Detecting Malware Domains: A Cyber-Threat Alarm System

Khalifa AlRoum[1], Abdulhakim Alolama[1], Rami Kamel[1], May El Barachi[2], and Monther Aldwairi[1(✉)]

[1] Zayed University, Khalifa City B, Abu Dhabi, United Arab Emirates
{M80006834, M80006863, M80006762, monther.aldwairi}@zu.ac.ae
[2] University of Wollongong Dubai, Knowledge Village, Dubai, U.A.E
MaiElbarachi@uowdubai.ac.ae

Abstract. Throughout the years, hackers' intentions' varied from curiosity, to financial gains, to political statements. Armed with their botnets, bot masters could crash a server or website. Statistics show that botnet activity accounts for 29% of the Internet traffic. But how can bot masters establish undetected communication with their botnets? The answer lies in the Domain Name System (DNS), using which hackers host their own domain and assign to it changing IP addresses to avoid being detected. In this paper, we propose a multi-factor cyber-threat detection system that relies on DNS traffic analysis for the detection of malicious domains. The proposed system was implemented, and tested, and the results yielded are very promising.

Keywords: DNS analysis · Cyber-threat · Malicious domains' detection · Botnets

1 Introduction

With the rapid increase in the newly registered domains around the world, the challenge of identifying the malicious domains from the legitimate ones becomes more complicated. It is well known that without the domain name system (DNS), surfing the Internet would become nearly impossible. Hackers around the world use the DNS to direct the traffic coming from their botnets, so if a system admin of a specific network blocks a traffic from flowing to a suspicious IP address, the hacker still can get the traffic by updating his domain with a new IP address. Blocking the traffic flowing to suspicious IP addresses would solve the problem in the past, but nowadays using the frequent DNS entry change feature, this technique is less effective. Therefore, there is a need for building a system for the detection of malicious domains rather than suspicious IP addresses.

Many cyber-attacks have been launched using botnets, - a botnet consisting of a group of machines controlled by a hacker, via a command and control center. Such botnets cannot only be used to launch cyber-attacks, but also to collect a variety of useful information for hackers. The importance of botnets is such that some hackers may lease their botnets to other hackers in the dark-net.

F. Belqasmi et al. (Eds.): AFRICATEK 2017, LNICST 206, pp. 181–191, 2018.
https://doi.org/10.1007/978-3-319-67837-5_17

In response to such malicious activities, various companies have taken the responsibility to detect and stop any botnet reporting to command and control servers. The key to achieving this role lies in the development of an efficient multi-factor botnet detection and alarm mechanism.

In the next section of the paper, we give some background information about the DNS system as well as botnets. This is followed by the methodology we followed to develop a cyber-threat alarm system. In Sects. 4 and 5, the results obtained are analyzed and our conclusions are drawn.

2 Botnets and the Domain Name System

2.1 What are Botnets?

A botnet is defined as a group of computers connected to the Internet, which are controlled by a hacker without the awareness of their users/owners [1]. For a machine to be controlled by a hacker, it must be first turned into a zombie. This typically occurs through an Internet port that was left open and was used by the hacker to plant a Trojan horse or a malicious code with a backdoor on the machine – this backdoor can be used for later attacks. Whenever needed, the zombie botnet can be used to obey a command sent by the hacker. The hacker can use botnets to simultaneously send a very large number of bogus requests to a specific server causing it to crash [1].

2.2 How do Botnets Work?

Typically, botnets wander the Internet looking for exposed computers to quickly infect them and remain discrete waiting for the right time to perform a task given to them by their master. Tasks performed by the botnet can be classified into four categories (Table 1):

Table 1. Types of Botnet Activities

Sending	Botnets are used to send spams, viruses, and malware to different systems through the Internet
Stealing	Botnets are used to steal sensitive information from the infected computers - information such as credit card numbers, passwords
DoS attack	Botnets are used to perform a denial of service attack through redirecting transmissions to a specific server in the effort of crashing that server and blackmailing the owners
Click fraud	Botnets can be used to click on Internet ads to boost web advertisement

- **Internet Relay Chat (IRC) signals:** This concentration on IRC ports by the bot masters guided some information security specialist to block all IRC communications when setting up a business network environment. This has led bot masters to search for new ways of communicating with their bots, such as the following:

- **JPG files:** A more advanced way a bot herder can communicate with his bots is through the metadata in JPG file. Because those files are transmitted through HTTP port 80, most computers will allow them.
- **Microsoft Word 2007 files:** Microsoft Word 2007 files contain XML metadata and by using this metadata, the bot master can send commands to the bots, which will not raise any suspicion as the traffic is being passed through port 80.
- **LinkedIn.com Status:** bots can be programmed to use the LinkedIn API to receive commands by periodically checking the status message of a dummy account.

2.3 Botnets and DNS

It is obvious that the command and control server must be able communicate with its zombies. Thus, the perfect way in which a bot master can communicate with its zombies is a way that can assure that the communication remains undetected and discrete. Due to this, bot masters tend to use DNS as a communication channel to send the commands needed to carry out a specific malicious task, because of the following reasons: First, the fact that there is no effective mechanism to differentiate between the legitimate DNS queries from the malicious ones. Moreover, DNS as a protocol is left untouched in terms of firewalling and securing a system in most environments. Another advantage is the ability to change DNS records frequently, as DNS was built initially as a distributed system that assures resilience [2]. Fast Flux Networks (FFN) are a subset of botnets that changes IP and domain name association frequently to pose as Content Distribution networks (CDNs) in an effort to avoid detection [3]. This helps hackers in case the C&C IP address was blocked, as changing DNS record will allow botnets to continue communicating with the bot master.

3 Literature Review

According to Hao, Feamster and Pandrangi, malicious domains can be detected using certain parameters such as number of queries performed after the domain registration, the fraction of IP addresses associated with those domains, and the ACs containing the domains' records [4]. However, those conclusions were reached based one month of monitoring using locally installed probes – a technique which presented temporal and scope related limitations.

Based on Bilge's work presented in [5], Exposure (a system introduced in the paper) was scalable enough for detecting malicious domains using passive DNS analysis. The system was unique because of the 15 behavioral features it uses to detect malicious domains. The limitation associated with the paper, is that an attacker can avoid detection by Exposure if he studied the features it looks for and tries to avoid them. Another limitation of Exposure that was highlighted by Antonakakis et al. [6], is that it relies on monitoring traffic that is initiated from some local recursive DNS servers.

Konte et al. [7] focused on the monitoring of URLs associated with scan campaigns, in order to better understand the behavior of fast flux networks as their associated rapid changes in DNS mappings. Nevertheless, this work was concerned with

the study of scam websites, not addressing the issue of malware detection. Spring [8] presented an anti-phishing black listing can contribute to limiting the lifetime of a phishing website. Moreover, using specific DNS information may help in the automated detection method of fast flux networks.

Choi et al. [9] proposed to monitor DNS traffic in order to detect a group activity of resolving a domain by sending simultaneously DNS queries where this indicate a distributed botnets trying to resolve a bots' master domain. This approach was found to be effective for the detection of botnets. Its main limitation remains its large processing time. Furthermore, botnets can evade this algorithm if they use DNS only in the initialization stage. Finally, botnets can paralyze this algorithm by intentionally generating DNS queries that spoof their source addresses.

Another work that addresses botnets and C&C detection by monitoring the time period between the domain registration and its first DNS activity is presented by Spring et al. [10]. In this work, the authors propose a pattern for botnet detection in which legitimate domain activity will not take a long time to start DNS activity. The main limitation in this case is that this solution relies on passive DNS sources.

While all the mentioned algorithms tackled DNS activity at the low level of the hierarchy, Kopis [6] is more interested in the upper DNS hierarch, which ensures global visibility. The main advantages of Kopis are its use of real data for a period of six months, in addition to the ability to detect newly created and previously unclassified malware domains several weeks before their appearance in any blacklist.

Another botnet detection method consists in detecting illegal fast fluxing that ensures for a bot master a reliable hosting with high availability. In the paper presented by Holz et al. [11], an automated mechanism for the detection of new fast fluxing domains is proposed. Although the proposed approach yields low false positive and false negative rates, the algorithm needs enhancement to be more reliable and detect complex botnet communities. Freiling et al. [12] proposed approaches to prevent botnet attacks by observing the communication flow within the botnet and detecting the IP address that it resolved. Another approach is by terminating the infrastructure hosting the C&C server, by manipulating DNS replies.

4 Research Methodology

In this work, we propose a multi-factor cyber-threat detection system that relies on DNS traffic analysis for the detection of malicious domains. In order to achieve this goal, a simulated network was built. In this network, a computer acts as a bot trying to communicate with a specific Command and Control (C&C) server through DNS queries. The DNS queries are passed to a specific DNS server, which we have configured. An IDS implementing a DNS multi-factor detection mechanism is placed in the network to enable the differentiation between legitimate domains and malicious ones. In addition to the DNS traffic retrieved from our own DNS server, we also relied on AUE DNS records obtained for the last month from Etisalat (the main ISP in the UAE). The retrieved DNS records are analyzed to explore botnet activity in the UAE cyber space.

4.1 The Simulated Network Components

As depicted in Fig. 1, the main components of the built simulated network consist of the following:

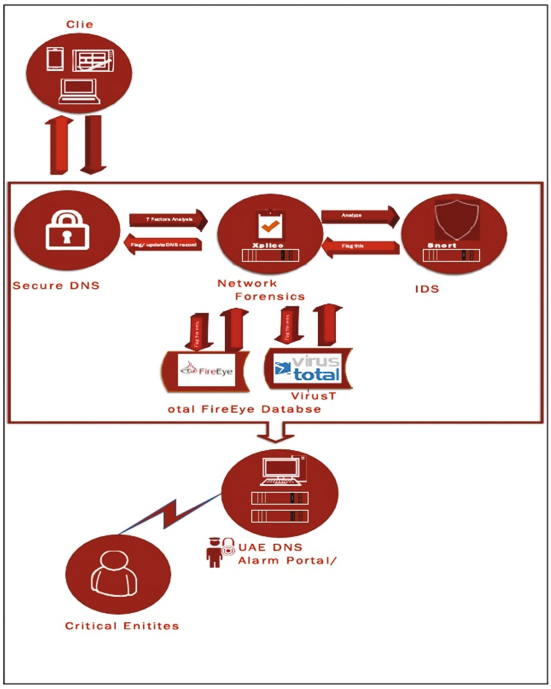

Fig. 1. Simulated network components

1. **Domain Name System (DNS) Server**: To build our DNS server, we choose the bind9 open source tool because it has the capability to operate on multiple platforms, nevertheless, adding the feature of forwarding the DNS queries to a real DNS server. Another reason why bind9 is being chosen is the friendly interface associated with the tool.
2. **Infected Machine:** A script to manipulate the cron tab will be planted in the infected machine, in which it will schedule a malicious communication with our C&C server. This way, we will ensure that malicious activities will take place in our environment, using which we can test the effectiveness of our intrusion detection system [13].
3. **Cyber Threat Alarm System:** Different applications and network components were optimized to ensure effective collaboration malicious domains detection. Figure 2 illustrates the applications used to build our cyber threat alarm system. Some applications may have features, which other applications do not. For instance, Snort lacks the ability of showing the geo location while Xplico, a network forensics analysis tool, provides this information [14]. Snort will be placed online,

thus all traffic will be passed through it. This ensures that all traffic will be examined and matched to our pre-defined rules [15], which would enable the flagging of any suspicious traffic. On the other hand, Xplico will provide us with additional analysis of the traffic where it will reflect a live acquisition of the network. This tool will automate the process of analyzing the traffic throughout the network, which will reduce the time needed to inspect suspicious queries. Additionally, the findings, which are extracted from sniffing the monitored network traffic can be compared to a list of well-known C&C servers, well-known malicious ports, and owners of other malicious websites. This list will be built by relying on trusted third party databases, such as FireEye and others.

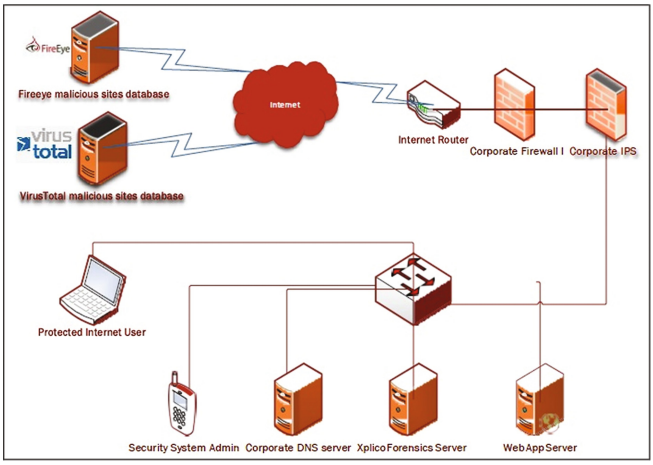

Fig. 2. Cyber threat alarm system components

To achieve effective results, a multi-factor malicious domain detection mechanism was developed and integrated with the IDS, to raise alarms in case of malicious activity detection. The developed mechanism relies on the seven factors listed in Table 2 for the detection of malicious domains.

Table 2. Malicious domains detection factors

Factors	Definition
Reputation	Domain reputation can indicate the suspiciousness of the domain as it reported from different entities/antivirus/security research center
Geo location	Geo location can be used as a factor on detecting suspicious traffic based on risk rating of the most countries hosting/generating such traffic (attacks/)
Destination port	Many suspicious traffic can be detected based on a well-known suspicious port it use as a destination (since it communicate to the bot master) adding to this, some ports which is not included in the list but is from the range of not allocated ports (1-1024) can also show a suspicious traffic

(*continued*)

Table 2. (*continued*)

Factors	Definition
Known C&C	Some leading organization are publishing any new botnets with their associated C&C. in this project we will refer to one of these organization (FireEye)
Domain owner	In a domain was owned by the same person who owned a well-known C&C, this also indicate that the new domain is most probably a suspicious one
Frequent DNS changes	Changing domain information/record should not be a frequent thing. Some organization (domain tools and others) keep record of these changes and raise an alert in case changes were very frequent
Behavior	In case that a client resolve a suspicious domain and then establish a communication with it followed by a misbehaving traffic (such as generating a DoS attack) then it raise a concern of being infected

All the factors mentioned are assigned weights and calculated as a weighted sum, which if it reached a certain threshold would trigger an alarm at the IDS level. Weight categories were distributed in a way to cover all possible scenarios that might happen and require the activity to be flagged. The weighted sum of the different detection factors is expressed as follows:

$$S = 1-(0.4(r+g+b+f))+k+o+p \quad (1)$$

If $S \leq 0$, flag it as Suspicious DNS record
If $S \leq 0$, flag it as good DNS record
S: DNS flag; r: Site DNS reputation.
g: DNS Geolocation; b: Suspicious traffic behavior.
f: frequent DNS changes; k: Known botnet command & control center
o: Blacklisted domain owners; p: Known blacklisted port

Our factors are divided into two categories; the first category contains the must-stop factors while the other category contains the partial-stop factors. The point where an alarm flag must be raised is when either one of the must-stop factors is met or if three of the partial-stop factors are met. Having the three partial-stop factors meet doesn't guarantee that the domain being flagged is a suspicious domain, however flag this domain for further inspection can eliminate the risk of a botnet being deployed in the network. The weight categories associated with each factor along with the reason of assigning that much weight is shown in Table 3 below.

Table 3. IDS factors to completely or partially block investigated DNS name

Factor	IDS Decision	Reason
Known C&C servers	Must stop	Already being flagged as a known C&C by a trusted third party leaves no doubt that this factor is a must-stop factor
Known C&C ports	Must stop	Communicating through an already known port as a preferred random port by C&C is a must-stop activity

(*continued*)

Table 3. (*continued*)

Factor	IDS Decision	Reason
Known Owners	Must stop	The name of known C&C servers' owners when associated with a new domain raise a must- stop flag
Reputation	Partial stop	Some domains may have a bad reputation although they are legitimate however others are surely malicious. The key phrase here is "no smoke without no fire"
Behavior	Partial stop	When a non-ordinary traffic behavior is experienced toward a domain a partial-stop flag should be raised
Frequent- change in DNS entries	Partial stop	A frequent change in DNS entries is a popular action done by bot masters, however other legitimate domains do that to achieve redundancy that leaves us with a partial-stop flag
Geo- location	Partial stop	Some geolocation unfortunately is well known for malicious activity, however that doesn't mean that anything coming from this location is malicious, but in the other sided it deserves a partial-stop category flag

5 Data Collection and Data Analysis

In this project, we will use primary data source in order to get accurate results. Our primary data will be the internal network is another primary source for data. These data will be analyzed as explained in the methods where this project will consider a domain as a malicious domain based on the final score that is assigned to it. The score is calculated based on several criteria such as the domain registrar information and when it was register, the review of this domain in the online reputation service as well as the IP reputation that is assigned to that domain, the behavior of the traffic generated toward this domain and the port is used for such communication. If most of these criteria were flagged, then this domain is suspicious. To test and check how accurate our solution is we compare the results with some ATP solutions such as FireEye. After verifying the accuracy of our solution, we can then detect C&C in real time.

Tables 4 and 5 depict the test results obtained, as well as the analysis of those results.

Table 4. Obtained Results

	Real threat	*Algorithm decision*
ftp.idm.ae	No Threat	Blocked
office.ontimedata ~ soluitions.com	Suspicious DNS	Blocked
zu.ac.ae	No Threat	Allowed
d99q.cn	Suspicious DNS	Blocked
datatoad.iptime.org	Suspicious DNS	Blocked

(*continued*)

Table 4. (*continued*)

	Real threat	*Algorithm decision*
doubleclick.net	Suspicious DNS	Blocked
fbcdn.net	Suspicious DNS	Blocked
gstatic.com	Suspicious DNS	Allowed
aptuslearning.com	Suspicious DNS	Blocked
Lucydriver ~ translations.com	Suspicious DNS	Blocked
rgmechanics.ru	Suspicious DNS	Blocked
eri.edu.pk	Suspicious DNS	Blocked
icet-logistics.ro	Suspicious DNS	Blocked
samdriver.com	Suspicious DNS	Blocked
www.kareenas.com	Suspicious DNS	Allowed
www.elderology.net	Suspicious DNS	Blocked
abrico.info	Suspicious DNS	Blocked
powervoice-2.tk	Suspicious DNS	Blocked
esportskart.com	Suspicious DNS	Blocked
www.lagunasderuidera.net	Suspicious DNS	Blocked
emazkid.ghettohost.tk	Suspicious DNS	Blocked
hank-moody2.tk	Suspicious DNS	Blocked
www.motorfliegen.ch	Suspicious DNS	Blocked
southwest.icims.com	Suspicious DNS	Blocked
www.stylenstitch.com	Suspicious DNS	Allowed
www.tamilkamadesam.in	Suspicious DNS	Blocked
google.bi	Suspicious DNS	Allowed

Table 5. Results' analysis

		Suspicious DNS records	
		True (23 records)	*False (two records)*
Algorithm Decision	*Positive (22 records)*	21	1
	Negative (5 records)	4	1

The malicious domains' detection accuracy % can be represented by Eq. (2) below:

$$a = 1 - \frac{tp + fn - fp - tn}{t} \cdot 100\% \tag{2}$$

a: detection accuracy; tp: True positive decisions.
fn: False negative decisions; fp: False positive decisions.
tn: True negative decisions

Based on the obtained results, we conclude that our system yields the following accuracy:

$$a = 1 - \frac{21 + 1 - 1 - 4}{27} \cdot 100\% = 62.9\%$$

6 Conclusions and Future Work

In this work, we presented a multi-factor cyber-threat detection system that relies on DNS traffic analysis for the detection of malicious domains. The conducted experiments show that our system yields a malicious domains' detection accuracy rate of 62.9%. This performance in terms of accuracy level can be improved by considering more factors for the detection. One of the additional factors that can be considered is the domain owner – a factor that may lead to the detection of malicious domains in advance and in some cases before that domain starts its malicious activities. This value comes with an overhead as it requires the tracking of not only the malicious domains but also their owners, in addition to the monitoring of owners to determine if they registered any new or existing domains, and if these domains actively change. Domain monitoring tools such as a domain service provider can assist in detecting dynamic changes but for the premier users and this is offered as well in their APIs. This will help detecting the fast fluxing in malicious domains as they depend on frequent changing in their records. On the other hand, the IDS rules were implemented in order to detect any suspicious domains by comparing all DNS content with the list of well-known C&C domains. This list can be obtained from FireEye as they publish all the newly detected ones. Adding to this, detecting known bad ports as well as any misbehaving traffic are implemented in IDS rules, which alert the network admin of the existence of a botnet in the network.

Acknowledgements. This work was supported by Zayed University Research Office, Research Cluster Award # R17079.

References

1. Dietrich, C.J., Rossow, C., Freiling, F.C., Bos, H., Steen, M.V., Pohlmann, N.: On Botnets that use DNS for command and control. In: 2011 Seventh European Conference on Computer Network Defense (EC2ND), Gothenburg (2011)
2. Botnet (zombie army) definition. http://searchsecuritytechtarget.com/definition/botnet (2012)
3. Al-Duwairi, B., Al-Hammouri, A., Aldwairi, M., Paxson, V.: GFlux: a Google-based system for fast flux detection. In: IEEE Conference on Communications and Network Security (IEEE-CNS 2015), Florence, Italy, 27–29 Sept 2015 (2015)
4. Hao, S., Feamster, N., Pandrangi, R.: Monitoring the initial DNS behavior of malicious domains. In: Proceedings of ACM SIGCOMM Conference on Internet Measurement Conference, New York, NY, USA (2011)

5. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: EXPOSURE: finding malicious domains using passive DNS analysis. In: Proceedings of 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2011
6. Antonakakis, M., Perdisci, R., Lee, W., II, N.V., Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: Proceedings of 20th USENIX Security Symposium, San Francisco, CA, August 2011
7. Konte, M., Feamster, N., Jung, J.: Dynamics of online scam hosting infrastructure. In: Proceedings of Passive and Active Measurement (PAM), Seoul, South Korea, April 2009
8. Spring, J.M.: Large Scale DNS Traffic Analysis of Malicious Internet Activity with a Focus on Evaluating the Response Time of Blocking Phishing Sites, Master's Thesis, School of Information Science, University of Pittsburgh, Pittsburgh, PA, p. 26 (2010)
9. Choi, H., Lee, H., Kim, H.: Botnet detection by monitoring group activities in DNS traffic. In: 7th IEEE International Conference on Computer and Information Technologies (2007)
10. Spring, J.M., Metcalf, L.B., Stoner, E.: Correlating domain registrations and DNS first activity in general and for malware. In: Proceedings of Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom, April 2011
11. Holz, T., Gorecki, C., Rieck, K., Freiling, F.C.: Measuring and detecting fast-flux service networks. In: NDSS (2008)
12. Freiling, F., Holz, T., Wicherski, G.: Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks. In: 10th European Symposium on Research in Computer Security (2005)
13. Aldwairi, M., Khamayseh, Y., Al-Masri, M.: Application of artificial bee colony for intrusion detection systems. Secur. Commun. Netw. **8**(16), 2730–2740 (2015)
14. Kharbutli, M., Aldwairi, M., Mughrabi, A.: Function and data parallelization of Wu-manber pattern matching for intrusion detection systems. Netw. Protocols Algorithms **4**(3), 46–61 (2012)
15. Aldwairi, M., Alansari, D.: Exscind: Fast pattern matching for intrusion detection using exclusion and inclusion filters. In: Proceedings of the 2011 7th International Conference on Next Generation Web Services Practices, pp. 24–30 (2011)