An overview of the WCMS brute-forcing malware landscape

Anna Shirokova[1] and Veronica Valeros[1]
[1] Cognitive Threat Analytics, Cisco Systems

This paper was presented at Botconf 2017, Montpellier, 6-8 December 2017, www.botconf.eu
Published in the Journal on Cybercrime & Digital Investigations by CECyF, https://journal.cecyf.fr/ojs

Abstract

Web Content Management Systems (WCMS) provide simple tools to manage web content that enables users with little knowledge of programming languages and web design. WCMSs have become extremely popular in the last decade. WordPress, with more than 18M websites world wide, is the most prominent WCMS. Is because of its popularity that this and other well-known WCMSs have been systematically attacked for the past years by different threat actors seeking disposable infrastructure for their attacks.

Brute-force attacks are one of the most common types of attacks against WCMSs. The goal of such an attack is to guess a valid user name and password in order to access the WCMS administration panel. Attackers especially take advantage of users choosing weak credentials. Successfully brute-forced websites are typically used for hosting C&Cs, scams, and drive-by attacks to spread malware.

This paper presents an historical overview and current state of WCMS brute-force attacks with a focus on botnets and techniques used. We present a case of study of Sathurbot, a modular HTTP-based botnet. Finally, we discuss detection methods to identify these type of attacks.

Keywords: botnet, brute-force, network, IDS.

1 Introduction

Web Content Management Systems (WCMS) [1] are sets of tools designed to simplify the creation and administration of web content. The fact that users do not require prior knowledge on programming or web design in order to use them caused WCMSs to quickly gain popularity. Most well known WCMSs are based on PHP, such as WordPress [2], Drupal [3], Joomla [4] and vBulletin [5]. Other WCMS are written in Python, such as Blogger [6]. Among WCMSs, WordPress is the most popular with more than 18M users in 2017 [7], as observed in Table 1.

WCMS	Release	Technology	Websites
WordPress	2003	PHP	18M
Joomla	2005	PHP	2M
Blogger	1999	Python	787k
Drupal	2000	PHP	701k
vBulletin	2000	PHP	24k

Table 1: Summary of Web Content Management Systems (WCMS) by popularity. Source: https://trends.builtwith.com/cms.

Since their appearance, because of their rapid and wide adoption, WCMSs have been systematically attacked by threat actors looking for disposable infrastructure for their cyber attacks. Among the most common attacks against WCMSs are brute-force attacks. The aim of this type of attack is to find a valid user name and password combination that would allow attackers complete access to the web administration panel of the WCMS. Attackers work on the assumption that users will choose weak credentials. Successfully brute-forced websites are commonly sold and used for hosting C&Cs, scams, and drive-by attacks to spread malware among others.

However, brute-force attacks still remain prevalent against WCMSs and there is no extensive research done in this area. During our research we were able to find a few blog or forum posts describing brute-force attacks; most of these posts described the user experience of a brute-force attack on their personal

WCMS site. Although these bits of information are extremely valuable, they still lack the deep technical knowledge which can be applied in creating a better defence mechanism against brute-force attacks.

The rest of this paper is divided into 5 sections. Section 2 presents an historical overview of brute-force malware and techniques used. An in-depth analysis and case of study is presented in Section 3. Section 4 outlines limitations encountered during the research and provides suggestions for the future work in this area. Finally, in Section 5 the conclusions of this work are presented.

2 Historical overview of brute-forcing malware and techniques

Brute-force attacks are one of the most common type of attacks against WCMSs. The main goal of this attack is simple: to obtain a valid user name and password and get access to the WCMS administration panel. To understand the scale of the problem this section will provide an historical overview of the evolving brute-forcing threat landscape.

The first report of distributed brute-force attack against a WCMS was in 2009 [8]. The article describes a small PHP script designed to launch a distributed attack against WordPress administration panels. The script brute-force function received three parameters: cURL structure to perform HTTP request, the target website, and the password to attempt. The script connected to a MySQL database to retrieve a list of websites and passwords to be used in the attack, giving the attacker the possibility to run multiple scripts in parallel. The attack could be distributed in terms of sites to try and also in terms of passwords to attempt. The article mentions that brute-force attacks against WordPress have been around for a while, but this is the first documented case found at the time of this research.

The Stantinko botnet was operated since 2012 [9], and uncovered by ESET in 2017. This is a modular botnet with backdoor and brute-force capabilities. The Stantinko plugin capable of performing brute-force attacks is called *brutplugin*. This plugin is used to start a distributed dictionary-based attacks against WordPress and Joomla WCMS. Each bot retrieves a list of user names and passwords for the attack. All the attempts performed by the bot are reported back to the Stantinko C&C. The paper does not provide metrics on the success rate of attempts performed by brute-force plugin.

The Trojan *WPCracker1*, also known as *Fort Disco*, was discovered in 2013 by DrWeb [10] and further analyzed later that year by Arbor Networks [11]. It was the first widely known malware that used brute-force attacks as a spreading mechanism. This malware was targeting Windows users. Once infected, the bot will download from the C&C a list of target websites. Fort Disco targeted Joomla and WordPress sites. According to the reports, the list of websites could be shared among bots. The bot will also retrieve from the C&C a password or list of passwords to use in the attack. The user names used by the bot were hard-coded in the binary. As mentioned in the reports, the infection mechanism is unclear.

In 2014, independent security researchers working under the handle of MalwareMustDie[1] reported an attack to web servers based on Linux and FreeBSD, and they called the threat *Mayhem* [12]. Researchers from Yandex expanded the research on this threat in their VirusBulletin paper [13]. The Mayhem botnet has a modular structure enabled by a diverse set of plugins. There are eight known plugins that give the botnet a range of functionality ranging from FTP brute-force to crawling and WCMS brute-forcing. The *brute-force.so* plugin is designed to brute-force WordPress and Joomla websites. Researchers discovered that Mayhem is a continuation of the Fort Disco brute-force campaign reported in 2013.

In February 2015, miss-configured *Aethra* routers were compromised due to a weak default password. The compromised devices got infected with a malicious piece of code used to launch a distributed brute-force attack against WordPress sites [14]. Unfortunately, the report does not provide any additional information to help to identify which malware family was installed on the compromised routers. In September 2015, Kaspersky discovered a ransomware variant called *Troldesh* [15]. Kaspersky researchers discovered that this ransomware did not only encrypt the files of the infected victim but also contacted its C&C server in order to obtain new payloads. Troldesh was delivering four additional malware families into the infected machine: Zemot, Muret, Kovter, and CMSBrute. The CMSBrute malware contacts its C&Cs located in the Tor network to download additional plugins. These plugins determine the WCMS installed on the targeted sites, searches for the admin panel, and performs the brute-force attacks via dictionary attack. CMSBrute targets Joomla, WordPress, and DataLifeEngine websites. In December 2015, a Cisco researcher discovered a new payload delivered by Andromeda/Gamarue botnet. The new malware, named *CMSCatcher*, was designed to perform brute-force attacks against WordPress websites [16]. Once CMSCatcher contacts the C&C server, it will download a list of websites to brute-force. For every site on the list retrieved from the C&C, the malware attempts to log in with a default user name and password combination (admin, admin). If successful, it will report back to the C&C. This piece of malware was reported to be very aggressive, attempting to log in more than 200,000 sites in less than four days of activity.

In 2016, it was reported that the botnet known as *ChikenKiev* was using WordPress sites to spread malicious content [17]. To get access to the WordPress sites, malicious actors were using brute-force attacks [18]. There is no further research on this threat and no

[1] MalwareMustDie blog: http://blog.malwaremustdie.org/

other publication on ChickenKiev was found to date.

3 In-depth analysis of the Sathurbot brute-forcing botnet

As a practical case study, we will use *Sathurbot* brute-forcing botnet. Sathurbot first appeared in 2013 [19], and it is still active and affecting hundreds of users. To this date, Sathurbot has four known modules: backdoor, downloader, web crawler, and brute-forcing module. The *downloader module* allows the Trojan to deliver additional malware to the infected machine. Sathurbot is known to deliver Boaxxe, Kovter, and Fleercivet Trojans. The *web crawler module* allows the Trojan to search in different search engines for websites using WordPress WCMS. The *brute-force module* is how the Trojan attempts to log in to the WordPress admin panels with different credentials. The case of study focuses on the web crawling and brute-forcing modules with specific insights obtained from a real life infection. The rest of this section will cover the data collection process, the dataset used for the analysis, and the dynamic analysis of a Sathurbot infection. In particular, this section will describe the infection mechanism, how the crawling and brute-forcing module work, and insights of the behavior of the botnet such as the password distribution, attack prevalence, botnet infrastructure, and C&C domain organization.

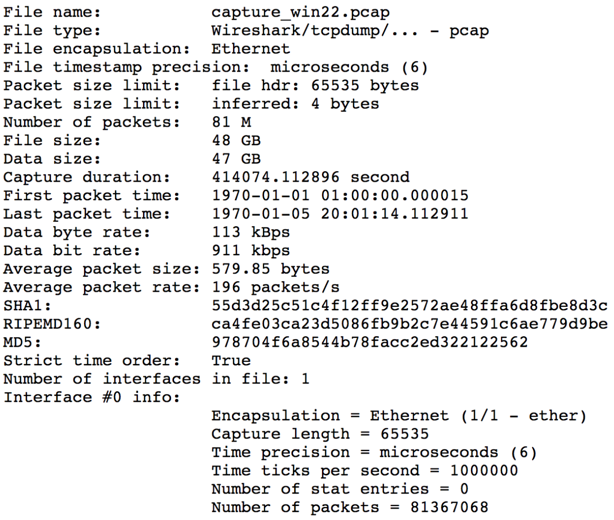
Data collection

For the analysis described in the following Sections we used a packet capture of a real Sathurbot infection provided by the StratosphereIPS Laboratory [20]. In particular, we used capture *300-1* [2].

The capture 300-1 was obtained by running a Sathurbot sample [3] in the StratosphereIPS Laboratory. The sandbox infection timeline was as follow:

- Start the MITM Proxy [4] for HTTPS interception
- Start a Windows VM.
- Uninstall VirtualBox Guest Additions & restart.
- Install the original BitTorrent client [5].
- Quit Skype.
- The BitTorrent client automatically started.
- Quit BitTorrent client.
- Execute the Sathurbot sample.

The Sathurbot infection ran in the sandbox environment for four days starting on 19 July 2017. In the investigation, the primary sources of data used were HTTP flows (*capture_win22.weblogng*), pcap file (*capture_win22.pcap*), and dnstop results (*capture_win22.dnstop*). Figure 1 shows additional information about the pcap capture of the Sathurbot infection obtained with the Capinfos tool.



```
File name:           capture_win22.pcap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Packet size limit:   inferred: 4 bytes
Number of packets:   81 M
File size:           48 GB
Data size:           47 GB
Capture duration:    414074.112896 second
First packet time:   1970-01-01 01:00:00.000015
Last packet time:    1970-01-05 20:01:14.112911
Data byte rate:      113 kBps
Data bit rate:       911 kbps
Average packet size: 579.85 bytes
Average packet rate: 196 packets/s
SHA1:                55d3d25c51c4f12ff9e2572ae48ffa6d8fbe8d3c
RIPEMD160:           ca4fe03ca23d5086fb9b2c7e44591c6ae779d9be
MD5:                 978704f6a8544b78facc2ed322122562
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1/1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 81367068
```

Figure 1: Information about capture file used for analysis, including the file type, number of packets, date and time information, and file hashes.

Infection mechanism

The infection chain starts when a user is searching for pirated content in search engines. Search engines such as Google, Yandex, and Bing will index compromised sites and show them as a result of users' queries. These compromised websites host malicious torrent files infected with the Sathurbot Trojan. The URL below is an example of an URL leading to a torrent file infected with Sathurbot:

```
hxxp://hkcs.lk/land-of-mine-2015-kickass-
free-movie-download-torrent/
```

Malicious movie torrents contain a video file, a codec pack installer, and a text file with instructions. Malicious software torrents contain an installer executable and a text file with instructions. When the executable is launched in a system that has a Torrent client installed, it loads the Sathurbot DLL (Dynamic-Link Library). A pop-up error message is displayed to the user while the malware is being installed in the background. After successful infection, the infected host becomes part of the Sathurbot botnet.

[2] https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-300-1/
[3] 20ae9e5f8f26635c627afce5eaeeb749af459f55138c80f29da9d787ecc38f92
[4] https://mitmproxy.org
[5] http://www.bittorrent.com/

Sathurbot crawling module

Once a victim is infected with Sathurbot the crawling module starts. This module is in charge to find target websites by performing web searches in three search engines: Google, Bing, and Yandex. The bot performs the search engines queries using the HTTP protocol, therefore all the requests and responses are not encrypted. The lack of encryption allowed us to collect the complete combination of words that the bot is searching for in each search engine.

In order to understand the behavior of this module, we extracted all the HTTP requests performed by the infected host to each search engine. We confirmed that Sathurbot only performs queries to Google, Bing, and Yandex. After the information was extracted, we performed a comparison of the requests to the different search engines in order to determine possible differences by search engine. A summary of the findings in detailed below.

Sathurbot performs HTTP requests to the Bing search engine in plain text, without using encryption, as observed in Figure 2, and queries for a combination of two to four words. Once the bot obtains a response, the bot will parse it and hand it to the brute-force module. The top five most common words queried in Bing are *beauty, report, data, google, and practice*. This is illustrated in Figure 3 by a cloud of words with annotated number of occurrences.

```
GET /search?q=makers%20manage%20manual HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Host: www.bing.com
```

Figure 2: HTTP requests to the Bing search e engine as performed by Sathurbot in capture 300-1.

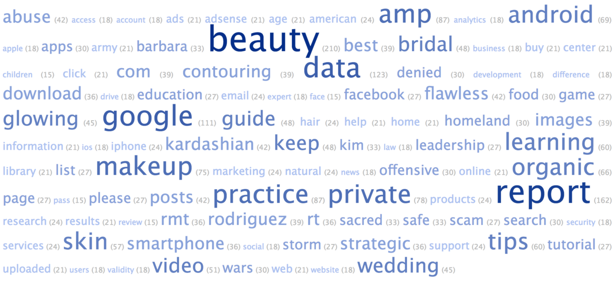


Figure 3: Most common words queried on Bing search engine by Sathurbot malware as observed in the 300-1 analyzed capture.

Sathurbot also performs plain HTTP requests to the Google search engine. An example of an HTTP requests is shown in Figure 4. We also observed that Sathurbot queries Google for a combination of two to four words. Additionally, the bot adds the parameter `num=100` to the query, which is used to restrict the number of results per query. The words queried in Google are very similar to those queried in Bing. The top five most common words are *beauty, data, report, google, and skin*. This is illustrated in Figure 5 by a cloud of words with annotated number of occurrences.

```
GET /search?q=madness%20mailing%20makers&ie=utf-8&oe=utf-8&gws_rd=cr&num=100 HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Host: www.google.com
```

Figure 4: HTTP requests to the Google search engine as performed by Sathurbot in capture 300-1.

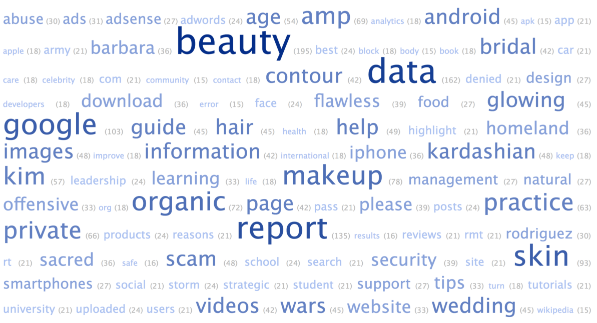


Figure 5: Most common words queried on Google search engine by Sathurbot malware as observed in the 300-1 analyzed capture.

The queries performed by Sathurbot to the Yandex search engine are different from those performed against Bing and Google. The queries to Yandex are performed without encryption, using plain HTTP requests as illustrated in Figure 6. The line of dots on the GET request are the result of the interpretation of the characters by Wireshark. Every dot is a special character in Cyrillic that Wireshark substitutes in order to be able to display the request. This behavior is quite different from the other search engines, as for Yandex, Sathurbot instead of searching for a combination of words it searches for a combination of letters. The most common combination of letters used to query Yandex are illustrated in Figure 7. During our research we could not find the reason why the botnet could do this change in the search queries.

```
GET /search/?text=....................%20........%20.............. HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Host: yandex.com
```

Figure 6: HTTP requests to the Yandex search engine as performed by Sathurbot in capture 300-1. The line of dots on the GET request are the result of the interpretation of the characters by Wireshark. Every dot is a special character in Cyrillic that Wireshark substitutes in order to be able to display the request.

p,k,c,a	r,j,g,q	t,e,d,o	f,c,m,t
g,g,k,o	d,p,b,r	k,n,q,b	k,o,j,l
n,q,j,i	g,d,j,e	e,k,s,m	l,l,j,l
p,p,o,c	o,c,l,l	f,h,b,s	r,c,s,h
p,l,b,b	q,i,d,t	o,i,k,e	l,h,t,b
g,g,k,q	d,d,g,p	d,j,b,a	j,f,h,m
o,l,i,g	g,q,b,t	g,i,o,l	d,k,l,m
t,c,g,p	n,t,m,k	j,s,j,i	e,k,o,e
c,g,h,d	r,i,e,b	g,e,n,t	e,q,d,i

Figure 7: Most common letters queried on Yandex search engine by Sathurbot malware as observed in the 300-1 analyzed capture.

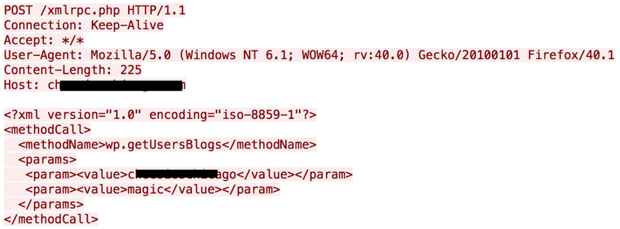
In this analysis we could determine that to perform a search in Google and Bing, the bot used an almost identical combinations of words. In Yandex however, the bot was using consistently a combination of four letters. After the crawling module harvested lists of domains, they were probed to identify WordPress WCMS. After this phase, the bot performed an HTTP GET request to the website administrative panel. Successful results were reported to the C&C server.

Sathurbot brute-forcing module

Sathurbot brute-force attack targets two authentication methods of WordPress WCMS, specifically *XML-RPC* call and *form based authentication*. *XML-RPC* [6] is a remote procedure call which uses XML for the data exchange and the HTTP protocol as a transport method. This functionality is implemented in WordPress WCMS. This method is not primarily used for authentication, however, many XML-RPC calls are required to provide credentials. *Form based authentication* is the primary authentication method used by WordPress. To authenticate, a user needs to provide valid credentials in a web form. For the majority of the WordPress based websites, the authentication form is located in the same resource `wp-login.php`. An example of a WordPress administrative panel URL is:

```
http://www.example.com/wp-login.php
```

Sathurbot uses these two methods to perform the brute-force attacks. The first brute-force attempt against a website is through the misuse of a XML-RPC call, specifically the method `wp.getUserBlogs` [21]. This method is not the only RPC call requiring authentication. Other RPC methods which require authentication could also be used to brute-force WordPress websites. An example of an HTTP request using the XML-RPC method is shown in Figure 8.

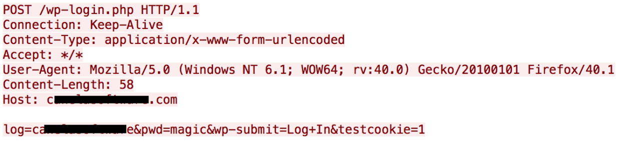


```
POST /xmlrpc.php HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Content-Length: 225
Host: ch[redacted]

<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>wp.getUsersBlogs</methodName>
  <params>
   <param><value>c[redacted]ago</value></param>
   <param><value>magic</value></param>
  </params>
</methodCall>
```

Figure 8: One of the methods used by Sathurbot to brute-force WordPress sites is by the misuse of the `wp.getUserBlogs` XML-RPC call. The bot performs a HTTP request to the target website using this method and specifying the credentials to be used.

The second brute-force attempt against websites is preformed by custom HTTP requests to the website web form used for authentication, typically called `wp-login.php`.It was previously reported [19] that Sathurbot performs a single log in attempt where the user name is also the domain name of the target website and the password is obtained from the C&C server. We can confirm this behaviour to some extent. In our research we also identified that in many cases the bot was behaving differently. In the traffic capture 300-1, it is possible to observe that the bot was attempting to log in to the same target website several times. In many of these cases, the bot used a special user name which was not related to the domain name of the target website. In at least one occasion the bot used a specific user name which, upon further investigation, we determined that it belonged to one of the website administrators. It is unknown at the time of this research how these special user names were gathered; one possibility is that other unknown parts of the botnet were abusing the WordPress user enumeration capability [22]. Previous reports [19] also stated that Sathurbot used one password per bot for brute-forcing. Our analysis indicates that every bot is using more than one password. We observed several brute-forcing attempts to the same target website in a different time period and with different passwords used. An example of an HTTP request using the form based authentication as performed by Sathurbot is shown in Figure 9



```
POST /wp-login.php HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Content-Length: 58
Host: c[redacted].com

log=ca[redacted]e&pwd=magic&wp-submit=Log+In&testcookie=1
```

Figure 9: Sathurbot performs HTTP requests to the form based authentication page as a brute-forcing method.

[6] http://xmlrpc.scripting.com/

Password distribution

One of the discoveries of this analysis was that each Sathurbot is not using a single password but many of them. Since the Sathurbot brute-force attack is performed via plain HTTP POST requests, it was possible to obtain the combinations of user names and passwords that the infected machine was attempting. The majority of the user names were the same as the domain names of the targeted sites, which for privacy reasons we are not covering in this work. The attempted passwords, however, proved to be quite interesting as they were provided directly by the C&C server.

In the four days of the infection, the bot used 546 unique passwords for the brute-force attack. Contrary to possible expectations, the password 'pericles' was the most common, used 46,569 times, followed by ----, yamaha, panda1, and root123. The top 20 most common passwords observed in the capture 300-1 are shown in Figure 10.

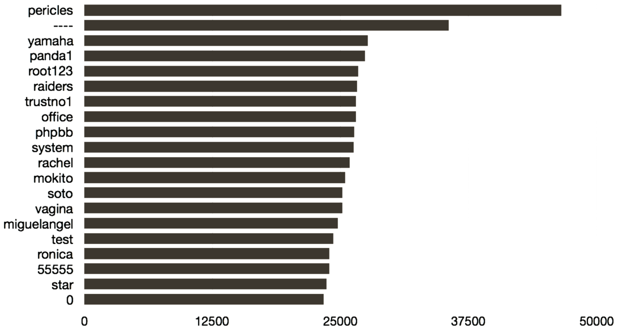


Figure 10: Sathurbot top 20 most common passwords observed in the capture 300-1.

The passwords used by the botnet seem to be unique and were collected specially for the brute-forcing purpose. We were not able to find these passwords in any known password lists publicly available online up to date.

Attack prevalence analysis

To analyze the brute-force attack prevalence, we analyzed Top-level Domains (TLD) [7] of the target websites observed in capture 300-1. For this purpose, we divided the analysis per TLD type: general purpose and country based TLDs. In the Sathurbot capture 300-1 we observed domains with 50 different TLDs. There were ten general purpose TLDs: com, org, net, info, xyz, top, club, edu, biz, and pro. The most requested TLD was com with 58% of all the domains. The high number of com domains is due to the number of websites hosted in wordpress.com; these websites would look like myblog.wordpress.com. The distribution of general purpose TLDs is shown in Table 2.

TLD	Count	Percentage
com	1552601	58 %
org	139582	5.2 %
net	102798	3.8 %
info	23288	0.9 %
xyz	16076	0.6 %
top	11233	0.4 %
club	9659	0.4 %
edu	9254	0.3 %
biz	7067	0.3 %
pro	5971	0.2 %

Table 2: Most common general purpose TLDs requested by the Sathurbot infected machine as observed in capture 300-1.

An analysis of the country based TLD distribution indicates that attack was not targeted to any specific country or region. In total we observed domains with 50 different TLDs. Table 3 shows the distribution of requests to country based TLDs. The highest number of sites in this group are from Germany, with 2.5% of requests, closely followed by UK and the rest.

Sathurbot infrastructure

The study of the Sathurbot infrastructure was performed using the packet capture file of the 300-1 malware infection. In order to identify the command-and-control communication in the capture, we discarded XML-RPC and WordPress authentication form connections performed by the bot. The result was a smaller packet capture which we used to identify the core behaviors associated to the core functionality of the botnet. The observed behavior is described next.

The capture starts with traffic to the BitTorrent trackers and advertisements. This may be due the fact that the BitTorrent client started automatically after it was installed just before the infection. This traffic can also be associated to the botnet itself, which may be seeding malicious torrents to keep spreading.

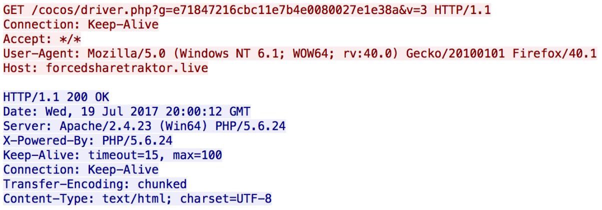
The BitTorrent traffic is followed by a HTTP GET request to google.com. This type of isolated request is usually performed by malware to check the internet connectivity of the host.

After the connectivity check, the bot performed a DNS request to retrive the IP of a domain that was hard-coded in the binary: forcedsharetraktor.live. We determined that this is the first C&C server of the Sathurbot botnet. After successfully resolving the domain, the bot contacts the C&C performing via HTTP. An example of the HTTP request and response to the first C&C server is shown in Figure 11. The periodicity of the connections to this C&C server is every two hours via HTTP GET requests and every 10 seconds via HTTP POST requests.

[7] https://en.wikipedia.org/wiki/Top-level_domain

TLD	Count	Percentage
de	68078	2.5 %
uk	59681	2.2 %
nl	45528	1.7 %
cc	45419	1.7 %
cn	36527	1.4 %
au	35410	1.3 %
it	32400	1.2 %
br	28158	1.1 %
pl	26216	1.0 %
fr	25319	0.9 %
ca	24766	0.9 %
ru	21802	0.8 %
es	17372	0.6 %
eu	14732	0.6 %
se	14284	0.5 %
in	13431	0.5 %
cz	13365	0.5 %
ch	11686	0.4 %
us	11434	0.4 %
za	10814	0.4 %
co	10631	0.4 %
ro	9589	0.4 %
dk	9567	0.4 %
be	8809	0.3 %
ir	7395	0.3 %
at	6735	0.3 %
tk	6411	0.2 %
jp	6194	0.2 %
me	5937	0.2 %
id	5555	0.2 %
hu	5507	0.2 %
nz	4962	0.2 %
no	4930	0.2 %
cl	4777	0.2 %
tv	4706	0.2 %
gr	4611	0.2 %
lt	4377	0.2 %
mx	4373	0.2 %
fi	4349	0.2 %
ar	4328	0.2 %

Table 3: Full list of targeted country TLDs as observed in the capture 300-1. The table shows that there is no specific country or region targeted by the brute-force attack.



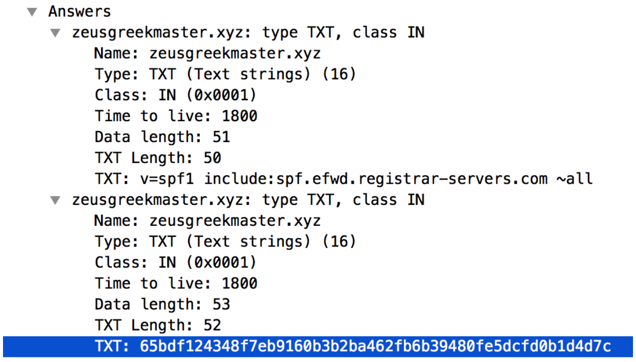
```
GET /cocos/driver.php?g=e71847216cbc11e7b4e0080027e1e38a&v=3 HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Host: forcedsharetraktor.live

HTTP/1.1 200 OK
Date: Wed, 19 Jul 2017 20:00:12 GMT
Server: Apache/2.4.23 (Win64) PHP/5.6.24
X-Powered-By: PHP/5.6.24
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

Figure 11: The infected machine contacts the first C&C server, forcedsharetraktor.live, via HTTP.

Simultaneously to the first C&C request, the infected machine starts the crawling activity. This is described at the beginning of section 3, and is characterized by constant HTTP requests to Google, Bing, and Yandex.

After the successful response of the first C&C the bot proceeds to perform a DNS request to the domain zeusgreekmaster.xyz. We determined that this is the second C&C server. The DNS response contains an encoded DNS TXT record. After decoding, this TXT record contained the location of the third C&C. There was only one request to this C&C in the capture. An example of the DNS response TXT record is shown in Figure 12.

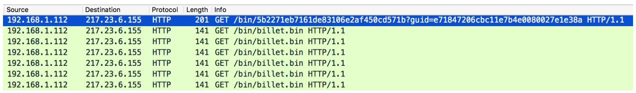


```
▼ Answers
  ▼ zeusgreekmaster.xyz: type TXT, class IN
      Name: zeusgreekmaster.xyz
      Type: TXT (Text strings) (16)
      Class: IN (0x0001)
      Time to live: 1800
      Data length: 51
      TXT Length: 50
      TXT: v=spf1 include:spf.efwd.registrar-servers.com ~all
  ▼ zeusgreekmaster.xyz: type TXT, class IN
      Name: zeusgreekmaster.xyz
      Type: TXT (Text strings) (16)
      Class: IN (0x0001)
      Time to live: 1800
      Data length: 53
      TXT Length: 52
      TXT: 65bdf124348f7eb9160b3b2ba462fb6b39480fe5dcfd0b1d4d7c
```

Figure 12: The infected machine contacts the second C&C server, zeusgreekmaster.xyz, via DNS. The response contains encoded information in the TXT record, which contains the address of the third C&C server.

Once the bot decodes the information retrieved from the second C&C, it contacts the domain uromatalieslave.space via HTTP. We determined that this is the third C&C server. We observed that bot was performing only POST requests to this C&C with a periodicity of sixteen minutes.

The bot contacted a fourth C&C server, hosted in megafreecontentdelivery.club, several times during the capture. In each of these occasions, the bot downloaded binaries from the C&C. Our assessment indicates that these binaries were updates to the bot. Example of HTTP requests performed to the fourth C&C are shown in Figure 13.



```
Source          Destination     Protocol  Length  Info
192.168.1.112   217.23.6.155    HTTP         201  GET /bin/5b2271eb7161de83106e2af450cd571b?guid=e71847206cbc11e7b4e0080027e1e38a HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
192.168.1.112   217.23.6.155    HTTP         141  GET /bin/billet.bin HTTP/1.1
```

Figure 13: The infected machine contacted the fourth C&C server, megafreecontentdelivery.club, in several occasions. Each time it was downloading binary files, which may be updated for the bot.

The third and fourth C&Cs are resolving to the same IP address which is 217.23.6.155. The full list of C&C domains observed in capture 300-1 are listed in the Appendix A.

The behavior described above is illustrated in Figure 14. The upper part of the illustration shows the identified C&C servers. In the lower part of the chart are the activities triggered by the different botnet modules.

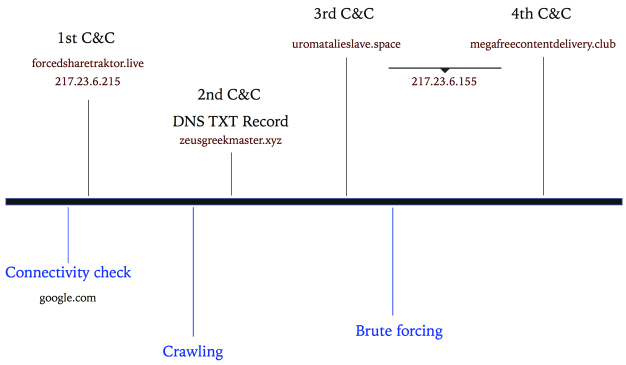


Figure 14: The sequence of connections to the Sathurbot C&Cs observed by analyzing the traffic of the 300-1 infected machine.

C&C domain name analysis

During our research we discovered an interesting naming pattern used in Sathurbot C&C's infrastructure. After collecting all known C&C domains associated to Sathurbot, we could observe there were specific words repeating in those domain names. The words repeated were `force`, `master`, `slave`, and `boom`.

Our analysis indicates that all the domains sharing the word `force` were domains found to be hard-coded in the Sathurbot binaries. This first stage C&C domains were used for reporting, receiving initial updates, and potentially obtaining new second stage C&C domains. Similarly, domains sharing the word `master` were used for obtaining the location of the third C&C servers via encoded DNS TXT records. The domains sharing the word `slave` were used for reporting successful brute-force attempts. Finally, the domains sharing the word `boom` were used to retrieve updates, and they seem to be also associated with torrent tracker traffic. Figure 15 illustrates the groups of domains and the common words among them.

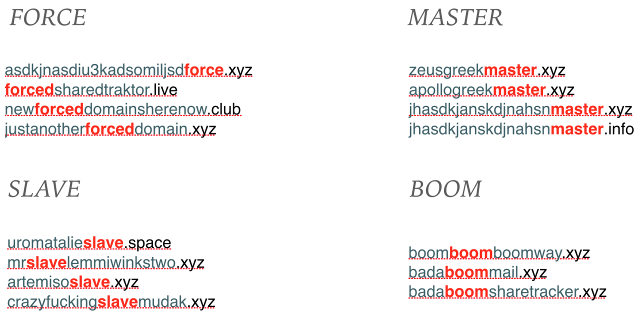


Figure 15: Sathurbot C&Cs grouped by common words, and the particular role played by the C&C in the botnet infrastructure.

4 Limitations and Future work

Our research faced two main limitations. First, the lack of research papers on WCMSs brute-forcing attacks. Second, the absence of a proved method to measure the successful rate of web brute-force attacks.

In our future work we consider to proceed with experiments that will help develop a method to measure the success rate of brute-force attacks on WCMSs. We believe a measure of this kind will help the security community to compare different brute-forcing botnets, identify how successful this botnets are, and how long is required for the botnets to successfully compromise a website. This information can help to implement accurate detection mechanisms. Other area that we consider to work on is the study of how quickly these botnets change the passwords among bots, and to track down successfully brute-forced sites in order to understand how these sites are used after they are compromised.

5 Conclusion

In this paper we introduced the first historical overview of malware with brute-forcing capabilities targeting WCMS. We focused primarily on WordPress WCMS as is the most popular in terms of users and attackers. The historical overview confirmed that WCMS brute-forcing botnets have existed since the beginning of this technology, employing a variety of methods. We presented an analysis of the botnet Sathurbot from the network behavioral perspective. We described in detail the main modules of Sathurbot botnet which are crawling module and brute-forcing module.

Our investigation showed that brute-force attacks against websites are automated, primitive, and yet successful. Detection of brute-force attacks requires a certain expertise in different types of attack techniques, and defendants still struggle in this area.

Brute-force attacks against WCMSs are still one of the major threats in the Internet, and they will continue to exist as long as they are successful. This research aims to increase the public attention to this topic and to encourage the security community to expand the research in this area.

Appendices

A List of observed Sathurbot C&C

The full list of C&C servers we found associated to the Sathurbot botnet during the dynamic analysis covered in this paper:

```
asdkjnasdiu3kadsomiljsdforce.xyz
forcedsharedtraktor.live
newforceddomainsherenow.club
justanotherforceddomain.xyz
zeusgreekmaster.xyz
apollogreekmaster.xyz
jhasdkjanskdjnahsnmaster.xyz
jhasdkjanskdjnahsnmaster.info
uromatalieslave.space
mrslavelemmiwinkstwo.xyz
artemisoslave.xyz
crazyfuckingslavemudak.xyz
boomboomboomway.xyz
badaboommail.xyz
badaboomsharetracker.xyz
```

Acknowledgment: The authors would like to thank Sebastian Garcia for providing the datasets of Sathurbot Trojan used for the analysis presented in this paper [20].

Author details

Anna Shirokova

Cognitive Threat Analytics, Cisco Systems
Karlovo namesti 10, Praha 2, Czech Republic
ashiroko@cisco.com

Veronica Valeros

Cognitive Threat Analytics, Cisco Systems
Karlovo namesti 10, Praha 2, Czech Republic
vvaleros@cisco.com

References

[1] Wikipedia, “Web content management system.” https://en.wikipedia.org/wiki/Web_content_mana-gement_system, 2017. Last accessed 04 August 2017.

[2] “Wordpress.” https://www.wordpress.com/, 2003. Last accessed 04 August 2017.

[3] “Drupal.” https://www.drupal.org/, 2000. Last accessed 04 August 2017.

[4] “Joomla.” https://www.joomla.org/, 2005. Last accessed 04 August 2017.

[5] “vBulletin.” https://www.vbulletin.com/, 2000. Last accessed 04 August 2017.

[6] “Blogger.” https://www.blogger.com/, 1999. Last accessed 04 August 2017.

[7] BuiltWith, “Wordpress usage statistics.” https://trends.builtwith.com/cms/WordPress, 2017. Last accessed 04 August 2017.

[8] SANS Internet Storm Center, “Brute distributed wordpress admin account cracking.” https://isc.sans.edu/diary/Distributed+Wordpress+admin+account+cracking/7663, 2009. Last accessed 04 August 2017.

[9] ESET, “Stantinko: A massive adware campaign operating covertly since 2012.” https://www.welivesecurity.com/wp-content/uploads/2017/07/Stantinko.pdf, 2017. Last accessed 04 August 2017.

[10] DrWeb, “New trojan compromises blog sites in russia and other countries.” https://news.drweb.com/show/?i=3811, 2013. Last accessed 04 August 2017.

[11] Arbor Networks, “Fort disco bruteforce campaign.” https://www.arbornetworks.com/blog/asert/fort-disco-bruteforce-campaign/, 2013. Last accessed 04 August 2017.

[12] MalwareMustDie, “MMD-0020-2014 - Analysis of Linux/Mayhem infection: A shared DYN libs malicious ELF: libworker.so.” http://blog.malwaremustdie.org/2014/05/elf-shared-so-dynamic-library-malware.html, 2014. Last accessed 04 August 2017.

[13] A. Kovalev, K. Otrashkevich, and E. Sidorov, “MAYHEM - A HIDDEN THREAT FOR *NIX WEB SERVERS.” https://www.virusbulletin.com/uploads/pdf/magazine/2014/vb201407-Mayhem.pdf, 2014. Last accessed 04 August 2017.

[14] Voidsec, “Aethra botnet.” https://voidsec.com/aethra-botnet-en/, 2015. Last accessed 04 August 2017.

[15] Kaspersky, “The shade encryptor: a double threat.” https://securelist.com/the-shade-encryptor-a-double-threat/72087/, 2015. Last accessed 04 August 2017.

[16] V. Valeros, “Make it count: An analysis of a brute forcing botnet.” https://journal.cecyf.fr/ojs/index.php/cybin/article/view/5, 2015.

[17] Wordfence, “Analysis: Methods and monetization of a botnet attacking wordpress.” https://www.wordfence.com/blog/2017/01/wordpress-botnet-monetization/, 2016. Last accessed 04 August 2017.

[18] Wordfence, “Huge increase in brute force attacks in december and what to do.” https://www.wordfence.com/blog/2016/12/how-to-protect-against-brute-force-attack/, 2016. Last accessed 04 August 2017.

[19] ESET, “Sathurbot: Distributed wordpress password attack.” http://www.welivesecurity.com/2017/04/06/sathurbot-distributed-wordpress-password-attack/, 2017. Last accessed 04 August 2017.

[20] S. Garcia, “Malware Capture Facility Project.” https://stratosphereips.org, 2017. Last accessed 04 August 2017.

[21] WordPress, “XML-RPC WordPress API.” https://codex.wordpress.org/XML-RPC_WordPress_API, 2017. Last accessed 04 August 2017.

[22] Rapid7, “Wordpress brute force and user enumeration utility.” https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_login_enum, 2017. Last accessed 04 August 2017.