit mentioned above we were also able to find a Pandora version of the toolkit at an anonymous file sharing site [16]. Since our Pandora bot sample mentioned in V.B.1 wasn't communicating we decided to use the toolkit to decrypt the domain it had hardcoded. As all toolkits in the Dirt Jumper family, the Pandora toolkit includes a builder.exe file which does the domain hardcoding into the bots. The fact that we were able to decompile the bots and see the encrypted domains gave us access to unlimited text to cyphertext samples. With this information and several trials we were able to discover the encryption algorithm for the domains. The algorithm consists of reversing the letters of the domain and applying Vigenère encryption with a key of length 6. However once we were able to decrypt the domain that had been hardcoded into the bot sample we realized it consisted of a repetition of the same letter. Hence this has led us to conclude that the domain must have been hardcoded somewhere else in the code, thus supporting the idea that this sample is unusual. The sample may pertain to a customized version of the Pandora toolkit.

VII. CONCLUSIONS

The findings of our research suggest that the Dirt Jumper C&C centers are still vulnerable to detection, as the structure of the C&C doesn't validate the "authenticity" of the bots attempting to become part of its botnet (see Section IV.2.a).

Our experimentation has also shown that the leaked toolkits can be used not only to scan probable C&C domains but also to help decrypt hardcoded domains from bot samples, since the "builder" executable provides a text to cyphertext engine.

Furthermore, we have been able observe that in most versions of Dirt Juper toolkit there is not only a lack of attention concerning the origin of the bots, but also concerning the information they submit to the server, as the submitted data is not sanitized. Thus the C&C is vulnerable to attacks to its database. Since this problem hasn't been handled in the newer versions of Dirt Jumper we believe they are relying on the secrecy of their server location for protection. This is supported by the fact that in the Pandora version the domain of the C&C is now encrypted, something that wasn't done in the previous versions.

The fact that the new Pandora version has implemented measures such as encryption of its domain, non-instantaneous communication, and modification of its 15-digit ID to an alphanumeric ID reflects the interest of the creators to guard the bots against decompilation, sandboxing, and communication detection.

The results from analyzing several Dirt Jumper binaries with the GFI Sandbox environment provide new evidence to confirm the idea that there exist a variety of Dirt Jumper bots in the wild, which alter their behavior in basic operations such as their installation. During our investigation we have observed the fast evolution of the Dirt Jumper family of toolkits. Since the writing of this paper 3 new versions have been commercialized and leaked and we have been able to gain access to them. Such a rapid evolution and dissemination of different Dirt Jumper variants is the best proof of its wide acceptance as one of the main tools in today's botnet-for-DDoS market.

VIII. FUTURE WORK

This document outlines the preliminary findings of our ongoing research on Dirt Jumper botnet-for-DDoS malware. Some of our immediate and future goals include:

1) Conduct systematic tracking of active Dirt Jumper C&C servers, and gain a better understanding of their geo-distribution and migration, as well as their DDoS attack patterns.

2) Build an isolated Dirt Jumper botnet environment, with one fully functional C&C server and several fully functional bots. By experimenting with different instruction codes issued by the C&C server, and by observing the traffic generated by the bots, we hope to be able to obtain a better perspective on different types of attacks that can be executed using Dirt Jumper.

REFERENCES

- [1] Prolexic, "Threat: Dirt Jumper v3", Prolexic Threat Advisories <http://unknown.prolexic.com/pdf/ProlexicThreatAdvisoryDirtJumper.pdf>, (Access Date: 15 Feb, 2012).
- [2] DiMino, Andre' M., and Mila Parkour. "DeepEnd Research: Dirt Jumper DDoS Bot - New Versions, New Targets." DeepEnd Research. <http://www.deependresearch.org/2011/10/dirt-jumper-ddos-bot-new-versions-new.html> (Access Date: 26 Feb, 2012).
- [3] Wilson, Curt. "Dirt Jumper Caught in the Act" Arbor Networks Security Blog. <http://ddos.arbornetworks.com/2011/08/dirt-jumper-caught/> (Access Date : 07 Feb, 2012).
- [4] Mohan, Ram. "The Rise of the Small Botnet" SecurityWeek, <http://www.securityweek.com/rise-small-botnet> (Access Date: 15 Feb, 2012).
- [5] Wilson, Curt. "Attack of the Shuriken: Many Hands, Many Weapons", Arbor Networks Security Blog. <http://ddos.arbornetworks.com/2012/02/ddos-tools/> (Access Date: 15 Feb, 2012).
- [6] GFI Sandbox, "Automated Malware Analysis tool" <http://www.gfi.com/malware-analysis-tool>, (Access Date: 15 Feb, 2012).
- [7] asdaddddaaaa.com/678/index.php (Access Date: 10 Feb, 2012).
- [8] xzrw0q.com/driver32/update/m_d.php (Access Date: Feb, 2012).
- [9] rghost.ru (Access Date: Mar, 2012).
- [10] gamearena.net84.net/raq/index.php (Access Date: Mar, 2012).
- [11] jawa360.500mb.net/index.php (Access Date: Mar, 2012).
- [12] bike53.ru (Access Date: Mar, 2012).
- [13] seriouspartner.ru (Access Date: Mar, 2012).
- [14] Onthar, "Pandora DDoS Bot Analysis", Malware Research <http://onthar.in/articles/pandora-ddos-bot-analysis/>, (Access Date: 27 Aug, 2012).
- [15] Prolexic, "Threat: Pandora DDoS Toolkit", Prolexic Threat Advisories <http://www.prolexic.com/e/9892/TkNCX4/7fz6v/107547155> (Access Date: 27 Aug, 2012).
- [16] Anonymous, "Pandora DDoS Bot", Anonfiles.com <https://anonfiles.com/file/a5dce6f5abd375d5c21a12a9a064802d> (Access Date: 31 Aug, 2012).
- [17] Prolexic, "Prolexic Issues Dirt Jumper Threat Advisory and Releases Free Security Scanner", <http://www.prolexic.com/company/news-events/prolexic-issues-dirt-jumper-threat-advisory-and-releases-free-security-scanner.html> (Access Date: 27 Aug, 2012).