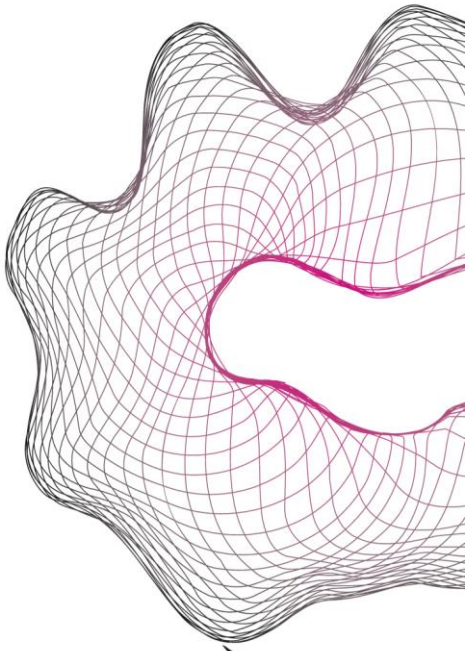
MASTER THESIS



BOOTERS (BLACK)LIST

Justyna Joanna Chromik

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE (EWI)
CHAIR: DACS (DESIGN AND ANALYSIS OF COMMUNICATION SYSTEMS)

EXAMINATION COMMITTEE
Prof. dr. ir. Aiko Pras,
Dr. Anna Sperotto,
José Jair C. Santanna, M.Sc.

27-02-2015

UNIVERSITY OF TWENTE.

Abstract

Distributed Denial of Service (DDoS) attacks are a continuously growing threat of the present Internet. Without proper protection, any machine connected to the Internet can be made unavailable to its intended users under a DDoS attack. Those attacks used to be performed only by people who had sufficient knowledge and resources to do that. Nowadays it can be done much easier than that: anyone can simply purchase a desired attack on the requested target, using a website, called a *Booter*. There are dozens or even hundreds of active Booters present on the Internet, and it is possible for anyone to access any of them and perform an attack on anything connected to the Internet. In order to tackle this problem, some researchers analyse the functioning of a handful of Booters - based on leaked databases of those Booters, however, no clear mitigation is proposed so far. To address this gap, this thesis proposed blocking the access to Booters by means of a blacklist. To achieve that, not affecting benign websites, the list needs to be accurate. To be effective and up to date, the list has to be automatically generated and maintained. Therefore, this thesis explained the methodology to classify a website as a Booter, which was used in an automated way to generate a Booters blacklist.

History has taught us: never underestimate the amount of money, time, and effort someone will expend to thwart a security system. It's always better to assume the worst. Assume your adversaries are better than they are. Assume science and technology will soon be able to do things they cannot yet. Give yourself a margin for error. Give yourself more security than you need today. When the unexpected happens, you'll be glad you did.

Bruce Schneier, 1997

CONTENTS

CHAPTER 1

INTRODUCTION

Distributed Denial of Service (DDoS) attacks are considered number one operational threat on the Internet [1]. DDoS attacks aim to make a target machine, service or network unavailable to its intended users, using distributed sources, such as infected computers or servers. For many industries, such as gaming, e-commerce, online financial services, a DDoS attack could be especially devastating, since they use the online services as the one and only way of sales or customer communication [2]. The consequences of these attacks are for example revenue losses in case of online businesses, reputation damage, customer attrition [2, 3].

Within a year, the amount of reported DDoS attacks has increased with 57%, compared to 2013 [4]. One of the reasons for this increase is the effectiveness, simplicity and availability of websites that provide DDoS attacks as a paid service, called *Booters* [4]. These websites offer attacks for a very cheap price, for instance, less than 5 USD [5], powerful enough to put offline most of small and medium-sized companies' websites [6]. Booters are especially popular among online gamers - to knock their opponents offline in order to win a game [7]. However, small gaming websites and private servers are becoming a popular target too [6]. DDoS offered as a commercial service via a website was already described by Lackey in 2010 [8]. From then on Booters are flourishing: they evolve, acquire new customers but also new Booters appear on the Internet. Booters are available for anyone - this means that DDoS attacks are no longer associated with hackers with knowledge. Since anyone can currently use or make those websites, it is crucial to investigate what can be done to decrease the amount of DDoS attacks performed by means of Booters.

There are three areas where Booters can be mitigated: at access, infrastructure, and target level [9]. This thesis focuses on the access level, since it was not yet investigated and has a huge potential to tackle the problem of Booters, because it treats the origin of the attacks. For such this thesis proposes to create a blacklist of Booters. Four challenges have to be addressed when proposing a blacklist. First of all, blacklisting websites brings *objections*, since in the era of net neutrality any website filtering gains its opponents. Although a benign URL or IP address could be accidentally blocked, blacklisting is still widely used for filtering spam messages, phishing websites or websites used for spreading malicious software. Second, because the mentioned phenomena are dynamic, these lists need to be *automatically generated* in order to save time and resources. Third, in order to use a list for filtering, blacklists used in the mentioned areas need to be *accurate*. This means that a list of filtered websites has to contain no benign websites, in order to avoid overblocking. Finally, *maintenance* of such list means not only to add the new examples of Booters, but also to remove old entries. In other words, maintenance keeps the list up-to-date.

Knowing the problem and proposed solution, the remaining of this introduction describes the goals of this thesis in Section 1.1, highlights the contributions in Section 1.3 and explains the structure of the thesis in Section 1.2.

1.1 Goals

The goal of this thesis is to create an accurate and automatically retrieved list of Booter websites. Therefore, to drive this research the following main research question (RQ) is used.

RQ: How to classify a website as a Booter?

To answer this research question four main challenges arise (i) to properly understand the phenomenon of Booters, (ii) to have a set of accurate search terms to retrieve websites to be classified, (iii) to characterize the Booters, and (iv) to determine the features that differentiate Booter websites from any other

website. Therefore, to address the main research question and the challenges imposed, a set of sub research questions is defined as following:

- RQ.A What is the state-of-the-art of Booters and their mitigation?
- RQ.B What are the proper search terms for searching for Booters?
- RQ.C What are the characteristics of Booter websites?
- RQ.D What is the set of features that could be used to classify a website as a Booter?

The next section will explain the approach which was taken in order to answer the questions above.

1.2 Structure

In order to answer the questions stated above, the research was conducted as depicted in Fig. 1.1. First, step 1 provided the state-of-the-art of Booters, their mitigation, but also the set of features that can be used to classify any website. This will be covered in Chapter 2. Secondly, Chapter 3 explains the way to retrieve as many URLs related to Booters as possible (step 2) and all the necessary information that could be used to classify a website (step 3). Based on the found websites, the most significant characteristics were defined in step 4. This will be explained in Chapter 4, as well as validated (step 5).

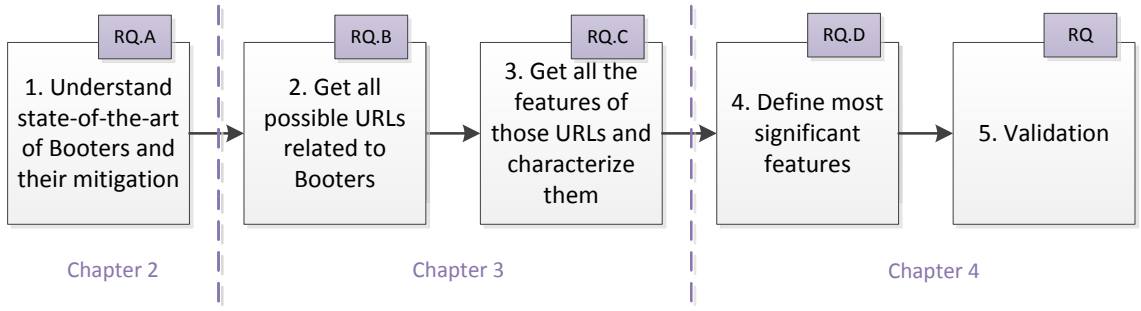


Figure 1.1: The approach to classify a website as a Booter.

The thesis is concluded in Chapter 5 together with suggestions for future work. Finally, it is wrapped up with some reflections and acknowledgements, after the conclusions.

Next section highlights the contributions of this thesis.

1.3 Contribution

This thesis addresses the four main questions asked in Section 1.2 and the additional challenges mentioned at the beginning of this introduction. As a result, the following contributions can be listed as an outcome of this thesis:

- To avoid the *objections* against blacklisting Booters, the problem of those websites is explained and the state-of-the-art of Booters is researched. It showed that both types of websites offering DDoS-As-a-Service and stress testing can be used for similar purposes - performing DDoS attacks - and should be considered a threat;
- To address *automatic creation* of a list, a crawler gathering relevant information related to Booters is developed. For this a proper set of *search terms* is defined. The dataset collected for purpose of this thesis is available in Appendix B for all interested researchers. Moreover, the process of updating the list is continuous, therefore the challenge of *maintenance* is addressed by the crawler as well;

- By careful investigation of websites *characteristics*, a set of proper *features* to classify a website as a Booter is defined;
- A framework to *classify* a website as a Booter is proposed based on the previously mentioned features. The goal of the proposed solution is to be as *accurate* as possible.

Beside this thesis, as a deliverable of this research a paper was written **"Booter website characterization: *Towards a list of threats*"**, and submitted to the 33rd Brazilian Symposium on Computer Networks and Distributed Systems Conference (SBRC) 2015 in Brazil [1]. The conference takes place in May 2015, and the further notice about the conference will be given in March 2015. The paper submitted for the conference is attached in Appendix A.

[1] http://sbrc2015.ufes.br/

CHAPTER 2

BACKGROUND

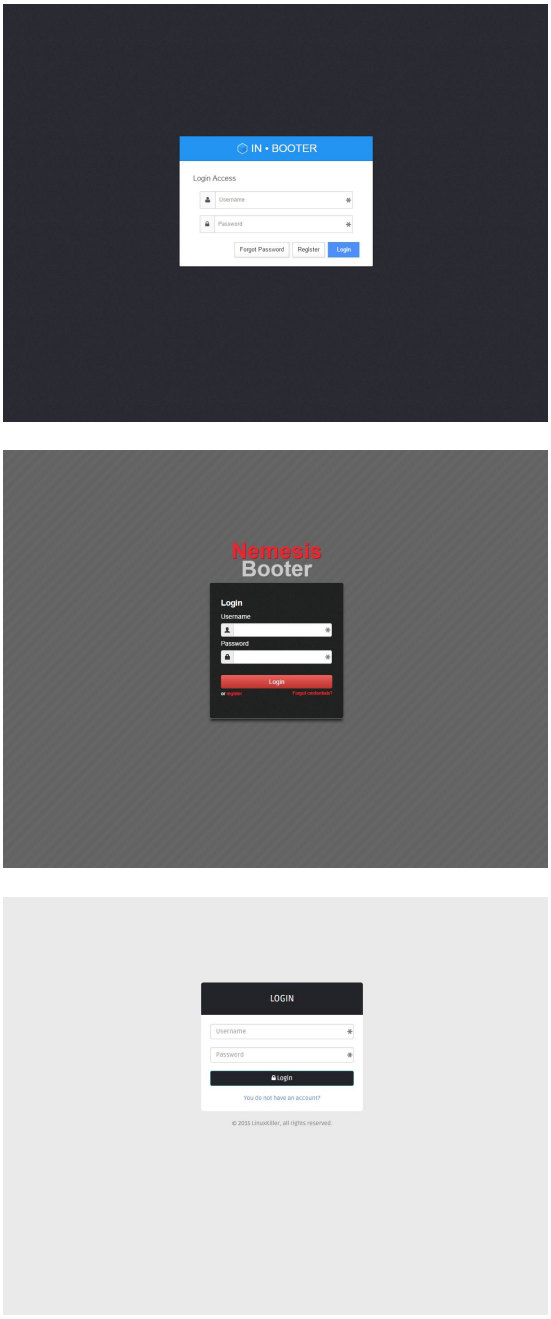
To understand the subject of Booters, this chapter investigates what is the state-of-the-art of Booters mitigation. This is done by analysing currently available literature about Booters, as well as blogs of security experts, and hacker forums. Based on such input, Booters are defined, and the existing methods for their mitigation is investigated. Section 2.1 gives a definition of a Booter, and explains the difference between a Booter and stress testing website. Section 2.2 explains the known mitigation methods that could be applied in tackling the problem of Booters. With the idea of characterizing a Booter website, this chapter is closed with a review on possible methods and features used to classify a website in Section 2.3.

2.1 What is a Booter?

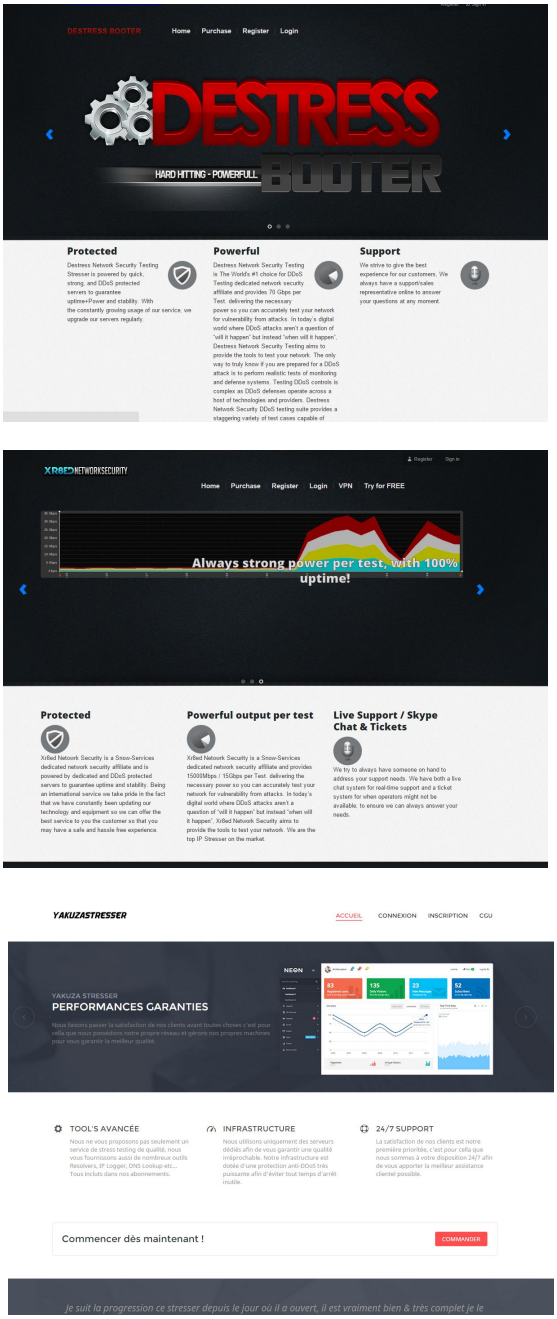
Karami and McCoy [5] refer to Booter services as the low-cost "DDoS-as-a-Service" phenomenon. Santanna et al. [9] considers Booters to be the front-end to access and launch DDoS-as-a-Service attacks. A security specialist, Krebs [10], refers to Booters as services hired to knock Websites and individual Internet users offline. Booter services are also mentioned in reports of leading DDoS protection companies, such as Akamai [4]. Such services are defined there as part of the DDoS-for-hire market allowing low-level, non-technical actors to threaten organizations without DDoS protection [4]. The main message in all the mentioned sources is that DDoS attacking is available to everyone nowadays. In the past, in order to perform a DDoS attack one needed to have specialized knowledge and resources to perform it. Booters made it easy - without any knowledge, for just 1 USD [11] anyone can perform a DDoS attack. In this thesis, name "Booter" means the following:

Booter website (Booter): a website which provides means for performing DDoS attacks for money.

Since it is so easy to access and use Booters, the question arises, how do they look like? The Figure 2.1 shows examples of a basic interface of a Booter.



a) Simple login page examples of Booters.



b) Booters with extended interface.

Figure 2.1: Examples of Booter websites.

In Figure 2.1 it can be noticed that Booters look a lot alike: in the left column, Fig. 2.1a) the main page of Booters is quite simple: they are built only from a login screen. In the right column, Fig. 2.1b) more information can be noticed, such as explaining that servers used are protected to ensure 100% time availability, or that the service they offer is powerful.

Regarding how powerful they are, it is useful to know what type of attacks are offered by Booters. Booters offer mainly two categories of attacks: layers 3&4, and application layer attacks [5]. Layer 3 and 4 attacks exhaust the network resources by overflowing the bandwidth of a link, or target a specific host by exhausting resources of such host [12]. Application layer attacks target a given application on the victim system, by exhausting the resources of such application only, meaning that other applications on

the attacked host continue working [12]. Among the attacks of layer 3 and 4 attacks offered by Booters, the following are the most common:

- **SYN flood attack** - where attacker sends a series of SYN requests to the target in order to exhaust the server resources. Due to many half open sessions, the target will be not able to process new requests, which could come from a legitimate source. The source IP sent by the machines is often spoofed, what makes it almost impossible to block the source machine. This attack comes under different names, what Booters list as e.g. TCP attack, SSYN (Spoofed SYN), ESSYN.
- **UDP flood attack** - where the victim target receives a lot of UDP packets, which it has to respond with many ICMP packets. As a result it will be overloaded and not able to receive more requests. The source IP can be as well spoofed, thus the name SUDP (Spoofed UDP) is also used.
- **DRDoS (Distributed Reflection Denial of Service)** - a special type of SUDP attacks. In this attack, forged requests are sent to large number of computers, which will reply to such request. By changing (spoofing) the sender address to the victim address, all the response is sent to the that system, exhausting the bandwidth. Those attacks are additionally amplified. Amplification is done due to fact that with short in size request, such a server can provide a long in size response. This is possible with the protocols like DNS (Domain Name Server), NTP (Network Time Protocol), Chargen (Character Generator), or SSDP (Simple Service Discovery Protocol). In case of DNS protocol, using open DNS resolvers, the request can be amplified with up to 179 times [13], while using NTP servers, with 556,9 times [14].
- **UDP lag** - which is mostly an amplified UDP attack, that does not send the traffic constantly. Instead, it makes some breaks, e.g. sending the traffic for 3 seconds and then 1 second break. The goal of such attack is to slow down ("lag") the victim's system instead of blocking it entirely.

Especially the DRDoS attacks are one of the most powerful attacks nowadays. Among others, they were used in the 300Gbps attack on SpamHaus in 2014 [15]. Although the attack on SpamHaus was not done by means of Booters it just gives the idea of their potential. The amount of NTP reflection attacks has increased 181% within the last quartile of 2014, and is confirmed that a source of the attacks were Booter websites [4]. As said before, Booters also offer attacks on the application layer, such as:

- **GET attack** - an HTTP based attack, based on a large number of GET requests, usually sent via proxies.
- **POST attack** - an HTTP based attack. The POST attack is based on sending complete HTTP POST headers with specified size of the message body following. That message is then sent at an extremely low rate in order to slow down the system.
- **Slowloris** - an HTTP based attack, tries to keep many connections to the victimized web server opened for as long as possible. Connections to the target system are opened, and a partial request is sent. Once in a while, it sends a new HTTP header, which adds up to the request, but which is never completed. This attack can be compared to SYN flood but on the application layer.
- **RUDY (R-U-Dead-Yet?) attack** - similar to Slowloris, but makes the sessions halt by using never ending POST requests and sending large value of header.
- **ARME attack** - exhaustion of SWAP memory on Apache Web servers. Once out or SWAP memory, the machine will kill all the processes.

There are more attacks listed among Booters, however, most of them repeat – they are just listed under different names. Overall, although there are many types of offered attacks, not all of them are always achievable. For instance, [11] shows that not all of the Booters deliver the requested service, e.g. among 14 Booters analysed in that paper, 5 did not provide the bought service, out of which 3 did not provide anything at all. On the other hand, 9 of them delivered DRDoS (DNS-based and CHARGEN-based) attacks, while UDP attacks were bought.

To understand how Booters get their resources in order to offer the DDoS as a paid service, some insight about their infrastructure is needed. In the papers [5, 9] it is pointed out that the infrastructure

and market of Booters is not known. The paper by Karami and McCoy reveals the infrastructure used by the one specific Booter, named *TwBooter*. Although in this case 15 distinct servers were used, it is a common practice that Booters use a number of compromised machines (e.g. botnet and Web-shells), and misuse UDP servers (e.g. DNS or NTP) for the DRDoS attacks [9]. More light is brought by Santanna et al. [16], where databases of 15 distinct Booters were analysed. As compared with research of Karami and McCoy [5], this was the only case where dedicated servers were used. All the other 14 analysed Booters had their infrastructure based on *Web-shells*, that are scripts hosted on some machines, which could be compromised.

The literature emphasizes that anyone can perform attacks by means of Booters [5, 11]. It is possible, because Booters additionally offer the following services:

- Skype resolver, to resolve the IP address of your game opponent, whose Skype login is known.
- IP address resolver, to resolve the IP address of a website URL, in case some website is the target.
- Cloudflare resolver, to resolve the real IP address of the website protected by a DDoS Protection Service (DPS) company.
- Live Chat Support, to guide the potential customers through their services.

Using Booters means that users intentionally attack a third party service, against the will of the victimized system. However, as pointed in the literature, Booters publicly name themselves as “stress testing websites” or shortly “stressers”. Using stressers would mean the attacks performed are deliberate. Since the difference between the two is not clear, the next section investigates the difference between “stressers” and “booters”.

2.1.1 Stress testing websites

This section addresses the difference between websites named Booters and websites offering stress testing - stressers. This will be addressed by comparing the goal, target, infrastructure used for attack, and methods of advertising of both. First, stressers will be defined, and then the results of search in Google Search engine for “Booters” and “stressers” will be compared.

Stress testing is a form of intentional thorough testing of a given system. Its goal it to test such entity beyond regular operational capacity up to the limits of the given system, to observe the results. The goal of such testing is, among others, to determine the safe usage limits, verify the specifications of a system or learn how the system fails [17]. One of the goals of stress testing is to check the resistance to DDoS attacks [18] and the most important characteristics is that it is deliberate and done in a test lab (so it does not affect other entities). This implies that stress testing websites should make use of a dedicated infrastructure for the provided services. In this thesis the name “stresser” means the following:

Stresser website (stresser): a website by means of which it possible to deliberately perform a thorough test of a system under test.

Although there are many stress testing websites, they do not ensure any identity verification of the person requesting the attacks. As mentioned before, Booters officially claim to offer stress testing, however, as the reviewers and bloggers point out, this is only to avoid legal aspects [8, 19, 20]. Truth is, technically they perform the same resource exhaustion actions as the stress testers, however, main difference is: the action done using Booters’ service is unauthorized by the target.

Moreover, the resources of Booters (advertising themselves as stressers) used for attack do not belong to a test lab, as often open DNS resolvers or NTP servers are used, as mentioned in Section above. Also, Booters recommend not to target own infrastructure [9]. If a customer would like to perform stress testing on their system, this would be required. This requirement would imply a need for some verification, but this is currently not done among the Booters or stress testing websites.

The comparison of Booters and stressers will be concluded by comparing the services offered by Booter and stresser respectively and the characteristics of them mentioned above.

2.1.2 Booters vs stressers and a box of matches.

Since many Booters and stressers are available on the Internet, the comparison was be done using just one result of search queries for each search word. This result can be thus biased by that result. To make the comparison, the top one result for the search terms “booter” and “stresser” is analysed. The URL/names of the Booter and the stresser are anonymized to avoid legal or ethical implications (the URLs are revealed in the Appendix C). Table 2.1 shows the findings. Please note that although attacks such as TCP, SSYN and ESSYN, or CHARGEN and DRDoS are technically the same as it was explained in Section 2.1, the table shows how the attacks were advertised in the websites.

Attack types		Booter A	Stresser B
Layer 3&4	TCP	✓	
	SSYN	✓	✓
	ESSYN	✓	
	UDP	✓	✓
	UDP-LAG	✓	✓
	DRDoS		✓
	CHARGEN	✓	
Layer 7	GET	✓	
	HEAD	✓	
	POST	✓	
	RUDY	✓	✓
	ARME	✓	✓
	SLOWLORIS	✓	✓

Table 2.1: Services offered by a Booter and a stresser

Looking at the Table 2.1, three observations can be done. The first observation is that technically the Stresser B offers the same attack types as Booter A. The second observation is that both offer most of the known types of DDoS attacks, which target the Layers 3&4 and the application layer. The third observation is that although Booter A does not explicitly offer Distributed Reflection Denial of Service (DRDoS), CHARGEN is a type of DRDoS. Therefore, both Booter A and Stresser B offer the strongest type of DDoS attacks reported nowadays.

Notice that the Booter A offers more attack types, but as often it is mentioned in reviews [21], behind different names the same type of attacks are hiding. For instance, as already pointed out, TCP, SSYN, ESSYN are all the same type of attack. In other words, Booter A would like to be “more appealing” by offering more options, while the truth is, only one or two types of attack are possible by means of this Booter.

To fully answer the question, whether there is a difference between stressers and Booters, additional characteristics are gathered in the Table 2.2. By manually checking Booter A and Stresser B more information about the target and used infrastructure was gathered. It was noticed that Stresser B has no restriction in relation to the target of “stress test”. Given this freedom, the customers can perform attacks against third party services.

The characteristics gathered in Tab. 2.2 suggest that there is no technical difference between Booter and stresser. Both types of websites are in practice used for performing DDoS attacks, do not verify the ownership of targeted system, do not request any permission for performing the attacks. Moreover, the infrastructure used for the attacks in both cases does not differ. Either Booter or stresser can be (mis)used as an alternative for the same purposes - since they provide service of DDoS attacks, they will be considered as a threat. All these observations lead to a conclusion that all the stressers will be considered Booters in the rest of this thesis.

The next question that arises is: why Booters advertise themselves as stressers? The answer, which can be found in hacker forums and many blogs is that Booters want to avoid legal problems by hiding illegal actions (DDoS attacks using compromised machines) behind legal services (stress testing) [20, 22, 23]. Booters have another strategy to avoid legal problems. Instead of advertising themselves as stressers, they include in their websites Terms of Services (ToS). It is a set of rules one need to follow in order to use a service [24]. It is a legal agreement between the user of a website and the owner and it often contains a disclaimer clarifying the website’s legal liability for customer’s actions. Table 2.3 shows

Characteristic	**Booter**	**Stresser**	
		Theory	*Practice*
Goal	Performing DDoS attacks	Testing a target system beyond its operational capacities	Offering DDoS attacks of layer 3, 4 and application layer
Target system	Third party service	Own infrastructure	No ownership verification - possible targeting third party service
Permission	Against the will of the third party	Deliberate	No permission requests
Infrastructure	Using botnets, dedicated servers, NTP servers, or DNS resolvers	Done in a test lab	Mention using public DNS resolvers

Table 2.2: Comparison of characteristics of Booters and stressers: in theory and in practice

parts of ToS found on Booters. It gives example of three different Booters, which are named below as C, D and F, and piece of text included in their Terms of Service. As neither the ethical nor the legal aspects of Booters are clear, they are also anonymized [1].

	Terms of Service statements
Booter C	"We are not responsible for how ever you use this stresser"
Booter D	"Illegal activity which occurs in your account is [...] associated to you"
Booter E	"Anything you do while on Booter E is your own responsibility"

Table 2.3: Examples of text included in Terms of Service

Table 2.3 shows that some Booters clearly do not take any legal responsibilities for users' actions, although their core business is to offer DDoS attacks against anyone and anything connected to the Internet. This type of action can be metaphorically compared to a parent leaving a 4 year old child in a stable filled with hay, giving it a box of matches and saying: *"Whatever you do with these matches is up to you. I do not take care the responsibility for how you use these matches"*. Replace the stable with the internet and hay with all the hosts connected to it, the child with the attacker and the matches with Booters, and you will understand the business model of a Booter.

The legal aspect of Booters is a nudging subject, which requires attention, but will be out of the scope of this research. From technical point of view it can be concluded that stressers are/can be used as Booters.

As mentioned above, for the purpose of this thesis all stress testing websites and Booters are treated equally as a mean for performing DDoS attacks. Because they provide such possibility, they could be misused, and due to that, they require attention. Due to that, the next section will address the possible methods of mitigating Booters.

2.2 Existing mitigation methods of Booters

This section reviews the existing methods that are or could be used to mitigate Booters. There are several levels where, in general, DDoS-As-a-Service phenomenon could be mitigated [9]. Figure 2.2 shows the basic work of a Booter. As depicted, ill-intentioned user is accessing Booter to attack the victim system. The Booter uses its infrastructure, which can consist of, for example, dedicated servers, botnets, open DNS resolvers etc. [6], depending on the type of the attack which was requested by a customer. By means of this infrastructure, an attack is performed to exhaust the resources of the victim system.

Reducing the impact of the attacks caused by Booters could be done on several levels, such as (i) the access level, (ii) the infrastructure (command control) level, and (iii) the target level. In the Fig. 2.2

[1] The names are revealed in the Appendix C.

the access level is depicted as the part between the ill-intentioned user and the Booter, symbolizing all the interactions between the user and the Booter. The infrastructure level, is symbolized by the Booter website and the controlled systems - depending on Booters. Target level is shown by the victim - where the traffic from earlier mentioned systems is reaching its destination. The following paragraphs will describe in a reversed order how Booters can be mitigated per level.

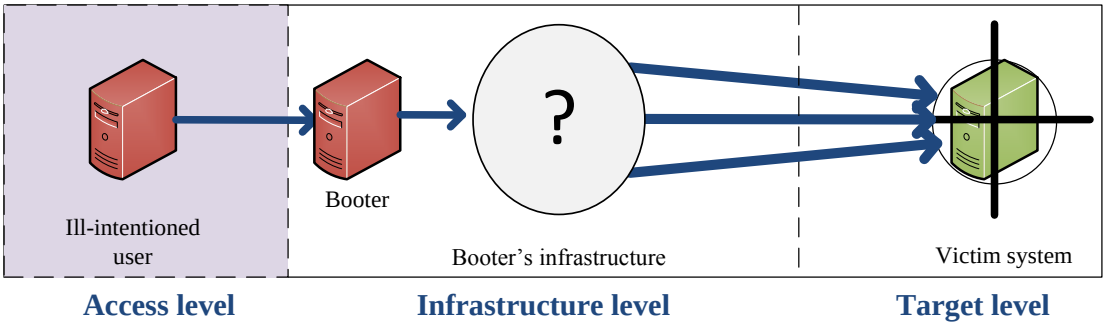


Figure 2.2: Mitigation areas of Booters.

- **Target level** Among the three mentioned levels of mitigating the impact of Booters, the generic DDoS defence mechanisms done at the target are already widely researched [25, 26]. In case of the large attacks they are however not effective [15]. A very neat overview of the defence mechanisms is described by Mirkovic and Reiher [12]. In brief, two main approaches are defined: *preventive* and *reactive*. Preventive mechanisms are all the actions taken beforehand in order to prevent an attack from happening. It could be, for instance, keeping the software up to date, patching the servers to avoid the known possibilities of misconfiguration. Reactive mechanisms are all the actions taken to stop an existing attack, after detecting it. The detection can be done either by detecting a pattern of an attack, or an attack can be concluded as an anomaly from the regular behaviour of e.g. the traffic. The actions taken to stop such an attack are for instance, rate-limiting or filtering out the malicious traffic. The DDoS mitigation mechanisms at the target level could be also outsourced to a third party, that is, a DDoS Protection Service (DPS) company such as CloudFlare [2] or Incapsula [3].

  At the target level it is not possible to differentiate whether the attack was caused by means of a Booter or by a hacker with advanced knowledge, unless the attacks were launched deliberately. The IP address of sending machine belonging to the Booter's infrastructure may be spoofed, that means, the IP address of sending machine can be changed to any IP address, so it would not known where the attack comes from, or who sent the packet. The reason why this area is so popular is that, especially in case of the volumetric attacks, it is the area where the attack is detected in the easiest way. Since the sources of traffic are distributed, the traffic generated around them has not as high load. However, the closer to the destination, the higher load of the traffic, thus, the easier is it to detect it.

- **Infrastructure level** The mitigation of Booters on the infrastructure level is the most difficult, because of vague information about it. Analysis of this level was done by few researchers [5, 16] respectively by investigating the infrastructure of 1 and 15 Booters, based on leaked databases. In order to propose a mitigation method for this level a good understanding on the infrastructure is needed. However, in the findings of Karami et al. [5], the authors mention that in the period of 23rd January till 15th of March 2013 15 distinct servers were used, not all at the same time. Since only three servers were active throughout the whole period, this means that the infrastructure of Booters is varying, and because of that, hard to address. However, the Booter explained by Karami et

[2] https://www.cloudflare.com/
[3] https://www.incapsula.com/

al. is apparently an exception when it comes to the used infrastructure: most Booters base their core on *Web-shells* [16], which are scripts hosted on machines and are accessed via HTTP/GET or HTTP/POST. Those machines can be compromised (or not), and the HTTP requests specify details of an attack, e.g. the targeted host, duration time, or a type of an attack. To avoid being dependent on a leaked database of a Booter, this research will not address this level of Booter mitigation.

- **Access level** Since one of the reasons for the popularity of Booters is that they are available via a user-friendly website, this could be a possible way of mitigating them. It was decided to investigate the known methods used for limiting access to websites, namely, blacklists. Blacklisting is a popular solution for blocking some machine or website, on DNS or application level. There has been quite some research done in area of generating blacklists and using blacklists for blocking the spam, phishing websites and websites with illegal content, such as child pornography [27, 28, 29, 30], however there is no research done about using them in blocking the access to Booters.

Due to the reasons mentioned above, the rest of this thesis will focus only on the blacklist of Booters as a possible mitigation method. For that the next section will explain the goal, drawbacks and advantages of a blacklist.

2.2.1 Blacklisting Booters

Main idea behind a blacklist is that an attacker willing to boot some website will not be able to do it in simple way, because he or she would not be able to access a Booter. Another idea is that instead of blocking the website entirely, a warning could be generated, saying e.g. “The website you are about to access is associated with use for illegal purposes.”, what could make the person change his mind. However, blacklists come with some challenges. Below four challenges are listed and explained followed by a counter argument in favour of blacklists.

First of all, blocking websites would require a list that is correct and up-to-date. In case a list is not accurate, then the entities are either wrongly classified as Booters, or wrongly classified as non-Booters. In first case, such list would block benign websites, in the latter case the ill-intentioned users would still be able to access Booters, but not all of them. Also, if the list is not up-to-date, URL which used to belong to a Booter may not be a Booter anymore, yet it would still be blocked. All those cases lead to overblocking, which is one of the arguments of opponents of blacklists. Moreover, blacklists are a reactive mechanism for blocking access to a website. This means, if a blacklist is not updated fast enough, an attacker has time to still use the website, before he cannot access it anymore. The mentioned challenges should be addressed in the method of generation of the blacklist and very often it is: it is possible to create blacklists automatically with a high precision as it is done in case of spam, phishing websites, websites spreading malicious software, etc. For example, researches show that the precision of generating a blacklist automatically comes with up to 99% precision [28], and also there are proactive methods of creating a blacklist [29].

Second of all, a person who would like to use a Booter could simply use a VPN or proxies to connect to it. This cannot be avoided, but requires some knowledge from the ill-intentioned user. This means, that using blacklists at least the not advanced users of Booters would be eliminated from the list of potential customers. As the research by Santanna et al. shows [16], there are many users that access a Booter using their original IP, so it can be concluded that there are many not advanced users of Booters.

Thirdly, nowadays Web filtering aspects bring a lot of objections from Net Neutrality adepts and is considered unlawful [31]. Somehow the problem of filtering spam or phishing websites is silent in the media. On the other hand, issue of filtering websites with child pornography requires assurance that such content is on such a website, otherwise it is considered illegal to block such website. Still, it is obvious - phishing, so identity theft is a crime, but so is DDoS and DDoS attempts, e.g. in the EU law [32]. This suggests that blocking websites allowing to perform DDoS attacks may be in future allowed.

Fourth argument against blacklists is the difficulty to measure effectiveness of such. In case of child pornography, so much content exchange is done in P2P manner, that it is difficult to judge the effects of blocking associated websites in the total spread of this illegal content, simply because it is not known how much child pornography is being spread through in which way [31]. In case of Booters

the problem will remain the same: although the number attempts to connect to Booter websites can be monitored, it cannot concluded whether this way a DDoS attack was successfully prevented from happening. However, despite problems with measuring the effectiveness of a blacklist, it has to be memorized that no security system is ever built of single type of protection. It can be said that security systems have to be like onions - have to be built of layers. Of course, despite filtering, some spam still reaches the mailbox of a victim, or some people still access phishing websites. However, this would be much more common if spam or phishing websites were not filtered.

Summarizing, despite the arguments against blacklisting, it is still a promising way of mitigating Booters. However, in order to use a blacklist, a verified list of such websites is needed in the first place. Currently there is no freely available list of Booters for the research purposes. Such list would be a great help for future research, for example, by monitoring the traffic coming to such website. For this reason, although the main idea of this thesis was to use a list of Booters as a blacklist, it is decided to change the title to “Booters (black)list”. The brackets suggest that the primary goal of this list is not to blacklist Booters, but to generate a list, which could potentially be used as a blacklist.

The list of Booters will keep changing, since Booters change their names, and new ones appear on the Internet. To get an idea if such list can be generated automatically for Booters, the next section will provide an overview on the current methods to characterize a website, in order to automatically classify it.

2.3 Characterizing a website

The question addressed in this section is: what are the existing methods for characterising a website? First, a literature review on website features is done. The purpose of such classification is also analysed, as well as the approach taken by the researches and the method the classification. Next, the features used in the literature are explained.

2.3.1 Purpose and approach

To characterize Booters, first a review on how to characterize websites in general is done. A literature survey on website features, gave an overview of website elements that could be used for classification. The papers discussing the features of websites have different purposes, which are gathered in Tab. 2.4. Note that this survey is not complete: there is much more work done in website classification, however the used features repeat. Due to that this thesis focuses on few main ideas, where some website features are used to distinguish a website.

Paper	Purpose	Accuracy
[33]	Defining the type of website (Academic, Blog, Corporate, Personal and Shop)	82%
[34]	Defining the type of website (Arts, Business, Computers, Games, Health, Home, News, Recreation and Reference)	70%
[35]	Defining the class of website (Academic, Blog, Community, Corporate, Information, Nonprofit, Personal and Shop)	92%
[30]	Detecting phishing websites	96,79%
[36]	Filtering pornographic websites	95%
[28]	Generating a blacklist of phishing websites	95-99%
[37]	Web page classification	85%

Table 2.4: The purpose, approach and accuracy of classifying websites.

In the Table 2.4 the purpose of classifying websites varies: mostly the goal is to classify the subject or type of a website [33, 34, 35, 37], some of them used the features for website filtering [30, 36], or use the features for blacklist generation [28].

The used approaches mostly consist of machine learning algorithms. Main idea of machine learning is to have representative set of every of classified types of websites (e.g. a set of websites belonging

to “Academic”, “Blog”, etc. categories). Based on a set of features of those websites, a classifier is trained, to later use it to define what class does a new website belong to. Most popular machine learning algorithm among the papers listed in Table 2.4 is Naive Bayes algorithm [33, 34, 35, 28]. In the set of papers presented in Tab. 2.4 it reaches the highest accuracy (99%). Another possible approach is creating a decision tree [36] or calculating a disparity measure of the actual website and the fake website [30].

In either way, any method of classification requires a reference dataset in order to define whether the decisions done either by machine learning classifier, or by the decision tree etc. are correct or not. For this such set has to be proposed in this thesis. To understand properly what is understood by correctness of decisions based on the datasets, the next section will explain the idea behind classifying websites.

2.3.2 Classification of websites

Any of the mentioned algorithms lead to correct and wrong results. Verification of such result is done by calculating the sensitivity and specificity rates. Provided a set of websites belonging to a set of classes, the hypothesized class is compared with the actual one. In case such classification is used for filtering websites, imagine the scenario: a website can be classified as malicious and be filtered out, or as benign and by allowed to be accessed by the user. Thus, the hypothesized class of a website is malicious or benign. This gives four possibilities, as shown in Fig. 2.3 as *error* or *confusion matrix*. These possibilities are explained below:

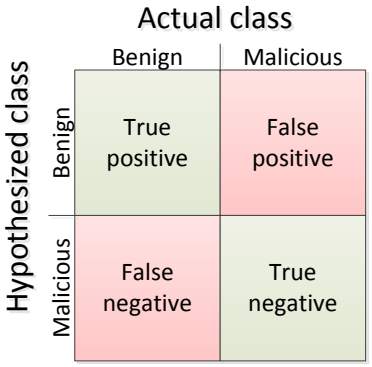


Figure 2.3: Confusion matrix

- **True positive** - when the website is classified as benign and it actually is so.
- **True negative** - when a malign website is blocked.
- **False positive** - when the website is benign and is classified as malign. This is called a Type I error.
- **False negative** - when the website is classified as malign, however it is benign. This is called a Type II error.

Knowing the actual and hypothesised class, the algorithm is most precise if there are as many as possible of true positive and true negative assignments, and as least as possible of false positive and false negative assignments. This way the highest accuracy is reached, what is desired in a good algorithm. The accuracy of used algorithms is calculated by dividing the total sum of true positives and true negatives by the total population of the set, that is all the cases. In case of classifying the subject of website, the error is not harmful. However, in case of website filtering it means that a wrong decision leads to blocking benign websites, what as was pointed out in Section 2.2, is not desirable.

Note, that in order to calculate the accuracy, the analysed URLs need to be compared to a reference list. In case of the papers mentioned above, the authors use existing databases where the websites are classified by some communities. For example, in case of classifying phishing websites, databases such as PhishTank [30, 28] or Spamscatter [28]. In case of classifying the type or subject of a website, DMOZ Open Directory Project [33, 34, 35], WebKB and BankSearch [37]. However, in case of Booter websites, such manually verified list does not exist. The possible reasons for that are: Booter phenomenon is relatively new, Booters are so far not clearly defined, and the constant change of name and number of Booters. This means that the chosen approach for classification of Booters will have to take onto account that all the URLs will need to be manually classified.

The next subsection describes the features of websites which are used for classification.

2.3.3 Features

The literature referenced in Table 2.5 refer to different elements of a website. Below the table it is explained how the mentioned features are used in different approaches.

Paper	URL	IP	Website structure	WHOIS	Website content		
					Textual	Meta	Visual
[33]			x			x	
[34]	x						
[35]	x		x				
[30]	x			x	x	x	
[36]			x		x	x	x
[28]	x	x	x	x			
[37]							x

Table 2.5: Website features used to characterize websites.

As it can be seen in Table 2.5, the listed papers were mentioning properties of the URL, IP address, the structure of the website, WHOIS information of the domain and the textual content of the website. Below each of them is briefly explained:

1. **URL:** this feature discloses the overall composition of a URL. Usually a URL is composed of three elements: (i) network application protocol, (ii) the URL host name, and (iii) the URL path name. For example, in *http://www.domainname.tld/path/to/the/article.html*, 'http://' is the protocol, 'www.domainname.tld' is the host name, and '/path/to/the/article.html' is the path name. In the literature the URL is used as a feature because it can be obfuscated, or plain IP address can be used instead of a domain name [30]. The websites could also be using certain string in the URL [34, 28], the presence of digits in the URL can be a feature [35].

2. **IP address:** this feature can provide the following information: (i) the IP address could be already blacklisted or (ii) the geographical information - where the website is hosted [28].

3. **Website structure**: is a feature that reveals, among others, two aspects: (i) the URL depth level and (ii) the number of known pages of a website [33, 35]. The former is a terminology defined by us that analyses the number of slashes ('/') a URL path has. For example, the URL *http://www.domainname.tld/path/to/the/article.html* has depth level '4', because the slashes in the protocol ('http://') are not counted. The latter aspect is the number of indexed webpages that have the same host name and are reached by Google Search engine. For example, the number of indexed pages is 2 if Google Search returns 2 pages that contain a same *www.domainname.tld*, such as *www.domainname.tld/1.html* and *www.domainname.tld/2.php*. Additionally, the literature suggests that e.g. number of dots ('.') to be significant [38] or the hyperlinks within the page [36].

4. **WHOIS:** WHOIS is a query/response protocol used to provide information about domain names to Internet users. The protocol delivers information such as a domain name, an IP address, or an autonomous system, in a human-readable format [39]. WHOIS record can provide information

such as (i) the registration date, (ii) the owner, (iii) the nameservers related to that domain name, and (iv) the entity responsible for the domain registration (i.e., registrar) [28], or it could be poorly maintained (missing data) or even not retrievable [30], what also can be a feature of Booters.

5. **Page content**: is a feature that reveals the elements of a website such as (i) the textual description, (ii) the meta data, and (iii) the visual content, (e.g., buttons, tables, and figures). Websites are often classified based on text - they are compared according to frequency of a word in a site - in a *"bag-of-words"* approach. It could be either text of the content [40, 36] or of the meta data [33, 36]. The visual content of websites is compared either by analysis of pictures on the website [36] or by analysing e.g. forms, buttons, tables, which and their position on a website [37].

When comparing the Table 2.4 with Table 2.5 it can be noticed that if the purpose of classification is website classification, the features such as IP address or WHOIS information do not play important role. However, features such as the URL or website structure are interesting in both cases. Moreover, the papers with most analysed features [28, 30, 36] reached the highest accuracies. This suggests that it is worth to investigate many different features in the classification for better precision.

As shown above, there are many features of websites which could be possibly used to classify a website. Depending on the goal of the classification, or specific characteristics of classified websites, different features play more important role. As shown in Section 2.3.1, the websites can be classified with up to 99% accuracy. Important message from this section is that such classification is possible and done with satisfactory accuracy. In characterising Booters precision is also important, so as many features as possible have to be investigated in order to determine if Booters can be characterized in a general way.

2.4 Concluding remarks

As the background research showed, the state-of-the-art of Booters mitigation is vague and requires some attention both from legal and scientific point of view. This was described in Section 2.1. Leaving out the legal point of view, it was shown that Booters are a threat and do not have any dedicated mitigation methods so far, as it was explained in Section 2.2. When analysing the mitigation areas it was suggested that the possible way of addressing Booters mitigation would be by means of blacklisting. However, in order to propose such technique, a comprehensive, up-to-date list of Booters is needed. Such a list is currently not available for research purposes [4]. Moreover, as it was also explained in Section 2.1, Booters constantly change, because of what such list would have to be generated automatically. Such automatically generated lists are existing and are used for blacklisting spam or phishing websites.

Section 2.3 provided an overview on the features of websites used in approaches for automatic classification of websites. Those features, such as URL structure, IP address origin, website structure and its content, can help to indicate whether a website is a Booter. To find out how the mentioned features can make it possible, the next chapter describes (i) how they were collected, and (ii) what are the main characteristics of websites manually classified as Booters.

[4] It is not at the moment of research: as a result of this research such list is available and listed in Appendix B

CHAPTER 3

THE CRAWLER AND THE DATASET

Previous chapter explained what a Booter is and why is it important to mitigate them, described possible areas of Booter mitigation and provided a literature study on the methods for website classification. The goal of this chapter is to characterize Booters, in order to see whether it is possible to distinguish them from benign websites. This chapter explains the methods taken to retrieve the required information and the data set that it was decided to focus on. Section 3.1 explains the crawler which was developed in order to retrieve the features mentioned in previous chapter. Section 3.2 explains the findings about the Booter websites: what were the most significant characteristics of those websites. Section 3.4 explains the rest of the gathered data: the set based on which the classification is validated in the following chapter.

3.1 The crawler

In order to find a list of search terms, with which Booter websites can be retrieved, the most obvious term: *"booter"* was used in the beginning. With such term many URLs related to the word *"stresser"* were found. In section 2.1.1 it was explained why both types of websites: "booters" and "stressers" will be considered in this thesis as the same type of websites. To find other terms Google was queried for materials related to Booters and stressers. Through an extensive literature study other two terms were found: *"ddos-for-hire"* [4, 41] and *"ddoser"* [42]. Finally, the set of search terms is closed with the term *"ddos-as-a-service"* [5], as mentioned in Section 2.1. Note that the reference on the side of earlier mentioned terms is not necessarily the first to use/define these terms, but the place that they were found.

Having the five search terms (*"booter", "stresser", "ddos-for-hire", "ddoser", and "ddos-as-a-service"*) defined, this section describes the approach to retrieve a list of URLs related to Booters and the additional information needed to classify them. To achieve this feature retrieval in an automated way, a crawler that fulfils the following requirements is developed:

1. Retrieve as many URLs related to Booters as possible based of the list of search terms. These retrieved URLs are called in this thesis as "potential Booters".
2. Extend the URLs retrieved in the previous requirement to include all known pages related to those URL domain names.
3. Retrieve the WHOIS information of the URL domain names.
4. Download the Booter website content.

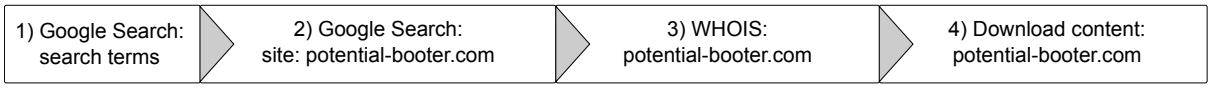


Figure 3.1: Crawler workflow.

The draft concept of the information retrieval is presented in Fig. 3.1, and more detailed it is presented in Algorithm 1. Below it is described how the earlier mentioned requirements were achieved.

In order to retrieve the URLs related to Booters (first requirement), Google Search was used. This was decided for several reasons: since owners of Booters are oriented at earning money, they want their

websites to be easily searchable. Moreover, it is assumed that most of the users of Booters have no knowledge about DDoS attacks, and Google Search engine would be first place to look for this service. Finally, deciding on one method of searching for URLs will save time for analysing the returned results. To crawl Google automatically for the URLs, a python search script is used [43]. This approach was chosen over other approaches [44, 45, 46] because it is not limited on the amount of results able to retrieve (e.g., 24 URLs). The script scrapes all the URLs which are returned in Google Search queries. In standard approach, the script returns all the URLs and also lists some subpages of the website, e.g. when looking for "cat" it would return URLs of all the green marked hyperlinks in Figure 3.2 (numbered 1-5). This means that a lot of hostnames will repeat. For purpose of this research it is more important to receive as many distinct URLs in the search part (that would mean URLs 1 and then 6 in Fig. 3.2). The adjustment was done with newer version of the Google Search function on 30th November 2014.

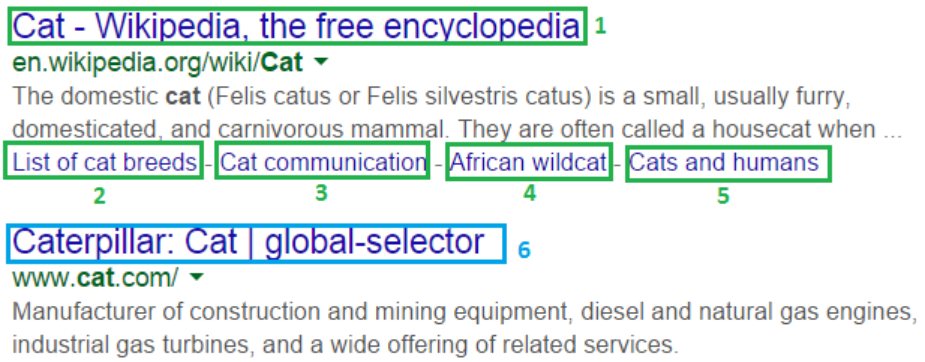


Figure 3.2: URLs scraped with the Google Search function

The method of searching for Booters has limitations, since it depends a lot on the possibility of Google to index those pages. Moreover, another drawback of this method was the time it took to crawl Google - it was required to make random pauses before requesting the pages from Google, otherwise the crawler was blocked. For this reason the overall time of search was long (e.g. one hour for one search word). However, for the purpose of this thesis it is sufficient, since the goal is to classify the website as a Booter, not to focus just on the extensive search. This will be a part of the future works (see Sec. 5.2).

To find all indexed web pages related to a website (second requirement) the Google Search with operator "site:" is used. This operator returns as result only URLs within the domain requested, for example the output of *"site:name-of-potential-booter.tld"* is a list of pages (URL paths) in the domain *"name-of-potential-booter.tld"*.

To retrieve the WHOIS information (third requirement) pythonwhois package [47] is used. Although such library is able to return more than 20 different pieces of information about a domain name, in this WHOIS analysis only 4 of them are used, ones most likely to be retrieved: (i) the creation date, (ii) the registrar of the domain name, (iii) the nameservers that the domain is pointing to, and (iv) the contacts of the domain administrator. Often the data in WHOIS records is poorly maintained, so this will also be noted.

To fulfil the last requirement and download the content of Booter websites, first, it was decided that the identity of the used system should be kept anonymous. This precaution was taken because a security specialist was successively attacked after starting investigating Booters [48]. Therefore, The Onion Router (TOR) network was used. It enables online anonymity by encrypting and bouncing the traffic through networks and open relays severs. There was a drawback of this approach. It was noticed that a lot of websites are not returning the content of it. Since, as it was pointed out in [11], a lot of websites are protected by a DDoS Protection Service (DPS) company, using TOR network made it impossible to automatically retrieve the content of the website. DPS companies generate CAPTCHA challenge if the IP address that the request comes from has a bad reputation [1], as nodes of TOR. Bad reputation of a website means that the IP address that a website is hosted on may be a source of e.g.

[1] https://support.cloudflare.com/hc/en-us/articles/200170116-What-do-the-Threat-Scores-mean-

spam or malicious software. There are many companies which track this kind of behaviour and create a reputation list of the IP addresses. The TOR nodes very often have a bad reputation [2], so the crawler usually gets a CAPTCHA challenge issued instead of getting the first page. It was decided to attempt to fetch those pages after disconnecting from TOR network.

Another requirement discovered when trying to download the potential Booters' websites is to support JavaScript. Since the hosts can determine it in different ways, sometimes looking at the setting in the header of HTTP request: *'X-JAVASCRIPT-ENABLED': 'True'*, and sometimes looking at the USER-AGENT field in that header, requesting it was decided to use a ready browser emulator. In this crawler, the Selenium browser [3] was used, which is a library that emulates a generic Web browser and therefore is able to support JavaScript. In order to run, Selenium needs to launch a browser. With no display on the used machine, the browser was launched using a fake display using Xvfb [4]. Xvfb is a display server allowing to perform graphical operations in memory, without showing any output on the screen.

The Algorithm 1 shows how the data was retrieved: provided the set of search words K and the functions for Google Search and WHOIS search. First, for the set of key words it returns the set of URLs of websites which are potential Booters (lines 1–3). In the next step for the set of mentioned URLs, it uses the pre-defined functions for Google Search and WHOIS search and returns the requested features of websites (lines 4–16). Retrieving the website is first done using TOR (lines 7–9), however, in case the website is not retrieved, the same operation is done after disconnecting from TOR (lines 10–14). This happens when the website protected by Cloudflare does not return the content of the website, but is redirected to a CAPTCHA challenge site.

Algorithm 1 Booter Crawler

Require:
$K = \{k_1, k_2, ..., k_5\}$ set of search words
$search(word)$ the google search function
$whois(url)$ whois function on a url
$W = \{\omega_1, \omega_2, ..., \omega_n\}$ set of URLs of websites

1: **for** k in K **do**
2: $\omega_{...} = search(k)$
3: **end for**
4: **for** ω in W **do**
5: $search(site : \omega)$
6: $whois(\omega)$
7: start TOR, start Browser
8: $get(\omega)$
9: stop Browser, stop TOR
10: **if** $get(\omega) = \emptyset$ **then**
11: start Browser
12: $get(\omega)$
13: stop Browser
14: **end if**
15: **end for**
16: **return** $F = \{f_1, f_2, ..., f_9\}$ set of features for each ω

The search and information retrieval script was running on a Virtual Machine with Debian OS. The information was stored in a relational database on this Virtual Machine. The decision to store the information was made due to the fact that the websites were changing what was influencing the results.
In summary, this crawler was built to attend the requirements to analyse Booters. Although specific libraries and APIs are used, those were examples of decisions on how to fulfil the requirements. Therefore other libraries can be used. Note that the crawler can be extended to perform an automated classification of Booter websites, but first the characteristics of the website features should be analysed.

[2] https://support.cloudflare.com/hc/en-us/articles/203306930-Does-CloudFlare-block-Tor-
[3] http://www.seleniumhq.org
[4] http://www.installationpage.com/selenium/how-to-run-selenium-headless-firefox-in-ubuntu/

3.2 Dataset characteristics

Using the crawler explained in previous section, a search was performed on 15th September 2014. This data set consists of 1238 distinct URLs. Below this set is characterized.

3.2.1 URL analysis

First of all, the URL composition of the 1238 URLs is analysed, as it was suggested in the review done in Section 2.3.3. Table 3.1 shows a summary of the types of URLs found.

URL Type	URL	# URL retrieved
1	potential-booter.com	71
2	potential-booter.com/login.php	1167
3	www.domain.com/potential-booter	
4	potential-booter.domain.com	

Table 3.1: Examples of retrieved URLs

In Table 3.1 four different types of URLs can be observed: with only a host name (type 1), with host name and path (type 2), where Booter is a page of a website (type 3), and where a Booter is a subdomain of a domain name (type 4). The goal is to focus on the URL type that has the highest probability to be a Booter. Note that often URLs type 2 are subpages of URLs type 1. Thus, it is decided to analyse only the type 1. Although URLs type 3 and 4 can potentially contain Booters, it was noticed that it is more likely that they provided information about Booters, not the websites themselves.

In the analysed set, only 71 were of Type 1. All the 71 URLs were accessed manually to decide whether the accessed website is a Booter. Out of the 71 URLs Type 1, 42 were Booters, what is almost 60% of the URLs Type 1. This means that Booters are indeed easy searchable using Google Search and are majority of URLs Type 1. The remaining 29 websites were classified as non-Booters. The analysis will further focus only on the mentioned 71 websites: 42 Booters and 29 non-Booters.

3.2.2 IP address analysis

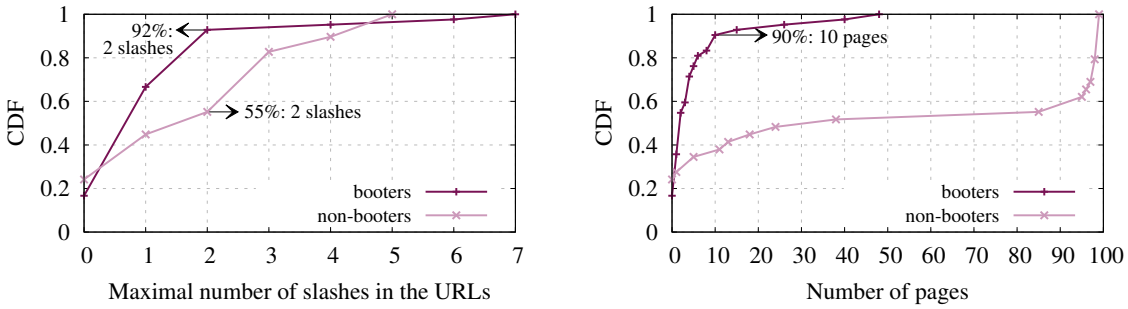
It was already mentioned in the literature that Booters use the services of DDoS Protection Service (DPS) companies [11]. The IP analysis was done in two aspects: (i) the geographical location of the IP addresses was verified and (ii) it was checked whether the IP address belonged to a blacklist already. The analysed location was verified using the database of *IP2Location* provider in an online tool [49] and it did not turn out to be a characteristic of Booters: most of all of the URLs of both Booters and non-Booters were hosted in the United States. When analysing the geographical location, something else seemed more interesting: the hosting provider. Since it is often so, that the name servers remain unchanged and are dependent on the hosting provider, it will be later analysed what name servers are the URLs pointing to.

Regarding the presence on a blacklist, it was verified if the name of the potential Booter's URL is on a domain name blacklist (such as *'.dbl.spamhaus.org'*, *'.multi.surbl.org'*, *'.multi.uribl.com'*), or if the IP address of the host is on one of the following blacklists: (*'.srnblack.surgate.net'*, *'.psbl.surriel.com'*, *'.ubl.unsubscore.com'*, *'.dnsbl.sorbs.net'*, *'.cbl.abuseat.org'*, *'.bl.spamcop.net'*, *'.zen.spamhaus.org'*, *'.sbl.spamhaus.org'*, *'.xbl.spamhaus.org'*, *'.pbl.spamhaus.org'*). As a result it was observed that more of the non-Booter websites (5/29) belong to a blacklist than Booters (1/42). This could be due to the following reasons: Booters actually pretend to be a legitimate business, whereas when searching for Booters, a lot of websites recommending and describing Booters can be related to some hacker forums which have a bad IP reputation.

3.2.3 Website structure analysis

The website structure was analysed in several ways. First, all the 71 URLs were analysed with the maximum number of slashes ('/') and the overall number of the indexed pages (by Google). In total,

1710 indexed pages were found for this dataset. This number would be bigger, but it was predefined not to crawl for more than a 100 pages. From this amount, 239 pages were found for 42 Booters and 1471 were found for the 29 non-Booters. Figure 3.3 summarizes the findings.



(a) Depth of URL. (b) Indexed pages.

Figure 3.3: CDF of website structure aspects.

In Figure 3.3, graphs of the Cumulative Distribution Function (CDF) of the website structure are shown. While Figure 3.3(a) shows the CDF of depth level of these 71 website, Figure 3.3(b) shows the CDF of the known pages. Both graphs show the results for Booters and non-Booters (classified manually). It can be observed in Figure 3.3(a) that 92% of Booters have their website depth up to 2 levels, whereas only 55% of non-Booters have the same depth. Considering Figure 3.3(b), it can be noticed that 90% of Booters have 10 or less pages. In addition, Booter websites never exceed 50 known pages, what is an interesting observation that can be used to eliminate non-Booters from a set of URLs (that in this analysis is almost 50%).

By in depth analysis of the indexed subpages of 71 potential Booters, it can be noticed that some page names appear more than once, such as “register”, “ToS”, “plans”, “buy”, and “hub”. The number of times those pages were observed is shown in Figure 3.4. Pages such as Terms of Service (ToS) and “registration” are more often appearing when analysing Booters. Almost 60% of Booters have a registration page (24/42) and around 40% of Booters have ToS page (17/42).

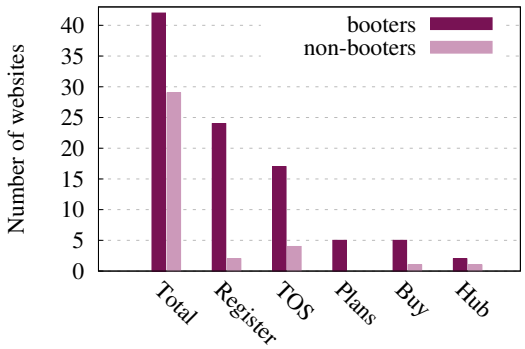
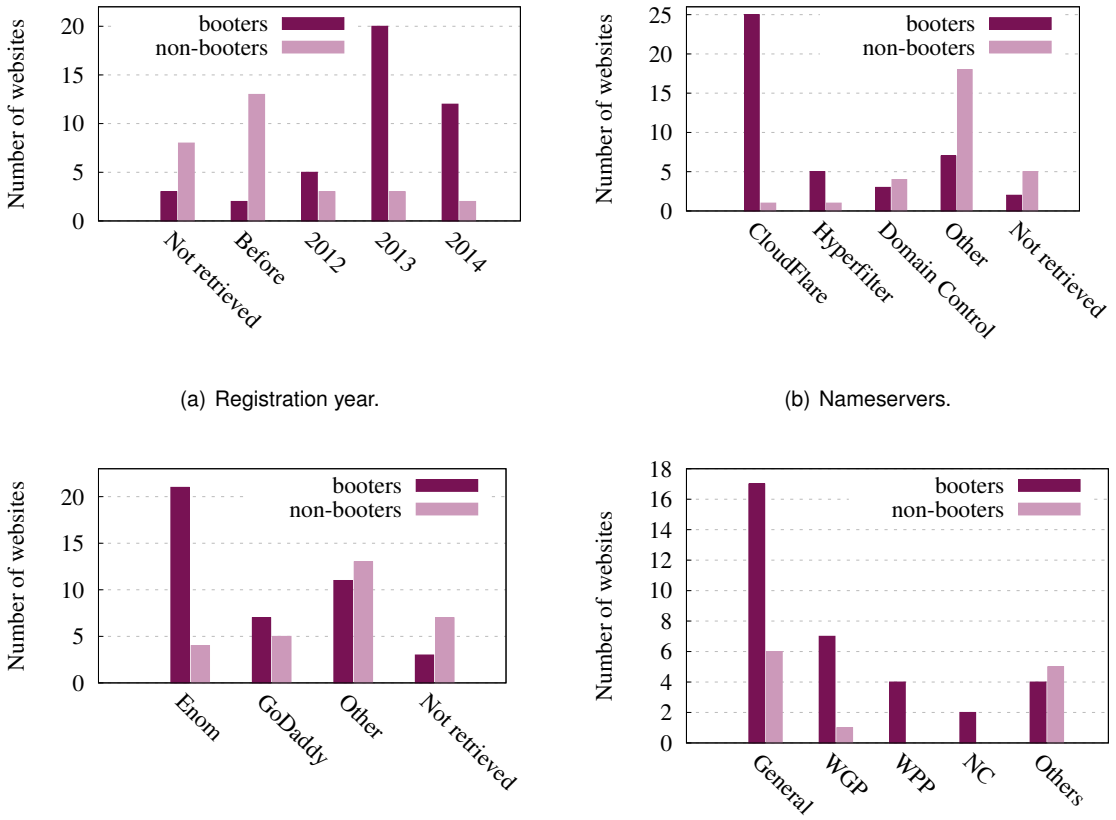


Figure 3.4: Page names analysis.

The other page names on the Figure 3.4, plans, buy, and hub, are not that promising. For example only 5 out of 42 Booters had pages referring to plans or redirecting to purchase page (buy). Although the page “hub” is crucial to Booters, because through that page a user can start an attack, it was found only twice. The reason for that could be that this page is accessible only after logging into the Booter and it is not indexed by Google.

3.2.4 WHOIS analysis

Next step is to analyse the WHOIS information of the 71 potential Booters retrieved by the crawler. The findings are presented in Figure 3.5.



(a) Registration year.

(b) Nameservers.

(c) Registrar.

(d) Contact of the domain name owner.

Figure 3.5: WHOIS information.

In Figure 3.5 four different types of information related to WHOIS are presented. From the first one, in Figure 3.5(a), it can be observed that more than 88% of the Booter domain names (37/42) were registered in 2012 (5), 2013 (20), and 2014 (12), whereas a bit less than half of the non-Booter websites were registered before 2012 (13/29). Overall, it shows that most of the Booters returned by the crawler are relatively new. By analysing the namesevers of the retrieved Booter domain names, showed in Figure 3.5(b), it is not surprising to observe that almost 60% of Booters (25/42) are associated to CloudFlare. It was already pointed out in [11] that Booter subscribe services from CloudFlare to be protected against DDoS attacks from the competitors (other Booters). However, a new company offering DDoS protection is observed, Hyperfilter. It is the first time that this company is observed in this phenomenon. Also in Figure 3.5(b), it can be confirmed that majority of non-Booters have more variety in nameservers (18/29).

By analysing the companies that provide the domain registration (registrar), showed in Figure 3.5(c), it can be noticed that half of Booters (21/42) have their domain registered with Enom, which is a known domain registrar. This is probably because of cheap price, but also because Enom offers the “Whois-Guard” service used for hiding the contact details. This information is more evident when the contact of the responsible for the domain names is analysed. In Figure 3.5(d) it can be observed that more than 40% of Booters (17/42) have their contact hide from the WHOIS information. Usually the contact is hidden by many services, among them: “WhoisGuard Protected” (WGP - 7/42), “Whois Privacy Protect” (WPP - 4/42), and “NameCheap” (NC - 2/42).

3.2.5 Website content analisys

The website content can be analysed following several aspects: the textual content, the meta data, and the visual content. Below the mentioned aspects are analysed or commented on.

Website - textual content: Booters provide either scant login page with no other text than “username”, “password” and “login”, or they provide more detailed information about themselves. In the first case, the textual analysis seems not appealing: there is simply not much text to analyse. In the analysed set more than half of Booters (22/42) provided such simple screen. Moreover, the textual content is usually analysed in order to determine the **subject** of the website [33, 34, 35]. This means, that choosing for text analysis, it would be possible to classify a website as a website about Booters, but not as a Booter itself. These two discouraging findings were not in favour of investigating the textual content, because of what this analysis was skipped.

Website - meta content: Looking into the meta data within the Booter websites, almost 62% (26/42) has neither description nor keywords, showed in Figure 3.6. This result is very similar and consistent with the simplicity of Booters visual interface (described in the previous section).

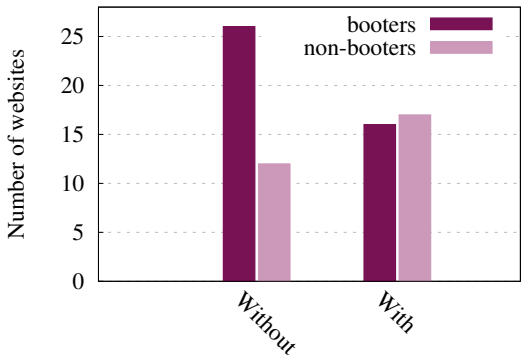


Figure 3.6: Meta data.

It was also investigated what is the content of the meta fields of the websites which had either meta description or meta keywords (the remaining 16/42 Booters and 11/29 benign websites). The meta data was searched with regular expressions for words related to:

- Stress testing - e.g. *stresser*, *stress testing*, *server stresser*
- DDoS - e.g. *ddos*, *attacks*, *rudy*
- Gaming - e.g. *xbox*, *ps3*, *minecraft*

The keywords related to categories listed above did not distinguish Booters, e.g. only 13 Booters showed words related to stress testing, but also 5 of the non Booters had this kind of words. The same followed for DDoS category. Gaming-related words were more popular among benign websites. This shows again: that a lot of websites of similar subject have this type of meta data, like it was suspected it would be in case of textual content.

Concluding, it cannot be defined whether Booters or non-Booters tend to use meta data, and it cannot be said that Booters tend to use specific key words within the meta data. Therefore, although the meta data can help on the understanding the purpose of a website, because of insufficient information, this feature cannot be used to define if a website is a Booter or not.

Note that in theory meta data is fundamental to have a better positioning in search engines, such as Google Search. Therefore it is interesting to correlate the popularity of a Booter with the existence or not of meta data. To do so, the Alexa’s Rank [5] is used to get information about the 42 Booter webpages. By doing so, in general it could be observed that Booters with meta data have a higher ranking, for example in Table 3.2 two Booters are shown. Booter E that has meta data was ranked around the 273k

[5] http://www.alexa.com

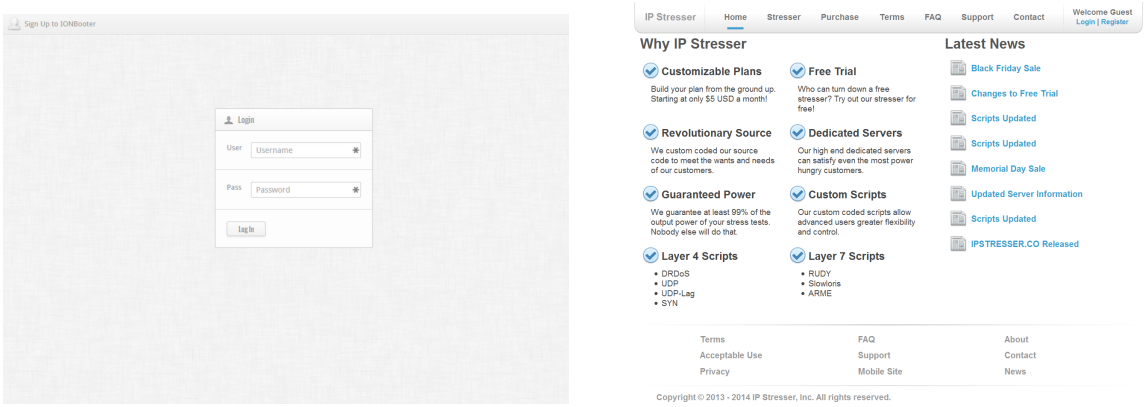
position, while Booter F had a thousand times lower position. Note that the highest Alexa's Rank value is 1 (that usually has Google or Facebook).

Name	Meta data		Alexa's Rank
	Description	Keywords	
Booter E	Powerful and Affordable Stress Testing	stresser, denial of service, dos, ddos, drdos, syn, ssyn, udp, sudp, udp-lag, rudy, slowloris, arme	272.979
Booter F	-None-	-None-	2.580.782

Table 3.2: Examples of Booter meta data and their popularity

Of course, meta data is not the only determining factor to have a higher ranking in the Internet. However, it is a positive surprise to observe a clear relation between both factors.

Website - visual content: As it was mentioned in Section 2.1, Booters tend to look alike. Among the 42 Booters described in this Section, the same notice was taken: that they have only two completely different types of first page: or (i) a very simple first page containing a simple login interface, for example showed in Figure 3.7(a), or (ii) a verbose page full of textual content and appealing advertisements, for example shown in Figure 3.7(b). For both types Booters provide a very simple and user-friendly interface. The most remarkable finding related to the visual interface was that while more than a half of Booters (22/42) has a "login" button in their main page only 1 non-Booter has such button.



(a) Booter F - login page.

(b) Booter E - login page.

Figure 3.7: Examples of login pages.

The dataset explained in this section provided valuable information about Booter websites: what are the characteristics of these websites that could be used in order to automatically classify a website as a Booter. This manually classified set was used as a training dataset, and will be further validated on a larger dataset. The next section will summarize the most significant features describes in this section.

3.3 Most significant features of Booters

In Section 3.2.1 was concluded that URL type 1, i.e. those that are composed by only the URL hostname, are more suitable to be a Booter. In Section 3.2.3 it is noticed that all Booters have less than 50 subpages and in general 2 levels of depth of the URL. It can be also observed that websites, which have pages "register" and "tos" are more often Booters. When it comes to WHOIS information, Section 3.2.4, highlights that Booters tend to use services from DDoS protection companies, such as CloudFlare or HyperFilter. The domain names used by Booters are most likely registered in 2012 or later, and the used registrar is most likely Enom. Moreover, information about the owners of Booters is hidden using services such as "WhoisGuard". Finally, in Section 3.2.5, it was shown that Booters often have a simple

interface with a login button. Through those observations it is decided that the main features of Booter websites are:

1. Number of pages - less than 50.
2. Depth level of the website - maximum 2.
3. Presence of registration page.
4. Presence of terms of service page.
5. Domain creation time - 2012 and later.
6. Obfuscated WHOIS data.
7. Protected by a DPS.
8. Specific registrar - Enom.
9. Login button on page.

Using the mentioned features, a comparison of Booters and non-Booters will be done in the following sections, starting with similarity measure in the next chapter. The next section explains the dataset that these features were validated on.

3.4 Extending the dataset

The dataset used for validation is described in this section and was collected through the time: 15 November 2014 and 7 January 2015. The goal of these two sets is to have two disjoint sets, where the validation will not be biased by the "training" set. The data set used for validation consists of 31942 distinct URLs. This dataset will be used in Chapter 4 for validation. The amount of found URLs per search word was more or less the same through the whole time, as it can be observed on Fig. 3.8.

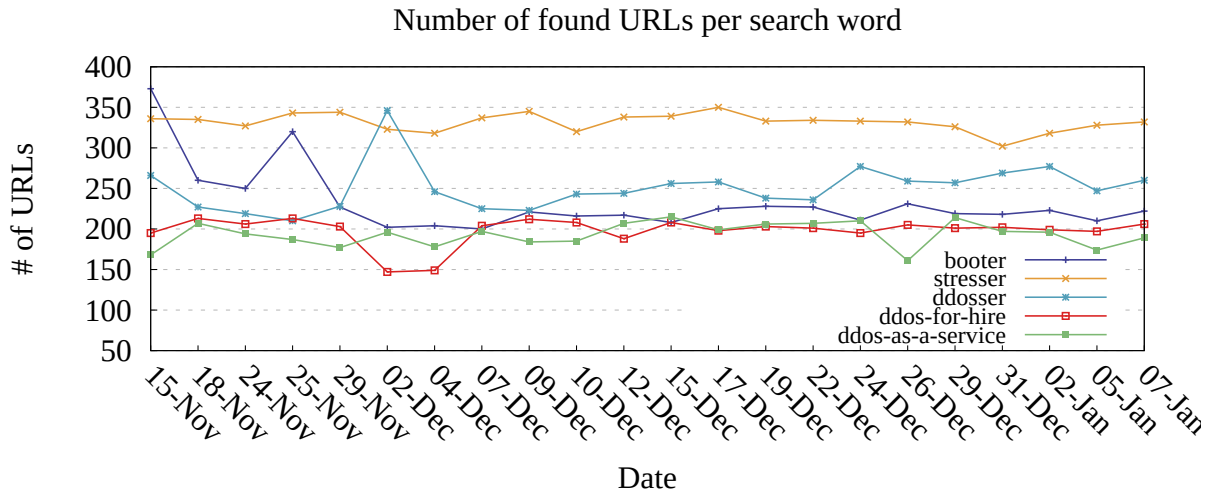


Figure 3.8: The number of found URLs using Google Search function between 15 November 2014 and 7 January 2015

Figure 3.8 shows, that some of the search words are more effective than the others: "stresser" was returning most results, only twice being exceeded by "booter" and "ddosser". Second word retrieving most results was at the beginning "booter", later overcame by "ddosser". Interestingly, around that time the search function was optimized by parameter which was skipping the URLs in footer of Google

Search results. This suggests that for word “ddosser” more distinct websites were retrieved whereas for the word “booter” the results were returning more subpages. The least effective through all time were search phrases “ddos-for-hire” and “ddos-as-a-service”. The reason for that could be the fact that these are specific phrases. Among the mentioned URLs, 220 were Type 1 URLs. The 220 URLs are disjoint with the 71 URLs belonging to the set explained in previous section. The validation uses only the URLs Type 1.

All of the 291 (220+71) URLs Type 1 from both: training and validation set were manually accessed to define whether the website is a Booter, for the validation purposes. As mentioned before, the training set consisted of 71 URLs, out of which 42 were Booters, and 29 were non-Booters. Overall, the initial validation set consisted of 220 URLs, and this set contained 107 Booters, 94 non-Booters and 19 pages which by the time they were accessed they were offline or contained no data. Since it is impossible to judge whether the website is a Booter if the content was not returned, those websites were not used for validation, and the set used for validation was narrowed down to **201 URLs**.

The set explained above was an initial set and was still changed due to the fact that it was not possible to retrieve some information from all of the Booters. The reason for this was that in the meantime, between accessing the websites manually to classify them and retrieving additional information about them, the websites were not “online” anymore (e.g. the domain name was available for sale again). For this reason the next section will explain how the list is being updated.

3.5 Keeping list up to date

Since as it was pointed out before, blacklists need to be up to date, this section introduces what is done in order to keep the list of Booters up to date. Since there is a necessity of keeping track of many URLs at the same time, it was decided to do it automatically. It is important to remember that maintaining the list is not only to add the new URLs but also to remove the old ones. Since crawling for new URLs was already explained in Section 3.1, this section explains updating the status of a URL.

An active Booter, is defined as a Booter (i) whose IP address is resolvable, (ii) whose content is possible to be downloaded, and (iii) whose domain is not for sale. For the first point, it is only necessary to resolve the IP address of the given URL. The second point verifies whether a DPS company generated a CAPTCHA challenge (see Section 3.1), since otherwise the content of the website was not downloaded. The first point matches the content of the website with regular expressions, most likely to retrieve when being redirected to a page offering the domain name for sale, e.g. *“This domain name is for sale”* or *“Buy this domain”*.

Before analysing the features it was taken onto account what is the status of the analysed, potential Booter URL. The validation set of 201 Booters from previous section was narrowed down to 163 URLs (38 URLs were inactive or for sale).

3.6 Concluding remarks

Concluding this Chapter, it was possible to retrieve URLs related to Booters using a set of search words defined in Section 3.1, with the majority of analysed URLs being Booters (60% in the training set and 53% of the classified URLs in the validation set). Moreover, this section explained how was it possible to gather all the features which could potentially be helpful to decide whether a website is a Booter. Based on URLs gathered by the crawler, the retrieved websites were characterized in Section 3.2. Based on this characterization it can be said that Booters definitely tend to follow some trends, e.g. they are relatively small (less than 50 indexed pages), as it was shown in 3.2.3. This, and other observations can be potentially used in order to classify Booters. The next Chapter will validate this approach based on a data set explained in Section 3.4.

CHAPTER 4

FEATURE ANALYSIS

As the result of the previous chapter, a crawler searching for Booters was developed. This crawler has two main tasks: to crawl Google Search engine in order to look for Booters, and to find additional information about them. The goal of this Chapter is to decide whether based on such characteristics it is possible to decide if a website is a Booter. First, Section 4.1 addresses the question whether Booter websites are similar, based on these features. In Section 4.2, Booters are ordered with the number of satisfied features, and in Section 4.3 the presence of features is additionally assigned a weight. These two approaches are compared in Section 4.4. Based on this comparison, an approach for classifying Booters is proposed.

4.1 Similarity measure

In this section, the question whether Booters resemble each other is asked. If Booters are more similar to each other than to other websites, it could be possible to compare any website to some reference and decide if that website is a Booter. To address the question of similarity of Booters, the similarity has to be defined first. Similarity is the opposite of distance, and there are several distance metrics defined, such as Euclidean distance, cosine distance or Hamming distance. The higher distance between two websites, the smaller the similarity between them. In this section, two distance metrics are considered: (i) Euclidean distance, and (ii) cosine distance. Below they will be first defined and then visualised based on the existing dataset.

Euclidean distance: Is a distance between two points in Euclidean space. For instance, having two vectors in Cartesian coordinates A and B, where $A = [a_1, a_2, ..., a_n]$ and $B = [b_1, b_2, ..., b_n]$ the distance $d(A, B)$ is then defined as in equation 4.1.

$$d(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{4.1}$$

Euclidean distance is used usually for explaining the distance geometry: in wireless networks, statics, data visualisation [50]. The advantage of Euclidean distance is that is can be used in n-space as long as a point is represented as a Euclidean vector. In case of the analysed websites, they are represented as 9-dimensional vectors, where each dimension represents a feature: $F = \{f_1, f_2, ..., f_9\}$. The value of each feature is binary: either a feature is satisfied or not.

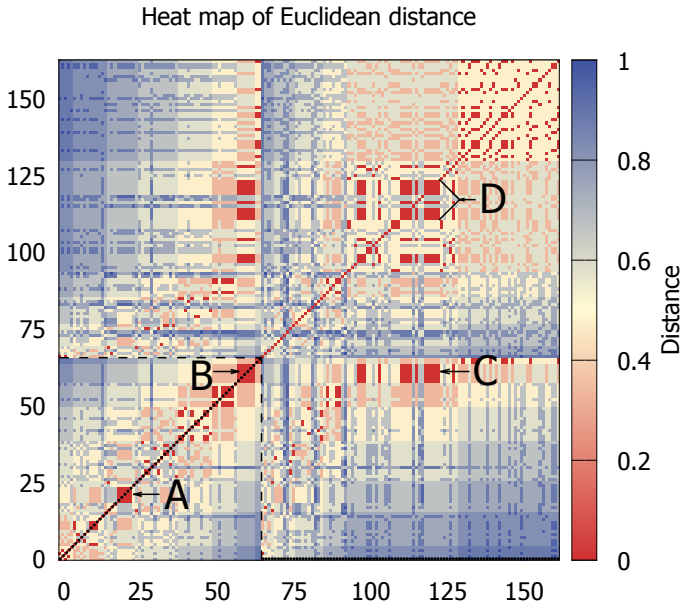
Cosine distance: Cosine distance is the similarity translated to positive space with equation 4.2. Similarity is defined as cosine angle between two vectors A and B as shown in formula 4.3. This distance metric is not a proper distance metric as it does not satisfy the triangle inequality.

$$D_C = 1 - S_C \tag{4.2}$$

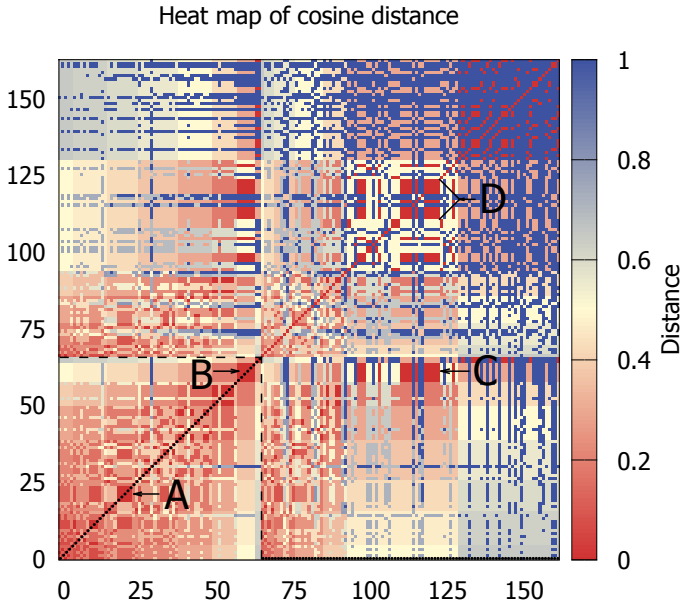
$$S_C = cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{4.3}$$

This metric can also be used for any number of dimensions and is popularly used in text mining or Information Retrieval. Since some of the websites have none of the features satisfied, this means that in equation 4.3 the denominator can be 0, so we replace these cases with distance equal to 1. The websites are represented as vectors like in the other distance metric: representing each feature as a dimension.

The dataset explained in the Section 3.4 consisting of 163 URLs, including 66 Booters and 97 non-Booters was analysed the way explained above. For each website represented as a binary vector of features listed in Section 3.3, this website is compared with each of the 163 websites (including itself). The larger the distance, the less similar the websites are. The results are visualised and presented as heatmaps in Figure 4.1. The analysed websites were sorted in following way: first Booters with highest satisfied features are presented, then decreasing with the amount of features satisfied. After that the websites are listed also in decreasing order of features satisfied.



(a) Euclidean distance heatmap.



(b) Cosine distance heatmap.

Figure 4.1: Heatmaps of the websites dissimilarity.

In the heatmaps in Figure 4.1, the red colour represents smaller distance, that means higher similarity, while the blue colour represents the larger distance and smaller similarity. For instance, if website 5 resembles website 10, there will be a red dot at the intersection of 5 and 10. However, this analysis will not focus on single points, but on attempt to observe a “cloud” of them.

It can be observed that the diagonal of the both representations is always equal to distance 0, meaning that the two websites look alike. This is expected since the diagonal represents a website compared to itself. Additionally, the 66 websites which are manually classified as Booters are marked in the graphs by black points across the diagonal. The websites classified as non-Booters are marked as black points for value of “0”. Please notice also that the graphs are in each case symmetric along the diagonal, due to simple fact that if website 5 resembles website 10, then 10 is also similar to 5.

It is expected that one “cloud” of websites will appear on both graphs for Booters, meaning for the first 66 websites, marked on the graph with the black dashed line. Since the rest of the websites could be random they may but do not have to resemble each other (websites 67–163). The ideal case of what is expected to obtain is shown in Figure 4.2. If all the Booters satisfied all the features and all the non-Booters satisfied no features, this would be the result.

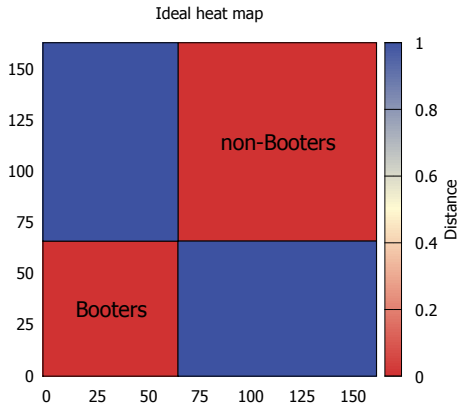
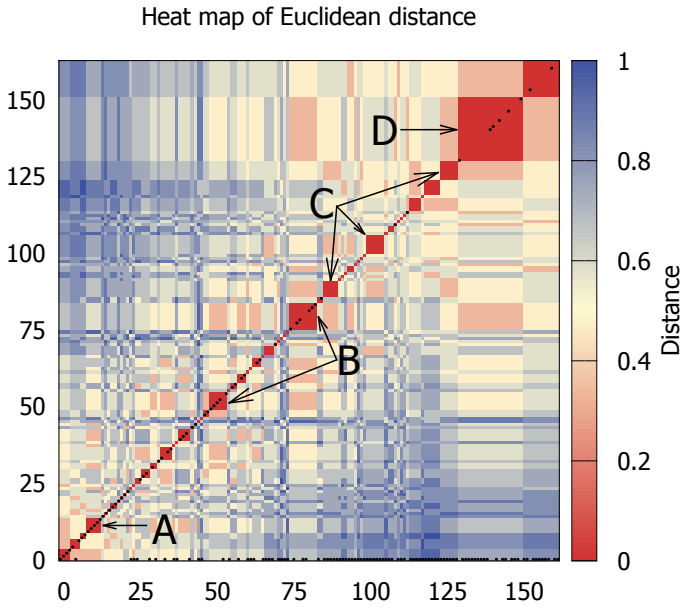


Figure 4.2: Ideal case of a heatmap indicating Booters.

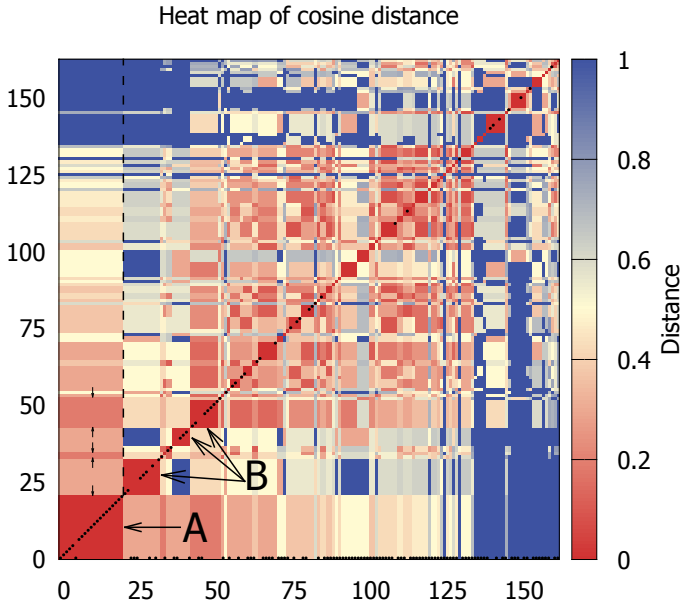
However, as compared to Figure 4.1, this is not the case: a “cloud” of Booters is not observed. For the Euclidean distance, the first 66 websites produce two small clouds: a set of 5 websites around 20th website (marked in Figure 4.1(a) as A) and a set of 6 websites around 60th website (marked in the same figure as B). Moreover, there are certain non-Booter websites on the graph which resemble the Booters with little amount features satisfied (denoted on the same figure as C). This due to the fact that the Booters shown in this graph for values 50 and more satisfy most likely 2 or less features. In case the features chosen are not typical features of Booters, but also of other websites, there is a higher probability that non-Booter websites will also satisfy that one or two features. If this is the case, this means that among the features chosen in this thesis, some of them are very generic and are not typical feature of a Booter. Comparing position of B and C, it can be observed that some Booters and non-Booters look alike in our metric. This is a very not satisfying observation, as it was hoped that the Booters and non-Booters should not resemble each other at all. Moreover, the non-Booters also resemble each other what can be observed in the area marked as D in the Figure 4.1(a).

For the cosine distance, the situation is similar, as it can be observed in Figure 4.1(b): the “clouds” of similar websites are noticed in the same places. However, the websites in cosine metric are much more similar, what can be concluded from the overall, more red colour of the heatmap. This makes more difficult to notice the most similar points, denoted with letters A, B, C and D in the Figure 4.1(b).

When presenting heatmaps, one important factor is the order of the analysed websites. Because of this, in second approach the sorting was done with side-to-side sorting algorithm. The goal of this sorting is to shape a cloud of points, or the fact that there are one or more separate clouds. This sorting is used in Figure 4.3. Again, the websites which are manually classified as Booters are marked in the graphs by black points across the diagonal and the non-Booters are black points for the value of “0”.



(a) Euclidean distance heatmap - sorted.



(b) Cosine distance heatmap - sorted.

Figure 4.3: Heatmaps of the websites dissimilarity.

First of all, it was expected to find a block of similar websites, consisting only of Booters (in ideal case). In neither metric can we find such example. In Figure 4.3(a), showing the Euclidean distance, several clouds of similar websites are presented: A, B, C, and D, among only A is consisting of Booters only. B and D represents few clouds built of Booters and non-Booters. C represents few clouds built of non-Booters only. Interestingly, cloud D in Figure 4.3(a) contains exactly 6 Booters, as it was also in cloud B in Figure 4.3(a). This suggests that cloud D in the Figure 4.3(a) is built of the earlier mentioned websites that satisfy up to 2 features, probably more generic, and thus many non-Booters are similar to them.

In the Figure 4.3(b) showing the cosine distance, the first cloud of websites is quite promising: out of 21 websites only one website is a non-Booter (marked in the figure as A). The next few clouds consist also of majority of Booters (marked as B). Looking at this figure, if it was decided that Booters are all websites similar to the cloud A, the following approach would follow: the websites are compared to the first block (left side of the dashed line). If distance is e.g. 0,2 or lower, consider the website a Booter. In the figure the blocks that satisfy this condition are marked between the small arrows. This threshold finds exactly 28 Booters and gives 4 (2%) false positives, what gives 74% accuracy. Still, many Booter websites would remain undetected.

As a conclusion, it is decided that cosine distance is much better metric for comparing websites based on features represented as binary vectors, however based on a set of predefined features classifying a website as a Booter is not satisfactory. The possible reason for that is the discrete values of the features. The features do not give any flexibility, that is, the feature is either satisfied or not. On the other hand, possible reason for this is the pre-defined, chosen set of features meaningful for Booters. For this a different approach will be further investigated in this thesis. Moreover, it was shown that several websites are similar to each other: both Booters and non-Booters, because of satisfying, probably, more generic features, that apparently are not entirely a characteristic distinguishing a Booter.

4.2 Classifying Booters using unweighted score approach

To present the potential Booters differently, we define a website ω and a feature f_n for n∈[1,9] for each of the nine features of the website mentioned in section 3.3. Each website is then given a value from 0 to 9, depending on the amount of features satisfied. The feature score of the website (FS_{w1}) could be described with the formula 4.4:

$$FS_{w1} = \sum_{n=1}^{n=9} \begin{array}{ll} 0 & \text{if } \omega \text{ does not satisfy } f_n \\ 1 & \text{if } \omega \text{ does satisfy } f_n \end{array} \tag{4.4}$$

Based on formula 4.4, normalised to the maximum of the features to specify (so divided by 9), the graph shown in Fig. 4.4 was obtained, for the dataset described in previous chapter. In total 163 websites were analysed including 66 Booters and 97 non-Booters.

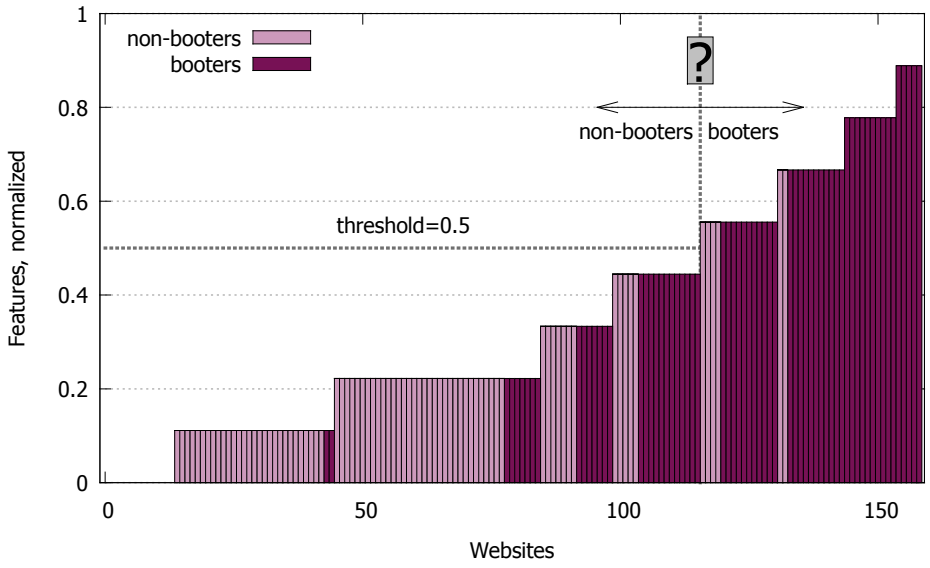


Figure 4.4: Classifying website as a Booter.

To define the website as a Booter, a certain threshold needs to be provided, above which all websites will be classified as Booters. For example, in Fig. 4.4, for threshold = 0,5 it is possible to classify Booters

with 79% accuracy and 4% False Positives. In Figure 4.4 it can be noticed that some Booters satisfy just one feature, and because of that, they score only 0,1. However, this assignment does not take onto account if the satisfied feature is more significant or not. Because of this it will be investigated if this problem can be addressed by assigning weights to the features.

4.3 Classifying Booters using score with weights

In one of the frameworks for detecting phishing websites, the more accurate features are determined by calculating the odds ratio of the features [51]. Odds ratio is defined as the ratio of the odds of an event occurring in the positive class (Booter) to the odds of it occurring in the negative class (non-Booter). An odds ratio of 1 indicates that a feature is equally useful in identification of both classes. An odds ratio greater than 1 implies the corresponding feature is more useful in identifying the positive class and ratio less than 1 implies that the feature is better for identifying the negative class. Suppose that in a sample of 100 websites, we find 50 Booters and 50 non-Booters. Among the Booters, 45 are protected by a DPS. Among the non-Booters only 10 are protected by a DPS. The odds of a Booter being protected by a DPS are 45 to 5, or 9:1, while the odds of a non-Booter being protected by a DPS are only 10 to 40, so 0.25:1. The odds ratio is thus 9/0.25, or 36, showing that Booters are much more likely to be protected by a DPS than non-Booters.

The higher the odds ratio, the more relevant the characteristic is to Booters and the less is it an indicator of the non-Booters. For this reason this information is used to assign a weight to each feature. The odds ratio was calculated for each of the features for the training dataset explained in Section 3.2: 42 Booters and 29 non-Booters. The weight is assigned by the odds ratio divided by the sum of the odds ratio of all the features. The result is presented in Table 4.1.

#	Feature	Odds Ratio	Weight (h)
1	Maximum number of '/' <=2	14,85	0,11
2	Number of indexed pages <=50	43,04	0,3
3	Register page	23,05	0,16
4	ToS page	2,7	0,02
5	Domain creation year >=2012	9,35	0,07
6	Obfuscated WHOIS information	10,62	0,08
7	Nameservers belonging to DPS	17,41	0,12
8	Specific registrar:Enom	7,35	0,05
9	Login button on page	12,97	0,09
	Total	141,34	1

Table 4.1: Odds Ratio of the features and assigned weight.

Based on the same set of features, the weights are defined as $h_1, ..., h_9$ respectively for features $f_1, ..., f_9$, as they were calculated and presented in table 4.1. The new feature score for each website (FS_{w2}) is defined in equation 4.5:

$$FS_{w2} = \sum_{n=1}^{n=9} \begin{array}{ll} 0 & \text{if } \omega \text{ does not satisfy } f_n \\ h_n * 1 & \text{if } \omega \text{ does satisfy } f_n \end{array} \tag{4.5}$$

The classification according to this approach is shown in the Fig. 4.5.

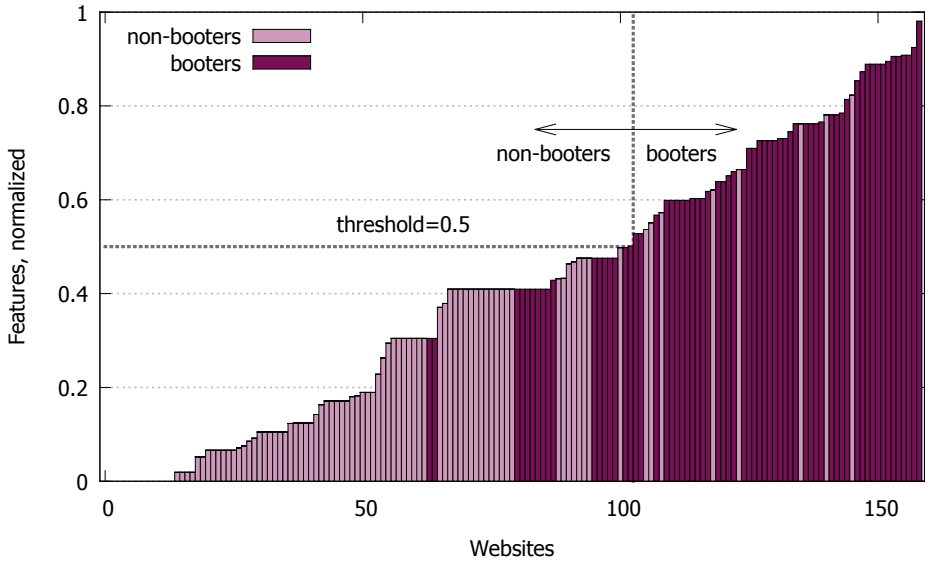


Figure 4.5: Classifying website as a Booter - weighted approach.

In the Figure 4.5, just for the reference, the threshold was marked on the level of 0,5. For this threshold the accuracy of the method is 85% with 5% false positives. What can be noticed is that using this approach many more Booters are assigned value more than 0,1, as it was in previous approach. This means some websites can be easily eliminated as non-Booters without many false negatives. Moreover, in the previous approach, in Figure 4.4, for the same threshold 0,5 the method was less accurate (only 79%). This means that assigning weights to the features is a good improvement of classifying Booters.

The drawback of this weighted approach is that the weights described in the Table 4.1 are calculated for a defined set, that was used in retrieving the characteristics of Booters. For the current work, the two approaches will be compared in the next section.

4.4 Comparison and proposed classification

Since the goal of this analysis is to provide the better metric for calculating the better accuracy with a low false positive rate, these values are compared in the Figure 4.6. The accuracy on highest level was achieved for the weighted approach for threshold=0,5 for the same false positive rate.

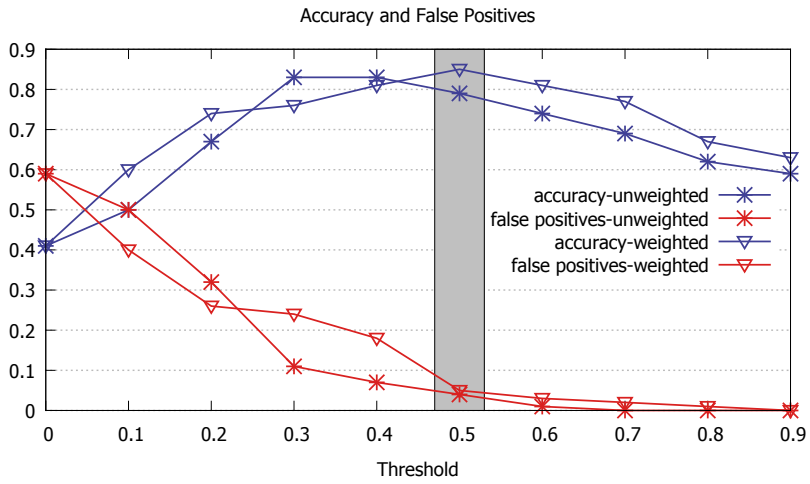


Figure 4.6: Comparison of accuracies and false positives error rate.

In the Figure 4.6 we can see that the accuracy of a weighted approach is for most of the time on higher level than the unweighted approach. For the trigger 0,5 and higher the false positive rate of both approaches are on similar level (5% and less).

Based on the results in this methodology, it would be recommended to use the weighted approach with threshold 0,5 for the best results of obtaining Booters. For this reason the crawler is rebuilt with classification function: it retrieves the features of a URL and assigns a weighted score to it. Based on the defined threshold the classification function can decide whether the investigated website is a Booter or not.

4.5 Decision tree

Because of all the lessons learned through the process of classifying Booters, one final approach is proposed: classifying the websites is done by means of decision tree. Some features were clearly implying that website is not a Booter, e.g. the number of indexed pages never exceeds 50, as it was shown in Section 3.2.3. For the decision tree additionally one more metric is introduced. During the research many websites related to Booters - not being Booters themselves - were found. A lot of users of Booters would like to know, which Booter is the most reliable, which one is a fake, and which one is overall “the best”. Since websites with such list of URLs exist (e.g. `http://booters-review.blogspot.nl/`, `http://top10booters.com/`), it was decided to introduce one more metric: website reputation. The website reputation is defined as “1” if the URL is present on the website about Booters or “0” if it not there. The list of sources of websites used for reputation is available in the Appendix D.

The decision tree is presented in Figure 4.7 and uses the following findings: (i) if the website reputation is 1 then consider URL as a Booter, (ii) if the creation date of domain name is earlier than 2011, consider website as a non-Booter (feature 5), and (iii) if there are more than 50 pages of the URL indexed by Google, consider the website as a non-Booter (feature 2). The first finding is supported by the fact that it is very likely that overlapping set of: URLs coming from the Google Search about Booters, and URLs coming from the mentioned Booters review websites, will be a set of Booters. The second finding bases on feature 5 - the creation year of the domain name has to be 2012 or later - but it is altered to 2011 or later. This is due to the fact that two Booters are still existing and their name was registered in 2011. Most importantly, there were no classified Booters with the registration date earlier than 2011. The third finding is supported with the mentioned example – the number of indexed pages never exceeds 50, as it was shown in Section 3.2.3. Moreover, it was shown in Section 4.3, this feature got quite a high odds ratio rate (43,04), meaning it was indicating Booters much more significantly than non-Booters. For the remaining URLs the following features are checked: the number of slashes (feature 1), register page (feature 3), “terms-of-service” page in the indexed paths (feature 4), obfuscated

ownership of the domain (feature 6), DPS protection (feature 7), “Enom” as a registrar (feature 8), and presence of login button on the page (feature 9). If URL satisfies at lest one of these features, then the website is classified as a Booter.

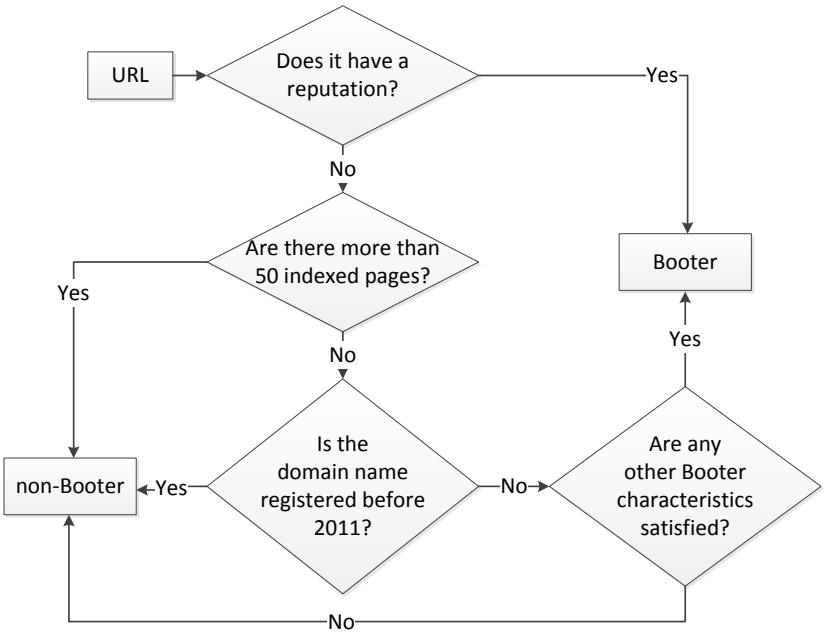


Figure 4.7: Decision tree - Booters.

Using these simple observations provided 88% accuracy with 4% false positives. The result coming from this simple approach to classify Booters shows that there is still a lot of space for improvement.

4.6 Concluding remarks

This chapter introduced most significant characteristics of Booters, proposed several ways to present and classify a website as a Booter and provided a comparison of those methods. It was decided to use the weighted approach of classifying a Booter, since this way it is possible to reduce the number of false negatives. This method provides the most accurate list of Booters, however, the false positive rate is still not ideal: around 5%. To improve this score some recommendations for future works will be suggested in next chapter.

CHAPTER 5

CONCLUSIONS

The problem of [computer] viruses is temporary and will be solved in two years.
John McAfee, 1988

Underestimating the possibilities of a harming phenomenon could be compared to giving up on taking any precautions in the first place. Just like in the quotation above, the founder of McAfee Antivirus underestimated the potential of computer viruses back in 1988. For that reason it is important to address the challenge of mitigating DDoS attacks in any form, also those caused by Booters. Many may argue that it is still a problem of gamers only, but probably not for long, as it also is underlined by some DDoS Protection Service companies [4]. Within the scope of this thesis, the proposed approach to mitigate Booters is by means of a blacklist. Therefore, in this thesis we propose a methodology to create such (black)list based on a Booter website classification.

The first part of this thesis addressed the state-of-the-art of Booters and their mitigation (Chapter 2). That part described the look of Booter websites, the overview of attacks they offer, and the (lack of) difference between a Booter and a stresser. Moreover, three mitigation areas of Booters were explained: the target level, the infrastructure level and the access level. The main finding was that one of the popular solutions in mitigating other malicious websites on access level is not yet used to mitigate Booters, namely by means of blacklists. This solution is used in mitigating phishing websites or websites spreading malware, and requires an accurate and updated list of malicious websites. Based on this finding it was clear that mitigating Booters in the access level is promising, provided that it is known which websites to filter. However, this way another gap was pointed out: there is no freely available list of Booters for the research purposes. For this reason the Chapter 2 is wrapped up with an overview of features that can be used in order to classify websites, to classify a website as a Booter, and provide a list of them.

To develop such list, in Chapter 3, it is investigated which search terms are the most suitable to retrieve Booter websites. Five search terms were defined, covering a wide list of Booters. Through these terms almost 32 thousands of URLs were retrieved, which provided 149 Booters manually classified and provided in Appendix B for research purposes.

In the same chapter, in the Section 3.2, the initial list of potential Booters was used to investigate the characteristics of them. This list consisted of 71 URLs, including 42 Booters and 29 non-Booters, found with the same search terms. The analysed characteristics were the URL structure, the IP address of the website, the website structure, the WHOIS information, and the textual, meta and visual content of the website. Since some of the characteristics explained in Section 3.2 were more typical for Booters, it was decided to choose a set of them that differentiate Booter websites from any other website. By doing so 9 main features were found:

1. Number of pages - less than 50.
2. Depth level of the website - maximum 2.
3. Presence of registration page.
4. Presence of terms of service page.
5. Domain creation time - 2012 and later.
6. Obfuscated WHOIS data.

7. Protected by a DPS.
8. Specific registrar - Enom.
9. Login button on page.

Finally, in Chapter 4, based on the set of features listed above it was possible to classify whether a website is a Booter depending on the features they satisfy. Using a separate dataset consisting of 163 URLs type 1, including 66 Booters and 97 non-Booters, three approaches were proposed: (i) by defining the similarity between websites, (ii) by assigning the score to the website, and (iii) by assigning score and varying the weights of the features. The latter two approaches showed that assigning the weights increases the accuracy of the list (for the same threshold it increased with 6%). Finally, after the three approaches a decision tree was proposed in Section 4.5, which highlights the best of our knowledge on classifying Booter websites. It was possible to classify Booters with only 4% false positives. For the first work done in this area, we consider this is a satisfying result.

Based on the contributions mentioned above the goal of this thesis was reached - all the research questions were answered. The research brought however more than that, and it will be explained in the following section.

5.1 Additional contributions

Among the additional findings, it was noticed how dynamic Booters are. From overall 149 found Booters, 101 Booters are still active (37% of Booters went offline during this research, between 15 November 2014 and 12 February 2015). Comparing to the benign websites, only 1,6% went inactive during the time of this Research. Thus, we see that Booters have much shorter life span, or are less stable than benign websites. This suggests that the automated way of generating a list of Booters is necessary in future.

One of the important outcomes of this research is a dataset containing websites classified as Booters or as non-Booters. This dataset can be reused in further research, for example, to classify Booters using one of the machine learning algorithms, described in Section 2.3.1.

Additionally, the research brings more light to the significance of the problem of Booters. Up till now, researches were focused on 15 Booters [16], with the biggest amount of Booters mentioned to be 59 in [9], so the phenomenon could have been undermined in its scale. Providing a list of 149 Booters retrieved only from Google search, this thesis reveals the scale of this phenomenon. This is however just a small next step, since these were Booter websites found only in a subset of 291 URLs out of almost 32 thousands URLs found. Certainly, there are more Booters to be found and new ones will also keep appearing.

Moreover, owning a list of Booters comes with a great responsibility: it comes with a question who should be granted an access to it? The idea behind this research was to provide it to the researchers, as it is expected that this way their work is shortened with the step of finding the Booters. On the other hand, if the list is public, someone unwanted could get in possession of it and be provided with a list of Booters such person could use. As a result of this thesis, the 149 manually accessed URLs will be given to Dutch National Research and Education Network (SURFnet) to observe the users accessing Booters and (hopefully) their correlation with attacks.

5.2 Future work

The current work was a first attempt towards an automatically generated list of Booters. This leaves a lot of space for improvement in the searching approach and the classification method.

One of the desired characteristics of a list used for blacklisting is its comprehensiveness - even if 99% of Booters would be blocked, an attacker could access the remaining 1% and perform the attack. This means that the search function of the crawler should not be limited to Google Search. This is a good start - as many ill-intentioned users would use Google in order to search for Booters, the owners of Booters would like to be found in order to earn money - so they allow the websites to be indexed by

Google. However, the search function should include also other sources to retrieve URLs, especially the hacker forums, where the URLs of Booters are announced long before they appear indexed by Google. Beside Google Search and hacker forums, possible place to look for Booters is the advertisement channels of them, such as YouTube, Twitter or Facebook.

Another suggestion for the future works in determining if a website is a Booter is to use machine learning algorithms. The main reason why it was not done this way within this thesis was the lack of the reference list. Moreover, it would be smart to consider changing the way the features of Booters were described.

Regarding the reference list of Booters, it is interesting to look how this was solved in case of phishing. Before automated generation mechanisms for phishing or malicious websites, those websites were discovered and reported manually (e.g. via PhishTank [1]). It could be that the manual selection and reporting of Booter websites cannot be avoided. Good thing is that, especially in the gaming community, using Booters can be especially annoying and gamers could be willing to report Booters they know about [2].

Regarding the way that Booters' features are described, there is also many possibilities for improvement. In the proposed approach it was only verified if a website satisfies a set of features, in a binary way. Instead, it could be done by mapping the features, i.e. transforming raw data input to some representation described with certain function. This representation could be better used in machine learning tasks. Since finding the proper mapping and verifying it opens a lot of new possibilities it was kept outside of the scope of this thesis.

5.3 Limitations

In the crawler, the goal of classifying the websites was achieved with several solutions which have their own flaws. The limitations can be divided into limitation coming from gathering the necessary data, preparing the data and the types of Booters.

One of the limitations of proposed crawler is fact of using Google Search engine. It was already mentioned in Sec. 5.2 that the search function should be extended to other sources, however, the current approach gives two other limitations: the search is dependent on the positioning of the websites, and the search function used does not return all the possible URLs found by Google, because the crawler is each time detected and blocked.

Another limitation of current search is English dictionary. Booters were found also in French and German, and could be wrongly classified because of the language. However, since the Booters in language other than English are still a minority, the proposed approach is a good start.

For complete automation of the crawler, the challenge is to download the websites hosted by DDoS Protection Service (DPS) companies. The challenge was explained in section 3.1. Briefly, using TOR network it is not possible to download the website hosted by a DPS company automatically, because the IP address that the crawler uses has most likely a bad reputation.

One of the main challenges in automating a system of classification of Booters is by preparing the data. Every website is different - some of them store meta data, some of them do not; the information type returned by WHOIS is different per domain name; when downloading the website sometimes its content is retrieved and sometimes a CAPTCHA challenge. It was crucial to eliminate the noise in the processed data. This was done as well as possible, however, it could be that some additional noise appears and has to be eliminated.

Wrapping up the limitations, there can be many types of Booters. The methodology used in this thesis used certain search method, it also described most significant characteristics of overall amount of Booters. It did not investigate more than one type of Booter. It could be that what was described in this thesis is just one class of Booters - those, which are easily retrievable via means of Google Search. For that it would be advisable to investigate the future works of this thesis - that is machine learning, to verify more than one kind of Booters.

[1] `www.phishtank.com`

[2] `community.callofduty.com/thread/200766781`

Now we know that the problem of computer viruses was not solved within two or twenty years, nor was it solved by now. We also do not expect the problem of Booters to be easily solvable in coming years. The list of Booters, and the crawler developed for the purpose of this thesis will hopefully support the ongoing research on Booters mitigation. This work is just a small, but important step for mitigation of Booters, however, it should not be the last one.

MY REFLECTIONS AND ACKNOWLEDGEMENTS

Working on the Master Thesis assignment was definitely most energy consuming tasks I faced lately, and I am glad it was possible to bring it to the end. I started my assignment working part time in a web hosting company. After a while I found it difficult to switch focus. Instead of regular time of 6 months it took me 8 months to accomplish the assignment, however, after three first months I quit with my side job. Both the job and thesis required a lot of attention and I was not able to focus on two, such major projects for that long.

Time management is both your enemy and friend: bad time management could result in frustration and procrastination whereas good time management and discipline results in measurable achievements. During the process I managed to switch from a bad time management to a good one. The best idea, which I always heard about but did not manage to put in place myself is to keep writing through the process of working on Master assignment.

For the purpose of the research we wrote an academic paper which was submitted for a conference. This was a new experience, which soon will become my everyday reality: as a PhD student. The guidance behind writing such paper was very helpful, for this, my special thank you to my supervisor, Jair. Hopefully soon we find out that the paper got accepted to the conference.

One more thing which was difficult for me to accept is: no result is also a result. During my research I hit many walls. I had problems with retrieving some data, sometimes I was hoping for certain result, that I never got. But then I realised this is a result as well: having difficulties with some data or not having a pattern I hoped for is also a result.

Finally, I would like to thank you the people who helped me to be where I am: in the last months especially thank you to my daily supervisor Jair for all the feedback, and to my boyfriend Jurriën for the support.

APPENDIX A

BOOTER WEBSITES CHARACTERIZATION: TOWARDS A LIST OF THREATS

This paper was submitted and accepted to the 33rd Brazilian Symposium on Computer Networks and Distributed Systems Conference (SBRC) 2015 in Brazil [1]. The conference took place in May 2015.

Reference:

Chromik, J. J., Santanna, J. J., Sperotto, A., & Pras, A. Booter websites characterization: Towards a list of threats. In *Proceedings of 33rd Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2015)*, pages 445–458, 2015.

[1] http://sbrc2015.ufes.br/wp-content/uploads/138741.1.pdf

Booter websites characterization: *Towards a list of threats*

Justyna Joanna Chromik, José Jair Santanna, Anna Sperotto, and Aiko Pras

[1]University of Twente - The Nederlands
Design and Analysis of Communication Systems (DACS)

j.j.chromik@student.utwente.nl,{j.j.santanna,a.sperotto,a.pras}@utwente.nl

Abstract. *Distributed Denial of Service (DDoS) attacks mean millions in revenue losses to many industries, such e-commerce and online financial services. The amount of reported DDoS attacks has increased with 47% compared to 2013. One of the reasons for this increase is the availability and ease of accessibility to websites, which provide DDoS attacks as a paid service, called Booters. Although there are hundreds of Booters available, current researches are focused on a handful sample of them - either to analyse attack traffic or hacked databases. Towards a thorough understanding and mitigation of Booters, a comprehensive list of them is needed. In this paper we characterize Booter websites and demonstrate that the found main characteristics can be used to classify Booters with 85% of accuracy. The Dutch National Research and Education Network (SURFnet) has been using a list generated by our methodology since 2013, what demonstrates high relevance to the network management community and the security specialists.*

1. Introduction

Distributed Denial of Service (DDoS) attacks are considered number one operational threat on the Internet. DDoS attacks aim to make a target machine, service or network unavailable to its intended users. To perform powerful attacks, attackers (mis)use hundreds or even thousands of distributed sources, such as infected computers or misconfigured servers. For many industries, such as e-commerce and online financial services, DDoS attacks are especially devastating. To those industries, DDoS attacks cause millions in revenue losses, reputation damage, and customer attrition [Arbor Networks 2014, Ponemon Institute 2014].

The amount of reported DDoS attacks has increased with 47% compared to 2013 [Akamai Technologies 2014]. One of the reasons for this increase is the effectiveness, simplicity and availability of websites that provide DDoS attacks as a paid service, called *Booters*. These websites offer attacks for a very cheap price, for instance, less than 5 USD [Karami and McCoy 2013], powerful enough to put most of small and medium-sized enterprise companies' websites offline [Santanna et al. 2015b]. Researches on mitigation of Booters phenomenon focus basically in two areas: (i) characterising the attacks [Santanna et al. 2015b] and (ii) analysing the leaked databases [Santanna et al. 2015a]. Although both of them have a clear contribution towards Booter mitigation, in order to address the whole phenomenon (not a particular set of Booters), they fail in a key element: a relevant, updated, and extensive list of Booters.

In order to help security specialists to retrieve such list and get a thorough understanding about the Booter phenomenon, the goal of this paper is to reveal the main characteristics of Booter websites. By using those characteristics we show that a Booter list can be retrieved with high accuracy. The main contributions of this paper are the following:

- To provide a relevant set of search terms to retrieve URLs related to Booters and a crawler to retrieve all needed information of the websites, such as the structure, visual and textual content, and the WHOIS information (Section 3).
- To reveal the 9 main characteristics of Booter websites, achieved by an extensive analysis of features used to classify websites (Section 4).
- To demonstrate that the 9 main characteristics of Booter websites can classify Booter with an accuracy of 85% (Section 4.6).

Since 2013, a Booter list generated based on our methodology have been used by the Dutch National Research and Education Network (SURFnet) to observe the users accessing Booters and their correlation with attacks. This is just one example that demonstrates high relevance of our work to the network management community and the security specialists. Only by knowing the threats we can mitigate them.

2. Related work

In the literature there is no work related to Booter websites classification. Therefore, in this section we focus on approaches that classify generic websites based on features. Although the existing approaches have several different goals we focus on specifically three of them: (i) approaches that classify the subject or type of a website [Lindemann and Littig 2006, Rajalakshmi and Aravindan 2011, Lindemann and Littig 2007, Kovacevic et al. 2004], (ii) approaches that aim to filter websites [Jo et al. 2013, Hammami et al. 2006], and (iii) approaches that aim to generate blacklists [Ma et al. 2009]. The features used in those approaches could possibly be used in Booter area. The list of features that we found is the following:

1. **URL**: this feature discloses the overall composition of a URL. Usually a URL is composed of three elements: (i) network application protocol, (ii) the URL host name, and (iii) the URL path name. For example, in *http://www.domainname.tld/path/to/the/article.html*, 'http://' is the protocol, 'www.domainname.tdl' is the host name, and '/path/to/the/article.html' is the path name.
2. **Website structure**: is a feature that reveals, among others, two aspects: (i) the website depth level and (ii) the number of known pages of a website. The former is a terminology defined by us that analyses the number of slashes ('/') a URL path has. For example, the URL *http://www.domainname.tld/path/to/the/article.html* has depth level '4'. The latter aspect is the number of indexed webpages that have the same host name and are reached by Google Search engine. For example, the number of know pages is 2 if Google Search returns 2 pages that contain a same *www.domainname.tld*, such as *www.domainname.tld/1.html* and *www.domainname.tld/2.php*.
3. **WHOIS**: is a feature that reveals information of domain names, such as (i) the registration date, (ii) the owner, (iii) the nameservers related to that domain name, and (iv) the entity responsible for the domain registration (i.e., registrar).

4. **Page content**: is a feature that reveals the elements of a website such as (i) the textual description, (ii) the meta data, and (iii) the visual content, (e.g., buttons, tables, and figures).

In Table 1 we summarize the features that each work addresses to classify websites. Such table shows that most of the existing works focus on the URL, website structure, and the website content. WHOIS and visual content are less popular, but definitely worth investigating. Therefore, in the next section we investigate those four features to characterize Booters.

Paper	URL	Website structure	WHOIS	Website content		
				Textual	Meta	Visual
[Lindemann and Littig 2006]		x			x	
[Rajalakshmi and Aravindan 2011]	x					
[Lindemann and Littig 2007]	x	x				
[Jo et al. 2013]	x		x	x	x	
[Hammami et al. 2006]		x		x	x	x
[Ma et al. 2009]	x	x	x			
[Kovacevic et al. 2004]						x

Table 1. Website features used to characterize websites.

3. The search terms, legal considerations, and our crawler

To investigate the four features described in the previous section, firstly we describe our methodology to define a list of search terms that we use to retrieve Booter websites. We give a special attention to the first search term that we found: "stresser", because it lead us to some legal considerations. After that we describe a crawler that we develop to investigate Booter websites features.

3.1. The first search term and legal considerations

In order to find a list of search terms, with which we can retrieve Booter websites, we started using the most obvious term: "booter". By using such term we found many URLs related to the word "stresser". By definition, Booters should be different from stressers. Booter is a website that provides DDoS attacks as a paid service [Santanna and Sperotto 2014], while by definition a stresser is a company (usually accessed via a website) that provides stress testing on a given system [Rouse 2007].

Both definitions are almost the same because performing a DDoS attack is a way to "stress test" a target system. However, note that stressers aim to intentionally test their customers' systems beyond regular operational capacity, while Booters are used by their clients to "stress test" a third party service. Another important characteristic is that (in theory) stressers perform their tests deliberately and in a test lab, not affecting other entities.

Putting the theoretical definitions aside, we compare the services offered by both, a Booter and a stresser. To do so, we analyse the top one result for the search terms "booter" and "stresser". We decide to anonymize the URL/names of the Booter and the stresser to avoid legal or ethical implications. Table 2 shows our findings.

Attack types		Booter A	Stresser B
Layer 3&4	TCP	✓	
	SSYN	✓	✓
	ESSYN	✓	
	UDP	✓	✓
	UDP-LAG	✓	✓
	DRDoS		✓
	CHARGEN	✓	
Application layer	GET	✓	
	HEAD	✓	
	POST	✓	
	RUDY	✓	✓
	ARME	✓	✓
	SLOWLORIS	✓	✓

Table 2. Services offered by a Booter and a stresser

The first observation is that technically the Stresser B offers the same resource exhaustion actions (attacks) as Booter A. The second observation is that both offer most of the known types of DDoS attacks, which target the Layers 3&4 and the application layer. Our paper has no intention to describe how each type of attack works, for more information see [Mirkovic and Reiher 2004]. The third observation is that although Booter A does not explicitly offer Distributed Reflection Denial of Service (DRDoS), CHARGEN is a type of DRDoS. Therefore, both Booter A and Stresser B offer the strongest type of DDoS attacks reported nowadays. Finally, we notice that Stresser B has no restriction in relation to the target of "stress test". Given this freedom, the customers can perform attacks against third party services. Therefore, through these four observations we conclude that Stresser B is also a Booter.

The question that arise is: why Booters advertise themselves as stressers? The answer, which we found in hacker forums and many blogs, is that Booters want to avoid legal problems by hiding illegal actions (DDoS attacks using compromised machines) behind legal services (stress testing) [Kassner 2013, Musthaler 2012]. Booters have another strategy to avoid legal problems. Instead of advertising themselves as stressers, they include in their websites Terms of Services (ToS). It is a legal agreement composed by a set of rules that clients need to follow to use their services. Table 3 shows parts of ToS found on Booters.

	Terms of Service statements
Booter C	"We are not responsible for how ever you use this stresser"
Booter D	"Illegal activity which occurs in your account is [...] associated to you"
Booter E	"Anything you do while on Booter E is your own responsibility"

Table 3. Examples of text included in Terms of Service

We observe in Table 3 that Booters clearly do not take any legal responsibilities for user's actions, although their core business is to offer DDoS attacks against anyone and anything connected to the Internet. The legal aspect of Booters is a nudging subject, which requires attention, but will be out of the scope of this research. From technical point

of view we see that stressers are used as Booters. Because of this we use term "stresser" in the set of terms to retrieve Booter websites. In the next subsection we describe how did we find the other search terms.

3.2. The other search terms

To find other terms we search on Google for materials related to Booters and stressers. Through an extensive literature we found other two terms: *"ddos-for-hire"* [Krebs 2013a] and *"ddoser"* [Safe Keys 2013]. Finally, by using the two former terms ("booter" and "stresser") and the later terms ("ddos-for-hire" and "ddoser") we search the literature using Google Scholar. This Google service aggregates most of the the digital libraries, academic publishers, and repositories worldwide, including IEEE Xplore Digital Library and ACM Digital Library. Through Google Scholar, and closing our set of terms, we found "ddos-as-a-service" [Karami and McCoy 2013]. Note that the reference on the side of earlier mentioned terms is not necessarily the first to use/define these terms, but the place that we have found them.

3.3. Our crawler

After defining the five search terms (*"booter", "stresser", "ddos-for-hire", "ddoser", and "ddos-as-a-service"*), in this section we describe our approach to retrieve a list of URLs related to Booters and the additional information needed to classify them according to the four website features (described in Sec. 2). To do so, we develop a crawler to fulfil the following requirements:

1. Retrieve as many URLs related to Booters as possible based of the list of search terms. These retrieved URLs are called in this paper as "potential Booters".
2. Extend the URLs retrieved in the previous requirement to include all known pages related to these URL domain names.
3. Retrieve the WHOIS information of the URL domain names.
4. Download the Booter website content.

In order to retrieve the URLs related to Booters (first requirement), we use a python search script [1]. This approach was chosen over other approaches [2][3][4] because it is not limited on the amount of results able to retrieve (e.g., 24 URLs). To find all web pages related to a website (second requirement) we use the Google Search with operator "site:". This operator returns as result only URLs within the domain requested, for example the output of *"site:name-of-potential-booter.com"* is a list of pages (URL paths) in the domain *"name-of-potential-booter.com"*.

To retrieve the WHOIS information (third requirement) we use pythonwhois package [5]. Although such library is able to return more than 20 different information about a domain name, in our WHOIS analysis (Sec. 4.4) we only use 4 of them: (i) the creation date, (ii) the registrar of the domain name, (iii) the nameservers that the domain is pointing to, and (iv) the contacts of the domain administrator. To fulfil the last requirement and

[1] http://breakingcode.wordpress.com/2010/06/29/google-search-python/
[2] http://googolplex.sourceforge.net/
[3] http://www.catonmat.net/blog/python-library-for-google-search/
[4] http://googlesystem.blogspot.nl/2008/04/google-search-rest-api.html
[5] http://cryto.net/pythonwhois/

download the content of Booter websites, we firstly decide that we should keep our identity anonymous. This precaution was taken because a security specialist was successively attacked after starting investigating Booters [Krebs 2013b]. Therefore, we use The Onion Router (TOR) network that enables online anonymity by encrypting and bouncing the traffic through networks and open relays severs. Secondly, another requirement needed to download the potential-booter-webpages is to support JavaScript. To do so, we use the Selenium browser[6], which is a library that emulates a generic Web browser and therefore is able to support JavaScript.

In summary, our crawler was build to attend the requirements to analyse Booters. Although we use specific libraries and APIs to build our crawler, these were our decisions on how to fulfil the requirements. Therefore other libraries can be used. Note that our crawler can be extended to perform an automated classification of Booter websites, but first the characteristics of the website features should be analysed. More ideas about future works are written in the last section of this paper.

4. Analysis on Booter websites features

In this section we analyse the features of Booter websites. Firstly we describe the list of URLs (Sec. 4.1) used to perform our analyses. Secondly, we describe our analysis based on the URL characteristics (Sec. 4.2), the website structure (Sec. 4.3), the WHOIS information (Sec. 4.4), and the website content (Sec. 4.5). We close this section highlighting the features that are more representative to classify Booters (Sec. 4.6).

4.1. The list of URLs

By using the search terms and our crawler described in the previous section (Sec. 3.3) we retrieved 1238 URLs, on 15th September 2014. From these 1238, 230 URLs were retrieved by using the term "booter", 370 by using "stresser", 265 "ddoser", 199 "ddos-for-hire", and 174 "ddos-as-a-service". In order to verify if these search terms are representative and distinct we analyse the intersection of retrieved URLs, showed in Table 4.

Search Terms	booter	stresser	ddoser	ddos-for-hire	ddos-as-a-service
booter	**230**	13	3	1	1
stresser	13	**370**	3	0	0
ddoser	**3**	3	**230**	1	1
ddos-for-hire	1	0	1	**174**	11
ddos-as-a-service	1	0	1	11	**199**

Table 4. Intersection of retrieved URLs based on different search terms

Based on Table 4 we notice that our search terms are distinct: very small number of same URLs are retrieved using different search terms, for example only 3 URLs were retrieved using the terms "booter" and "ddoser". " We are aware that our crawler did not retrieve all URLs related to those search terms because each search process was interrupted by a HTTP error (i.e., 503: service unavailable). Although our chosen approach retrieves more URLs than other current approaches, Google Search is still able to detect and block our crawler. Even though, the retrieved list of URLs is sufficient enough to analyse Booters features, since it contains both Booter and non-Booter websites.

[6]http://www.seleniumhq.org

4.2. URL analysis

Based on the list with 1238 URLs we analyse the first feature found in our survey: the URL composition. Table 5 shows a summary of the types of URLs found.

URL Type	URL	# URL retrieved
1	potential-booter.com	71
2	potential-booter.com/login.php	1167
3	www.domain.com/potential-booter	
4	potential-booter.domain.com	

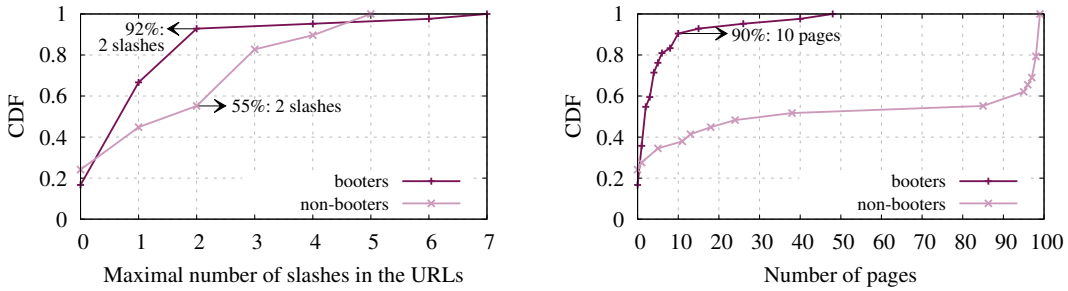
Table 5. Examples of retrieved URLs

In Table 5 we observe four different types of URLs: with only a host name (type 1), with host name and path (type 2), where Booter is a page of a website (type 3), and where a Booter is a subdomain of a domain name (type 4). We want to focus on the URL type that has the highest probability to be a Booter. Note that often URLs type 2 are subpages of URLs type 1. Thus, we decide to analyse only the type 1. Although URLs type 3 and 4 can potentially contain Booters, we noticed that it is more likely that they provided information about Booters, not the websites themselves.

Each one of the 71 URLs type 1 are later called as *potential Booter*, and they will be further classified. After a manual analysis of those potential Booters we found that 42 are Booter websites and 29 are non-Booter websites. All the further sections will be based on the 71 potential Booters.

4.3. Website structure analysis

After the filtering process described in the previous section, we analyse the structure of these 71 potential Booters. Figure 1 summarizes our findings.



(a) Depth of websites.

(b) Known pages.

Figure 1. CDF of website structure aspects.

In Figure 1, graphs of the Cumulative Distribution Function (CDF) of the website structure are shown. While Figure 1(a) shows the CDF of depth level of these 71 website, Figure 1(b) shows the CDF of the known pages. Both graphs show the results for Booters and non-Booters (classified manually in the previous section). We observe in Figure 1(a) that 92% of Booters have their website depth up to 2 levels, whereas only 55% of non-Booters have the same depth. Considering Figure 1(b) we observe that 90% of Booters have 10 or less pages. In addition, Booter websites never exceed 50 known pages, what is

an interesting observation that can be used to eliminate non-Booters from a set of URLs (that in this analysis is almost 50%).

By in depth analysis of the 71 potential Booters we notice that some page names appear more than once, such as “register”, “ToS”, “plans”, “buy”, and “hub”. The number of times we observed those pages is shown in Figure 2. We notice that pages as Terms of Service (ToS) and “registration” are more often appearing when analysing Booters. Almost 60% of Booters have a registration page (24/42) and around 40% of Booters have ToS page (17/42).

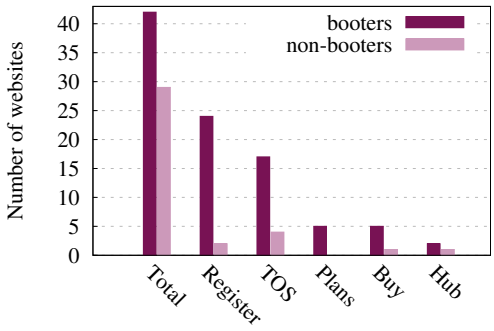


Figure 2. Page names analysis.

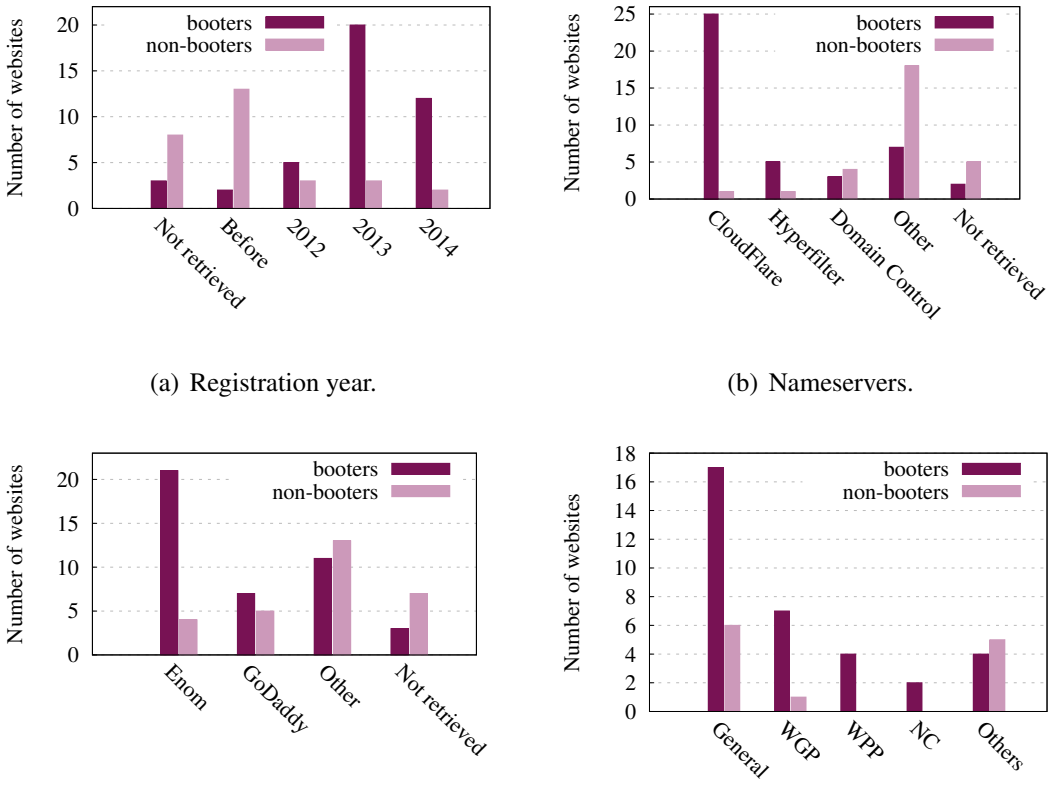
The other page names on the Figure 2, plans, buy, and hub, are not that promising. For example only 5 out of 42 Booters had pages referring to plans or redirecting to purchase page (buy). Although the page “hub” is crucial to Booters, because through that page a user can start an attack, it was found only twice. The reason for that could be that this page is accessible only after logging into the Booter and it is not indexed by Google.

4.4. WHOIS analysis

After analysing the website structure we check the WHOIS information of the 71 potential Booters retrieved by our crawler. We present our findings in Figure 3.

In Figure 3 we observe four different information related to WHOIS. From the first one, in Figure 3(a), we observed that more than 88% of the Booter domain names (37/42) were registered in 2012 (5), 2013 (20), and 2014 (12), whereas a bit less than half of the non-Booter websites were registered before 2012 (13/29). Overall, it shows that most of the Booters by our crawler are relatively new. By analysing the namesevers of the retrieved Booter domain names, showed in Figure 3(b), we are not surprised to observe that almost 60% of Booters (25/42) are associated to CloudFlare. It was already pointed out in [Santanna et al. 2015b] that Booter subscribe services from CloudFlare to be protected against DDoS attacks from the competitors (other Booters). However, we are surprised to observe another company offering DDoS protection, Hyperfilter. It is the first time that this company is observed in this phenomenon. Also in Figure 3(b), we confirm that majority of non-Booters have more variety in nameservers (18/29).

By analysing the companies that provide the domain registration (registrar), showed in Figure 3(c), we notice that half of Booters (21/42) have their domain registered with Enom, which is a known domain registrar. This is probably because of cheap



(a) Registration year.

(b) Nameservers.

(c) Registrar.

(d) Contact of the domain name owner.

Figure 3. WHOIS information.

price, but also because Enom offers the "WhoisGuard" service used for hiding the contact details. This information is more evident when we look into the contact of the responsible for the domain names. In Figure 3(d) we observe that more than 40% of Booters (17/42) have their contact hide from the WHOIS information. Usually the contact is hidden by many services, among them: "WhoisGuard Protected" (WGP - 7/42), "Whois Privacy Protect" (WPP - 4/42), and "NameCheap" (NC - 2/42).

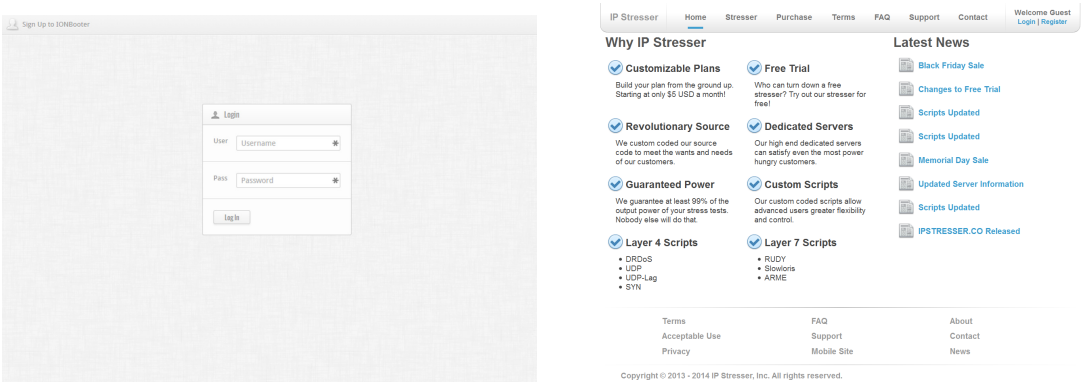
4.5. Website content analisys

The last feature that we use to characterize Booter websites is based on their content analysis. This analysis is divided in two parts: (i) the visual interface and (ii) the meta data, that is: the description and the keywords used to define Booter websites.

4.5.1. Visual interface

By manually accessing the 42 Booters we notice that they have only two completely different types of first page: or (i) a very simple first page containing a simple login interface, for example showed in Figure 4(a), or (ii) a verbose page full of textual content and appealing advertisements, for example shown in Figure 4(b). For both types Booters provide a very simple and user-friendly interface. The most remarkable finding related

to the visual interface was that while more than a half of Booters (22/42) has a “login” button in their main page only 1 non-Booter has such button.



(a) Booter F - login page.

(b) Booter E - login page.

Figure 4. Examples of login pages.

4.5.2. Meta data

When we look into meta data used to define Booters, almost 62% (26/42) has neither description nor keywords, showed in Figure 5. This result is very similar and consistent with the simplicity of Booters visual interface (described in the previous section). We also observe that there is not a clear distinction between non-Booters that use meta data. Therefore, although the meta data can help on the understanding the purpose of a website, because of insufficient information, we can not use this feature to define if a website is a Booter or not.

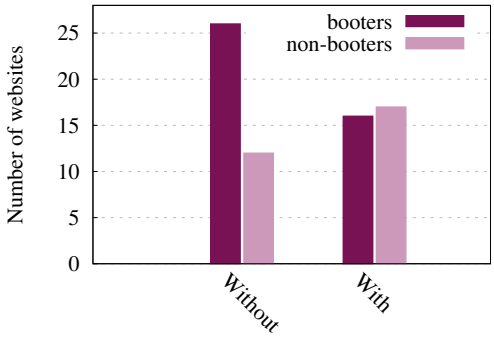


Figure 5. Meta data.

Note that in theory meta data is fundamental to have a better positioning in search engines, such as Google Search. Therefore we decide to correlate the popularity of a Booter with the existence or not of meta data. To do so we use the Alexa’s Rank [7] to get information about the 42 Booter webpages. By doing so, in general we observed that Booters with meta data have a higher ranking, for example in Table 6 two Booters are

[7] http://www.alexa.com

shown. Booter E that has meta data was ranked around the 273k position, while Booter F had a thousand times lower position. Note that the highest Alexa's Rank value is 1 (that usually has Google or Facebook).

Name	Meta data		Alexa's Rank
	Description	**Keywords**	
Booter E	Powerful and Affordable Stress Testing	stresser, denial of service, dos, ddos, dr-dos, syn, ssyn, udp, sudp, udp-lag, rudy, slowloris, arme	272.979
Booter F	-None-	-None-	2.580.782

Table 6. Examples of Booter meta data and their popularity

We are aware that meta data is not the only determining factor to have a higher ranking in the Internet. However, we are positively surprised to observe a clear relation between both factors.

4.6. A summary of the representative Booter features and a brief validation

The goal of this section is to summarize the most relevant features of Booters, which are based on what did we learn from our analysis. In addition we perform a brief validation to show that those features can be used to classify Booters.

In Section 4.2 we conclude that URL type 1, i.e. those that are composed by only the URL hostname, are more suitable to be a Booter. In Section 4.3 we notice that all Booters have less than 50 subpages and in general 2 levels of depth. We also observed that websites, which have pages "register" and "tos" are more often Booters. When it comes to WHOIS information, in Section 4.4, we highlight that Booters tend to use services from DDoS protection companies, such as CloudFlare or HyperFilter. The domain names used by Booters are most likely registered in 2012 or later, and the used registrar is most likely Enom. Moreover, information about the owners of Booters is hidden using services such as "WhoisGuard". Finally, in Section 4.5.1, we showed that Booters often have a simple interface with a login button. Through those observations we define a list with the main features that has a higher change to classify generic URLs as Booter websites:

1. Number of pages less than 50.
2. Depth level of the website of maximum 2.
3. Presence of registration page.
4. Presence of terms of service page.
5. Domain creation time 2012 and later.
6. Obfuscated WHOIS data.
7. Protected by a DPS.
8. Specific registrar: Enom.
9. Login button on page.

We decide to analyse if indeed the list of features can be used to classify Booters. To do so, we collected 3248 URLs related to Booters using the the search terms (Sec. 3.2) and our crawler (Sec. 3.3), on 30th November 2014. After filtering URLs based on URL type 1, we found 156 to perform a manual classification. From the manual classification we found 87 Booters websites and 69 as non-Booters. Then, for each one of the 156 URLs we count how many of the 9 features they have. The results are shown in Figure 6

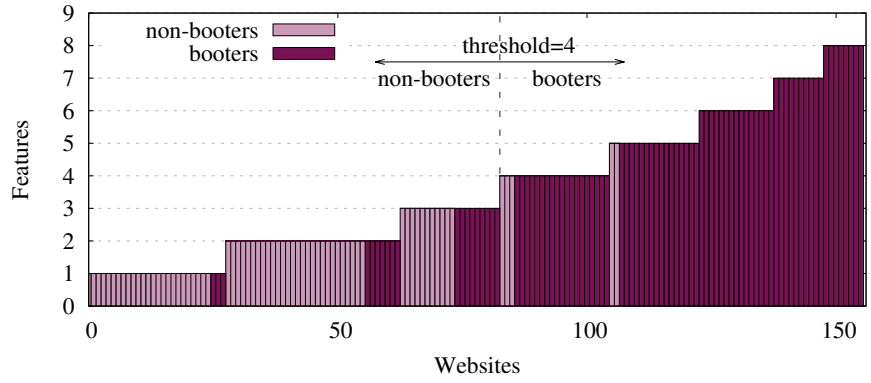


Figure 6. Representative features related to Booters per websites.

Table 7. Accuracy of threshold used to classify Booters

Threshold	True Positive	True Negative	Accuracy	False Positive	False Negative
3	78	52	**83%**	16	10
4	**69**	**63**	**85%**	**5**	**19**
5	50	66	**74%**	2	38
6	33	68	**65%**	0	55

The results in Figure 6 show clearly that the more features related to Booters a website has, the higher is the chance that such website is a Booter. Then, in the future, those features indeed can be used to automate the classification of Booters. However, the only problem that future researchers will have to determine where the threshold of features should be placed. Table 7 show the the accuracy of the threshold of our analysis.

Based on the retrieved URLs and our 9 features we found that URLs that have 4 or more features (threshold 4) show better results in terms of accuracy ((true negative +true positive)/population), 85%. Although this result is very promising, further research should be done to increase the accuracy.

5. Conclusion and Future work

By aiming to help security specialists to retrieve a comprehensive list of Booters, we performed a thorough characterization of Booter websites. First, we discovered the most common features to analyse websites. Then, we developed a crawler to retrieve a representative set of URLs and additional information to perform analysis. After an extensive analysis we highlight the 9 main characteristics of Booter websites. Finally, by using those characteristics we demonstrate that a list of Booters can be retrieved and classified with 85% of accuracy.

We conclude that although the 9 features are representative in the classification of Booter websites, more research needs to be done to achieve a fully automated methodology to retrieve a comprehensive list of Booters. As a future work, we aim to extend the number of retrieved URLs by adding additional sources of information, such as hacker forums, twitter, and youtube. Moreover, the analysis should include URLs type 3 and 4.

More research should be done to improve accuracy, for example by in depth analysis of the meta data or by assigning weights to the features. Since we expect Booters to

evolve, we consider further investigation of the analysis using machine learning. Moreover we urge for investigating the legal aspects of what is allowed as stress testing, including definition of procedures for authentication and verification of identity of people using these services.

Acknowledgments

This work was funded by the Network of Excellence project FLAMINGO (ICT-318488), which is supported by the European Commission under its Seventh Framework Programme.

References

[Akamai Technologies 2014] Akamai Technologies (2014). Prolexic Quarterly Global DDoS Attack Report (Q1 2014). http://www.prolexic.com/knowledge-center-ddos-attack-report-2014-q1.html.

[Arbor Networks 2014] Arbor Networks (2014). The Risk vs. Cost of Enterprise DDoS Protection. http://pages.arbornetworks.com/WebsiteRiskvsCost_Ent_DDoS_Protection.html.

[Hammami et al. 2006] Hammami, M., Chahir, Y., and Chen, L. (2006). Webguard: A web filtering engine combining textual, structural, and visual content-based analysis. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):272–284.

[Jo et al. 2013] Jo, I., Jung, E., and Yeom, H. Y. (2013). Interactive website filter for safe web browsing. *Journal of Information Science and Engineering*, 29(1):115–131.

[Karami and McCoy 2013] Karami, M. and McCoy, D. (2013). Understanding the Emerging Threat of DDoS-as-a-Service. In *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX.

[Kassner 2013] Kassner, M. (2013). What's better than creating your own DDoS? renting one. http://www.techrepublic.com/blog/it-security/whats-better-than-creating-your-own-ddos-renting-one/.

[Kovacevic et al. 2004] Kovacevic, M., Diligenti, M., Gori, M., and Milutinovic, V. (2004). Visual adjacency multigraphs-a novel approach for a web page classification. In *Proceedings of SAWM04 workshop, ECML2004*.

[Krebs 2013a] Krebs, B. (2013a). Ragebooter: 'Legit' DDoS Service, or Fed Backdoor? http://krebsonsecurity.com/2013/05/ragebooter-legit-ddos-service-or-fed-backdoor/.

[Krebs 2013b] Krebs, B. (2013b). The world has no room for cowards. http://krebsonsecurity.com/2013/03/the-world-has-no-room-for-cowards/.

[Lindemann and Littig 2006] Lindemann, C. and Littig, L. (2006). Coarse-grained classification of web sites by their structural properties. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, pages 35–42.

[Lindemann and Littig 2007] Lindemann, C. and Littig, L. (2007). Classifying web sites. In *Proceedings of the 16th international conference on World Wide Web*, pages 1143–1144.

[Ma et al. 2009] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1245–1254.

[Mirkovic and Reiher 2004] Mirkovic, J. and Reiher, P. (2004). A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53.

[Musthaler 2012] Musthaler, L. (2012). DDoS-as-a-Service? You Betcha! It's Cheap, It's Easy, and It's Available to Anyone. `http://www.securitybistro.com/?p=4121`.

[Ponemon Institute 2014] Ponemon Institute (2014). Cyber Security on the Offense: A Study of IT Security Experts. `http://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/CyberSecurityontheOffense.pdf`.

[Rajalakshmi and Aravindan 2011] Rajalakshmi, R. and Aravindan, C. (2011). Naive bayes approach for website classification. In *Information Technology and Mobile Communication*, pages 323–326. Springer.

[Rouse 2007] Rouse, M. (2007). Stress testing. `http://searchsoftwarequality.techtarget.com/definition/stress-testing`.

[Safe Keys 2013] Safe Keys (2013). Top 10 DDoSer's (Booters, Stressers. `http://www.safeskyhacks.com/Forums/showthread.php?39-Top-10-DDoser-s-(Booters-Stressers)`.

[Santanna et al. 2015a] Santanna, J. J., Durban, R., Sperotto, A., and Pras, A. (2015a). Inside booters: an analysis on operational databases. In *14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*. <to appear>.

[Santanna and Sperotto 2014] Santanna, J. J. and Sperotto, A. (2014). Characterizing and mitigating the ddos-as-a-service phenomenon. In *8th IFIP International Conference on Autonomous Infrastructure, Management, and Security (AIMS 2014)*, pages 74–78.

[Santanna et al. 2015b] Santanna, J. J., van Rijswijk-Deij, R., Sperotto, A., Hofstede, R., Wierbosch, M., Granville, L. Z., and Pras, A. (2015b). Booters - an analysis of ddos-as-a-service attacks. In *14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*. <to appear>.

APPENDIX B

MANUALLY CLASSIFIED BOOTERS

This section contains sensitive data and thus the content is not publicly available. To request the content of this section, please contact the author of the thesis.

APPENDIX C

BOOTERS USED IN THE THESIS

This section contains sensitive data and thus the content is not publicly available. To request the content of this section, please contact the author of the thesis.

APPENDIX D

SOURCES OF BOOTERS

This section contains sensitive data and thus the content is not publicly available. To request the content of this section, please contact the author of the thesis.

BIBLIOGRAPHY

[1] Darren Anstee, Andrew Cockburn, Gary Sockrider, and Carlos Morales. Worldwide Infrastructure Security Report, Volume IX. http://pages.arbornetworks.com/rs/arbor/images/WISR2014.pdf, 2014.

[2] Arbor Networks. The Risk vs. Cost of Enterprise DDoS Protection. http://pages.arbornetworks.com/WebsiteRiskvsCost_Ent_DDoS_Protection.html, 2014.

[3] Ponemon Institute. Cyber Security on the Offense: A Study of IT Security Experts. http://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/CyberSecurityontheOffense.pdf, 2014.

[4] Akamai Technologies. Akamai's [state of the internet] / security. http://www.stateoftheinternet.com/downloads/pdfs/2014-internet-security-report-q4.pdf, 2014.

[5] M Karami and D McCoy. Understanding the Emerging Threat of DDoS-as-a-Service. In *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX, 2013.

[6] Integrails Inc. DDoS Protection Bypass Techniques. https://media.blackhat.com/us-13/US-13-Nixon-Denying-Service-to-DDOS-Protection-Services-WP.pdf, 2013.

[7] BBC. Hacktivists step up web attack volumes. http://www.bbc.com/news/technology-31000908, 2015.

[8] Jason Lackey. A New Twist on Denial of Service: DDoS as a Service. "http://blogs.cisco.com/security/a_new_twist_on_denial_of_service_ddos_as_a_service".

[9] J. J. Santanna and A. Sperotto. Characterizing and mitigating the ddos-as-a-service phenomenon. In *8th IFIP International Conference on Autonomous Infrastructure, Management, and Security (AIMS 2014)*, pages 74–78, 2014.

[10] Brian Krebs. The Obscurest Epoch is Today. http://krebsonsecurity.com/2013/03/the-obscurest-epoch-is-today.

[11] J. J. Santanna, R. van Rijswijk-Deij, A. Sperotto, R. Hofstede, M. Wierbosch, L. Z. Granville, and A. Pras. Booters - an analysis of ddos-as-a-service attacks. In *14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, 2015. <to appear>.

[12] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[13] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 449–460. ACM, 2014.

[14] Christian Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, February 2014.

[15] NSFOCUS Information Technology Co. Analysis of DDoS Attacks on Spamhaus and recommended solution. http://en.nsfocus.com/SecurityView/Analysis%20of%20DDoS%20Attacks%20on%20Spamhaus%20and%20recommended%20solution-EN-20130510.pdf.

[16] J. J. Santanna, R. Durban, A. Sperotto, and A. Pras. Inside booters: an analysis on operational databases. In *14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, 2015. <to appear>.

[17] Stress testing. http://en.wikipedia.org/wiki/Stress_testing, 2014.

[18] Margaret Rouse. Stress testing. http://searchsoftwarequality.techtarget.com/definition/stress-testing, 2007.

[19] Unknown. Top 10 stressers-booters-ddosers. http://top10stressers.com.

[20] Brian Krebs. DDoS Services Advertise Openly, Take PayPal. http://krebsonsecurity.com/2013/05/ddos-services-advertise-openly-take-paypal, 2013.

[21] Booters Review. http://booters-review.blogspot.nl/.

[22] DDoS'er as Service - a camouflage of legit stresser/booter/etc. http://blog.malwaremustdie.org/2014/06/ddoser-as-service-camouflation-of-legit.html.

[23] Michael Kassner. What's better than creating your own DDoS? renting one. http://www.techrepublic.com/blog/it-security/whats-better-than-creating-your-own-ddos-renting-one/, 2013.

[24] Wikipedia. Terms of service. http://en.wikipedia.org/wiki/Terms_of_service.

[25] Dhwani Garg. DDOS Mitigation Techniques - A Survey. In *International Conference on Advanced Computing, Communication and Networks*, pages 1302–1309, 2011.

[26] Michael Walfish, Mythili Vutukuru, Hari Balakrishnan, David Karger, and Scott Shenker. DDoS defense by offense. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 303–314, 2006.

[27] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. Phishnet: predictive blacklisting to detect phishing attacks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5, 2010.

[28] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1245–1254. ACM, 2009.

[29] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*, pages 6–6. USENIX Association, 2010.

[30] Insoon Jo, Eunjin Jung, and Heon Young Yeom. Interactive website filter for safe web browsing. *Journal of Information Science and Engineering*, 29(1):115–131, 2013.

[31] W Ph Stol, HKW Kaspersen, J Kerstens, ER Leukfeldt, and AR Lodder. Governmental filtering of websites: The dutch case. *Computer Law & Security Review*, 25(3):251–262, 2009.

[32] European Parliament and Council of the European Union. Directive 2013/40/EU, 2013.

[33] Christoph Lindemann and Lars Littig. Coarse-grained classification of web sites by their structural properties. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, pages 35–42, 2006.

[34] R Rajalakshmi and C Aravindan. Naive bayes approach for website classification. In *Information Technology and Mobile Communication*, pages 323–326. Springer, 2011.

[35] Christoph Lindemann and Lars Littig. Classifying web sites. In *Proceedings of the 16th international conference on World Wide Web*, pages 1143–1144, 2007.

[36] Mohamed Hammami, Youssef Chahir, and Liming Chen. Webguard: A web filtering engine combining textual, structural, and visual content-based analysis. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):272–284, 2006.

[37] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, and Veljko Milutinovic. Visual adjacency multigraphs-a novel approach for a web page classification. In *Proceedings of SAWM04 workshop, ECML2004*, 2004.

[38] Yue Zhang, Jason I Hong, and Lorrie F Cranor. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*, pages 639–648. ACM, 2007.

[39] Leslie Daigle. WHOIS Protocol Specification. RFC 3912, September 2004.

[40] Hans-Peter Kriegel and Matthias Schubert. Classification of websites as sets of feature vectors. In *Databases and applications*, pages 127–132. IASTED, 2004.

[41] Brian Krebs. Ragebooter: 'Legit' DDoS Service, or Fed Backdoor? `http://krebsonsecurity.com/2013/05/ragebooter-legit-ddos-service-or-fed-backdoor/`, 2013.

[42] Safe Keys. Top 10 DDoSer's (Booters, Stressers. `http://www.safeskyhacks.com/Forums/showthread.php?39-Top-10-DDoser-s-(Booters-Stressers)`, 2013.

[43] Mario Vilas. Quickpost: Using Google Search from your Python code. `http://breakingcode.wordpress.com/2010/06/29/google-search-python/`, 2010.

[44] Googolplex Page. `http://googolplex.sourceforge.net/`, 2004.

[45] Peteris Krumins. Python Library for Google Search. `http://www.catonmat.net/blog/python-library-for-google-search/`, 2009.

[46] Alex Chitu. Google Search REST API. `http://googlesystem.blogspot.nl/2008/04/google-search-rest-api.html`, 2008.

[47] Sven Slootweg. pythonwhois. `http://cryto.net/pythonwhois/`, 2014.

[48] Brian Krebs. The World Has No Room For Cowards. `http://krebsonsecurity.com/2013/03/the-world-has-no-room-for-cowards/`, 2013.

[49] BBC. How to find geolocation of an ip address? /urlhttp://www.iplocation.net/, 2015.

[50] Leo Liberti, Carlile Lavor, Nelson Maculan, and Antonio Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[51] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 1–8. ACM, 2007.