



A Survey of Fast Flux Botnet Detection With Fast Flux Cloud Computing

Ahmad Al-Nawasrah, Taibah University, Saudi Arabia


Ammar Ali Almomani, Al-Balqa Applied University, Jordan

 <https://orcid.org/0000-0002-8808-6114>

Samer Atawneh, College of Computing and Informatics, Saudi Electronic University, Saudi Arabia

 <https://orcid.org/0000-0001-7590-7887>

Mohammad Alauthman, Department of Computer Science, Faculty of Information Technology, Zarqa University, Jordan

 <https://orcid.org/0000-0003-0319-1968>

ABSTRACT

A botnet refers to a set of compromised machines controlled distantly by an attacker. Botnets are considered the basis of numerous security threats around the world. Command and control (C&C) servers are the backbone of botnet communications, in which bots send a report to the botmaster, and the latter sends attack orders to those bots. Botnets are also categorized according to their C&C protocols, such as internet relay chat (IRC) and peer-to-peer (P2P) botnets. A domain name system (DNS) method known as fast-flux is used by bot herders to cover malicious botnet activities and increase the lifetime of malicious servers by quickly changing the IP addresses of the domain names over time. Several methods have been suggested to detect fast-flux domains. However, these methods achieve low detection accuracy, especially for zero-day domains. They also entail a significantly long detection time and consume high memory storage. In this survey, we present an overview of the various techniques used to detect fast-flux domains according to solution scopes, namely, host-based, router-based, DNS-based, and cloud computing techniques. This survey provides an understanding of the problem, its current solution space, and the future research directions expected.

KEYWORDS

Botnet Detection, Cloud Computing, DNS, Fast-Flux, Neural Network, Zero-Day Domain

1. INTRODUCTION

Networks of compromised pcs remotely controlled by attackers are the foundation of various cyber threats to cloud environments, including distributed denial-of-service (DDoS) attacks, identity theft, phishing and spam (Al-Fayoumi et al., 2019; Alauthaman et al., 2018; Alauthman et al., 2020; Almomani et al., 2018; Almomani et al., 2015; Almomani, Wan, et al., 2013; Alomari et al., 2016; Alomari et al., 2014; Barford et al., 2007; Dagon et al., 2008; Fabian et al., 2007; Grizzard et al., 2007; Gu et al., 2008; B. Gupta, 2011; Karasaridis et al., 2007; Levy et al., 2005; Rajab et al., 2006).

Fast flux networks (FFNs) are a unique form of a botnet that criminals use to give high availability and flexibility for their malicious websites, in the same way as roundrobin domain names (RRDNS) and content delivery networks (CDNs) (Alieyan et al., 2015). Botnet writers disguise their malicious

DOI: 10.4018/IJCAC.2020070102

activities and design new tactics and mechanisms to hide their communications. One method is the IP fast-flux, which is a mechanism that frequently changes IP addresses corresponding to a unique domain name. Another method is the domain flux, which is a mechanism that automatically and periodically generates domain names related to a URL of a C&C server (Alieyan et al., 2019; Zou et al., 2018). FFNs ' key concept is to use bot pcs as proxies (flux agents) to forward user queries to backend servers called "motherships".

These rapid changes in proxy IP addresses is crucial to avoid detection and prospective shutdown and guarantee high availability for that backend servers. FFNs are regarded as a fresh growth in spam campaign operation and leadership. In addition to campaigns, spammers send thousands of emails containing interesting product or service advertisements (e.g., pharmaceutical, adult content, and phishing) to users' email inboxes (Al-Duwairi et al., 2014). These advertisements generally contain hyperlinks of malicious websites for the campaigns. Until recently, only a single static IP address is related to a website for a certain period; such characteristic provides security defenders with the chance to take down that website. According to FFNs, the domain name of a malicious website points to more than one IP address (FF-agents), which is frequently and rapidly changing.

According to (Kalige et al., 2012), HTTP botnets are considered dangerous because they attack and exploit systems. Current HTTP botnets use the strongest techniques to perform attacks. An example is the Asprox botnet, which has affected about 3.5 billion computers in the United States. The Asprox botnet uses an advanced double fast-flux, called the hydra fast-flux, as its main technique (Al-Bataineh et al., 2012). This technique renders the efforts to take down C&C serves useless. Additional details are presented in Subsection 2.3.

The Cost of CyberCrime study Study (Enterprise, 2015) points out that notes that 252 benchmarked organisations have an annualized average cost of \$7.7 million a year. The study also demonstrates that either a botnet or a web-based attack performs or supports these exploits, and fast-flux is used as an avoidance method to provide accessibility and resilience.

The report mentions that the most dangerous cyber-crimes are those caused by denial of services (DoS) and web-based attacks. The fast-flux evasion technique has been widely used in botnets and web-based botnets to carry out DoS and other attacks (e.g., phishing and spam), with fast-flux serving as the backbone C&C communication between the compromised computers and the mothership/ malicious website.

Cyber-criminals have stolen around \$78 million through various means using financial malware (Dave Marcus, 2012). Also, McAfee stated that previous fraud cases in Eastern Europe could be attributed to Zeus and SpyEye activities; after tracing some of these attacks, they found a highly complex fast-flux botnet, as well as hidden compromised servers supporting the website's long life (Dave Marcus, 2012). Botnets are also responsible for spam e-mails. Spammers earn an annual average income ranging from \$50,000 to \$100,000 (Su et al., 2012). Fake online pharmacies are one of the many illegal activities available on the Internet; such activities are notorious for selling fake or inefficient medications and are involved in identity theft cases. A report from the Fortinet Global Cyber Security Research Team states that the fast-flux technique has been used in fake Canadian online pharmacies to avoid detection (Yadav et al., 2010). Security researchers have recently reported that a new variation of the "Gameover Zeus" botnet makes use of the fast-flux technique to protect its C&C servers (Micro, 2014).

The so-called unidentified Zero-day fast-flux domain is one of the key issues in botnet detection. The zero-day domain is described as those that are not blacklisted bots (FF-agents) (Lin et al., 2013). A fast-flux attack is a complex evasive technique that many current techniques cannot identify, as attackers can use fresh bots that have not earlier been seen. Several prospective alternatives have been suggested for fast-flux botnet assaults, but these alternatives are not yet efficient. They range from passive, active and real-time alternatives. The malicious and legitimate domain misclassification expands over time, particularly when interacting with unidentified fast-flux botnet domains (zero-day domains). The remaining survey is organized in the following order. Section 2 provides an overview and examination of fast-flux botnets. Section 3 highlights the motivations for the study.

2. BACKGROUND AND OVERVIEW OF THE FAST-FLUX BOTNETS

Numerous websites provide commercial services to users. The efficiency of these services is highly dependent on their availability. Server systems are distributed to large redundant service networks in multiple areas to achieve high availability (Scharrenberg, 2008). The DNS is a hierarchical distributed naming system for computers and resources that are connected to the Internet (Shaikh et al., 2001). A browser usually automatically acquires the IP address of the desired hostname to access a website. The DNS server typically returns the same reply each time. Thus, the same IP address is returned each time a hostname is requested. Some requests, such as Round Robin Domain Name Server (RRDNS), CDNs, and FFSNs, do not work in the same manner as previously described. RRDNSs, CDNs, and FFSNs share similar characteristics, such as low time to live (TTL). RRDNSs and CDNs are DNS-based methods for load balancing that provide a high degree of performance, availability, and scalability for content websites. RRDNSs distribute user requests to their distributed servers by swapping the IP addresses of the DNS response of the same domain each time to provide load balancing. CDNs represent a network of globally distributed nodes to return the IP address of the nearest accessible node to the client; they thus support service speed and availability. Similarly, fast-flux uses a similar concept of frequently changing IP addresses that correspond to a specific domain. This strategy helps cyber-criminals to remain undetected. The main difference between FFSNs and CDNs is that CDN nodes are fully administered machines, whereas FFSNs are malware-infected computers (Lin et al., 2013).

The business side of fast-flux hosting begins with malware authors. By developing phishing kits, this software package can be used to deliver phishing emails to a set (list) of victims and host an illegal website to which those emails are directed. Others sell lists of addresses for spam purposes, whereas others improve bot software. A flexible, remotely controllable software known as bot software enables subsequent downloads on a particular computer once it has been installed on a victim's computer. E-mail-borne worms are used by bot herders to infect and exploit thousands of computers. Such tools are the most appreciated these days by malware authors and cyber-criminals. Malware authors and bot herders are rich sources of the cyber-criminal community ((SSAC), 2008).

FFNs provide high availability and reliability to scam websites (Konings, 2009). The ICCAN report ((SSAC), 2008) defines a fast-flux technique as one in which multiple IP addresses (sometimes hundreds or even thousands) are assigned and re-assigned to a single fully qualified domain name (FQDN), such as www.example.com. The URLs and domain names for the announced content are not resolved to any IP addresses of backend servers. Instead of pointing to back-end servers, the URLs and domain names addresses are changed among many front-end agents, which serve as redirectors; thus, the content is forwarded to the back-end servers (the mothership) (SSAC, 2008; Gasster, 2008; Konings, 2009).

Fast-flux mainly involves two techniques, namely, the IP fast-flux and the domain flux. The IP fast-flux comes in two types (Figure 1): the single fast-flux and the double fast-flux. An extension type of the double flux is called the hydra flux (Subsection 2.3). The details of these techniques are discussed in the subsections that follow.

2.1 Single Fast-Flux

Domain names are registered in an official registrar by an attacker for use in illegal activities by an official registrar. The attacker registers a domain name for an FFSN referring to illegal websites (e.g., bad.com) and another domain name (Resolvenameserver.com) to serve the mapping domain name resolution services. As mentioned previously, the attacker adds IP addresses to the bulletproof server and then provides the control of the FFSN to a mothership.

In a single fast-flux, the attacker deploys a bulletproof server to host the zone file. The bulletproof web hosting server leads customers to the desired malicious website. Such services are well-known among botnet owners, who need a reliable environment and assist in deploying a botnet C&C server.

Figure 1. Comparison of IP resolutions of fast-flux techniques

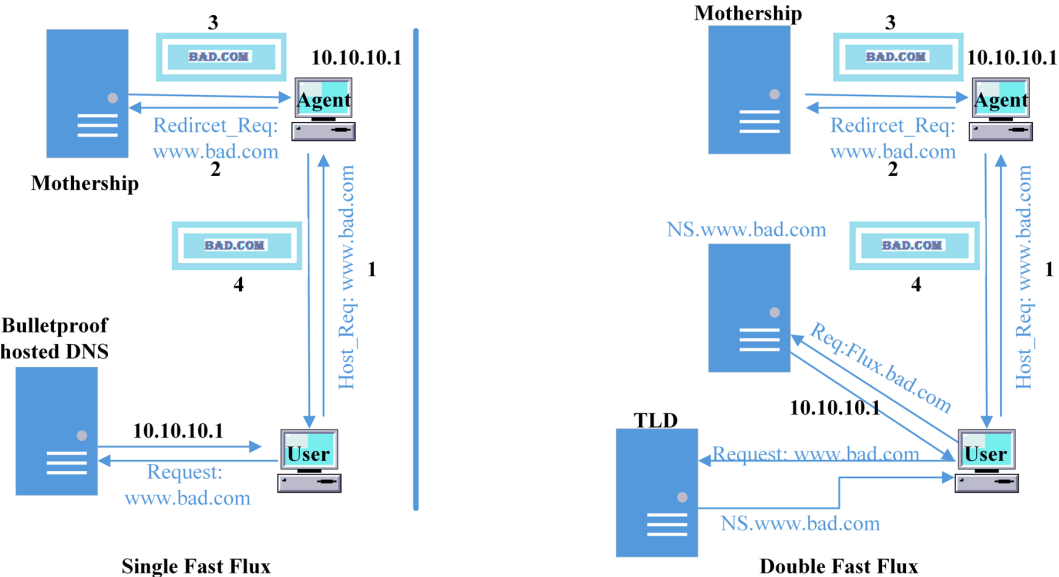
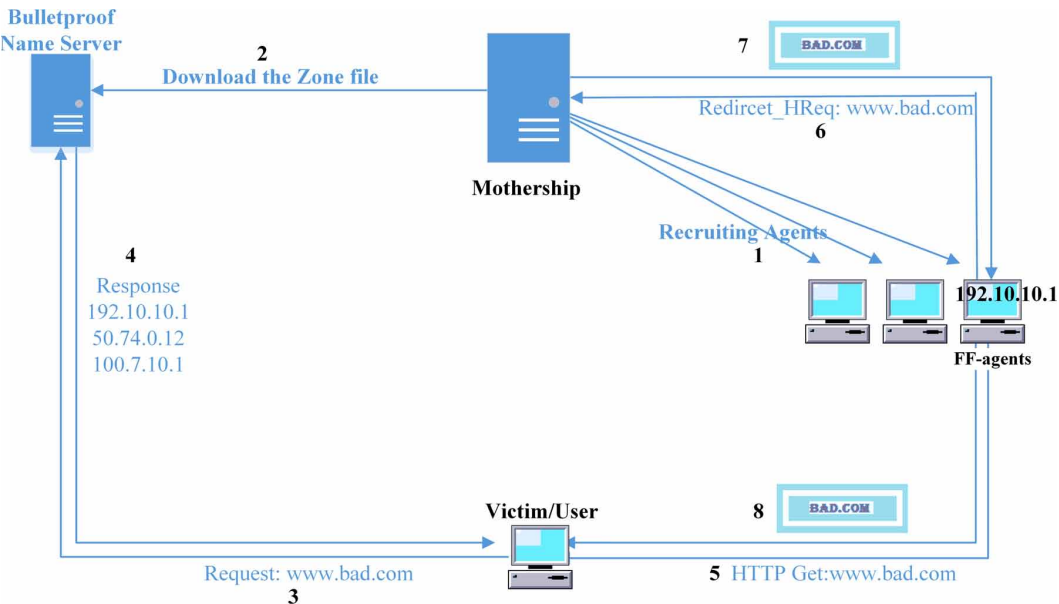


Figure 2 shows the process of a single fast-flux of the IP addresses of a malicious website.

Figure 2. Single fast-flux of IP addresses of a malicious website



1. The attacker recruits some of the compromised computers to work as proxies, which directly redirect user requests to the mothership/operator.

2. The attacker adds the name server (Resolvenameserver.com) records of the malicious website (www.bad.com/mothership) to the zone file via the registrar.
3. The victim (user) requests the FQDN (www.bad.com). Hence, a request is sent to the DNS looking to resolve the FQDN. Assuming the absence of caching, a recursive DNS server asks for the authoritative name server for this FQDN. The part of the recursive process from the top-level domain (TLD) to the authoritative server is omitted.
4. Instead of sending the IP address of the FQDN (www.bad.com), the authoritative name server sends back a list of the IP addresses of the proxies to the user.
5. The user initiates a GET message to one of the IP addresses in the list.
6. The FF-agent (proxy) redirects the message to the malicious webserver (the mothership) to handle the message.
7. The malicious web server sends the response (answer) back to the FF-proxy.
8. The FF-agent returns the response to the user.

The A records of the web servers are constructed with short TTLs (Holz et al., 2008). The FFSN operators directly provide a new set of A records to replace the old set of records (of the FF-agents) when the TTLs of the request expired. Thus, there is very little chance of identifying and shutting down the web servers, which are supported by this FF technique. The records associated with the illegal website in the zone file of the DNS bot (Resolvenameserver.com) might appear as follows:

```
bad.com. 180 IN A 192.10.10.1  
bad.com. 180 IN A 50.74.0.12  
bad.com. 180 IN A 100.7.10.1
```

The TTL for each RR is very low (180 s). The RRs are directly replaced with new bot (FF-agents) IP addresses when the TTL expires. The zone file might be read as follows after a time of TTL+1:

```
bad.com. 180 IN A 155.1.1.14  
bad.com. 180 IN A 180.88.0.9  
bad.com. 180 IN A 120.1.1.2
```

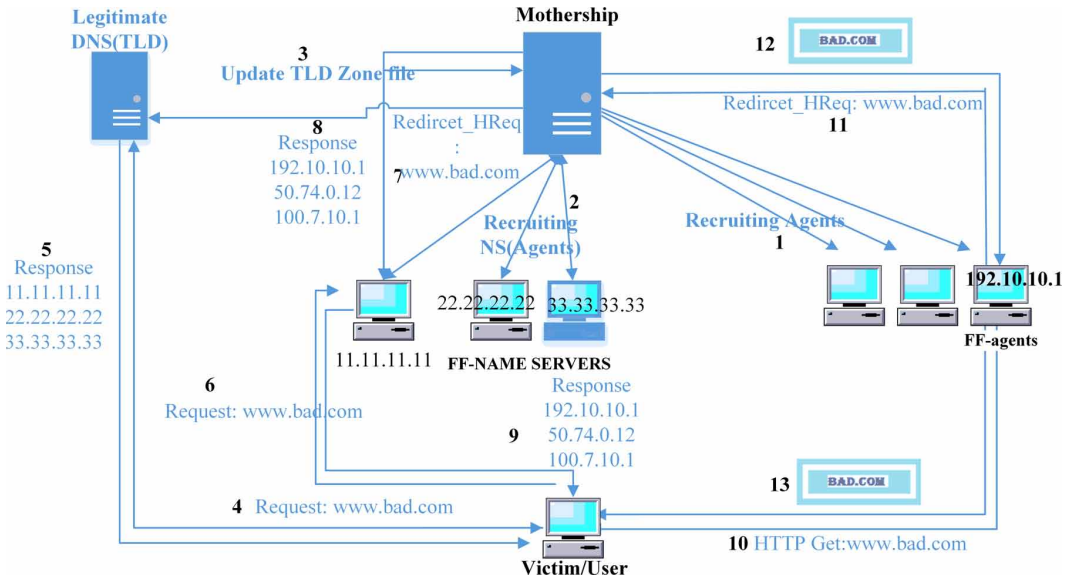
2.2 Double Fast-Flux

Furthermore, the fast-flux mothership/operator identifies the abovementioned domains, which correspond to its FFSN (Figure 3). The FF-agents in the two FFSNs are separated to simplify the understanding of the idea behind the double fast-flux because FF-agents are commonly used to serve both DNS and HTTP requests at the same time (Xu et al., 2013) as the mothership/operator.

Figure 3 summarizes the double fast-flux process of the IP addresses of the malicious website and the authoritative name server.

1. The attacker recruits some of the compromised computers to work as proxies, which directly redirect the user request to the FF mothership/operator.
2. The attacker recruits some of the compromised computers to work as NS proxies, which directly redirect the DNS request to the mothership/operator.
3. The attacker adds the name server records (Resolvenameserver.com) to the TLD zone file via the registrar and keeps updating the legitimate DNS RR of the authoritative name servers of the malicious domain.
4. The victim (user) sends a request of (www.bad.com) to the DNS server to resolve the FQDN.
5. The DNS returns a list of authoritative name servers for this FQDN, which are a part of the maliciously compromised pool of NS agents.
6. The user sends the authoritative NS asking for the IP address of the FQDN.
7. The authoritative name server forwards the DNS request to the mothership instead of resolving and directly returning the IP address of the FQDN.

Figure 3. Double FFSN of the name server and IP addresses of the malicious website



8. The mothership returns a list of IP addresses that are FF-agent proxies of the website server (mothership).
9. The authoritative name server sends the IP addresses back to the user.
10. The user initiates a GET message to one of the IP addresses in the list (which is one of the FF-agents).
11. The FF-agent (proxy) redirects the message to the malicious web server (mothership) to handle the message.
12. The malicious web server sends the response back to the FF-agent.
13. The FF-agent returns the response to the user.

The attacker continuously updates the NS records of the TLD. Through the registrar, the domain owner has the ability to modify the domain information. The attacker frequently changes the IP addresses of the NS servers to point to different hosts and sets the TTL value for these NS servers to a very small value (e.g., 180 s). The RRs of the NS might be shown in a TLD zone file as follows:

```
bad.com. NS NS1.Resolvernameserver.com
bad.com. NS NS2.Resolvernameserver.com
NS1.Resolvernameserver.com A 11.11.11.11
NS2.Resolvernameserver.com A 10.0.0.2
```

The attacker automatically replaces the A records of the NS when the TTL expires. Therefore, the RRs of the NS might be shown in a TLD zone file as follows:

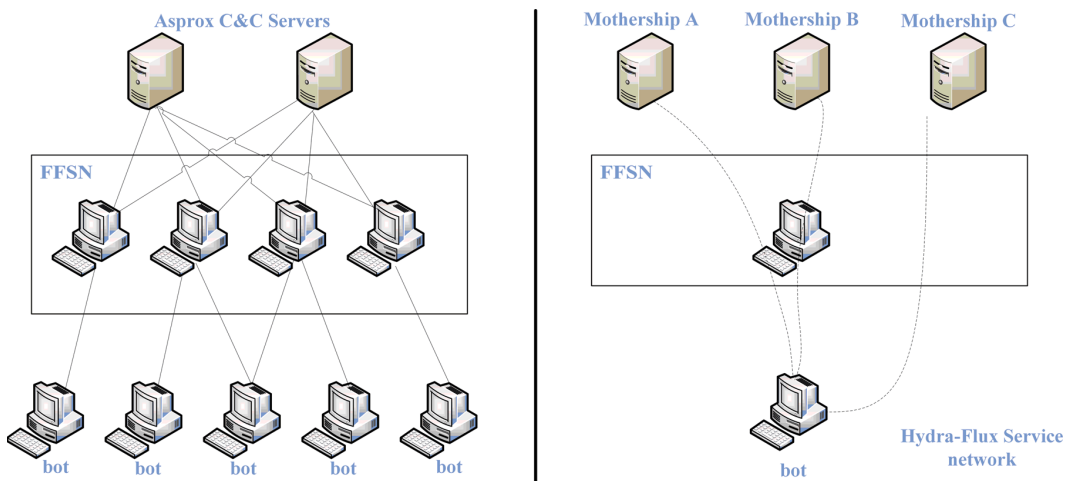
```
bad.com. NS NS1.Resolvernameserver.com
bad.com. NS NS2.Resolvernameserver.com
NS1.Resolvernameserver.com. A 22.22.22.22
NS2.Resolvernameserver.com. A 10.10.10.233
```

Consequently, there is very little opportunity to detect and shut down the name servers that support this fast-flux attack. Combining the two FFSNs is an effective method for keeping the website alive for longer periods than websites that do not use the same techniques.

2.3 Hydra Fast-Flux

The new advanced FFSN does the same as the traditional FFSN, but taking it down is impossible. Similar to that of the traditional FFSN, the mothership of the new advanced FFSN can be deactivated by law enforcement, but the bots have an alternative IP address to another mothership related to the same FFSN. As in the Asprox botnet, the bots download a list of available motherships. Ultimately, alternative IP addresses add multilayer of double fast-flux to the botnet and maintains extra availability to the malicious content. Figure 4 depicts the multilayer FFSN of the Asprox botnet, which is usually denoted as a hydra-flux service network.

Figure 4. Multilayer FFSN of the Asprox botnet and hydra-flux service network



2.4 Cloud Domain Flux

Another type of fluxing technique is cloud domain flux. In contrast to the fast-flux of the IP addresses related to a cloud domain name, the cloud domain flux is the process of fluxing domain names related to a cloud URL of the cloud C&C server.

The cloud domain flux is used by cloud bots to contact the cloud C&C server. The domain generating algorithm generates the same domain names for both the cloud C&C server and its cloud bots when seeded with the same value. The cloud C&C server is used to register some of the auto-generated domains. (Stone-Gross et al., 2009) revealed that the Torpig botnet calculates domain names by combining the current week and year and adding the TLD (e.g., “weekyear.net”) to them. These auto-generated domains are then used by bots to contact the C&C server; if the connection fails, then the bots attempt to use the day information to produce the daily domains, whereas if all the domains fail, then the bots use the hard-coded domain names in their configuration file as a last resort (Stone-Gross et al., 2009). All of these generated domain names are sent to the DNS server in an attempt to resolve it. The bots then establish contact with the C&C server. This process of failed requests generates a high observable number of non-existing domain responses in the DNS traffic that create a footprint of these bots that send most of the failed DNS requests (N. Jiang et al., 2010; Pappas et al., 2009; S. Yu, 2014).

3. SURVEY MOTIVATION

Many surveys discussed the fast flux problem according to the level of fluxing whether it was single or double. The current survey took advantage of discussing the previous proposed fast flux detection methods according to the scope of the solution. There are some shared aspects with the literature, but the current survey presented new aspects such as; Client-based, Router-based, DNS-based, FF passive, FF active, Hydra fast-flux, Neural network approaches, and Cloud computing approaches. Table 1 shows a summary of the comparison of this work with other related studies (Zhang et al., 2011; Zhou, 2015).

Table 1. Summary comparison of this survey with existing studies

		This work	Related studies	
			Zhang et al. (Zhang et al., 2011)	Zhou (Zhou, 2015)
Scope	Client-based approaches	✓		
	Router-based approaches	✓		
	DNS-based approaches	✓		
	FF passive approaches	✓		
	FF active approaches	✓		
	FF real-time approaches	✓	✓	
Surveyed approaches	Machine learning approaches	✓	✓	✓
	Geo-informational approaches	✓	✓	✓
	Hydra fast-flux	✓		
	Neural network approaches	✓		
	Score-based approaches	✓	✓	✓
	Behaviour-based approaches	✓	✓	
	Analysis of each technique	✓	✓	✓
	Comprehensive overview	✓		✓
	Cloud computing approaches	✓		

Additional reasons motivate the conducting of this work as follow:

- Internet is used in different fields, such as education, government services, communication, banking, and e-commerce. However, the growing need for user applications presents a threat to privacy and data security (Stevanovic et al., 2013). The comprehensive survey performed in the current work focuses on fast-flux botnet detection analysis.
- Many botnets distribute HTTP services using the fast-flux of the A records of the domain name of the host or the DNS servers (Burghouwt, 2015). Botnets pose a serious risk to network security and the user privacy of these networks. Web-based communication is mostly used as a medium of botnets and thus makes the detection of web traffic difficult; web traffic constitutes about 70% of Internet traffic (Al-Bataineh & White, 2012). According to a report of losses in 2014, Internet traffic related to spam traffic makes up 80% of botnet traffic (Losses, 2014).

- Fast-flux is an evasion technique used by botnet herders to mask C&C channels, distribute malware, and hide websites (for phishing attacks, download malware, and spam attacks) behind a network of proxies (Sood et al., 2013).
- At present, large-scale botnets, which are ready to launch an attack, consist of more than a million PCs (C.-M. Chen et al., 2013). Botnets perform various types of attacks, such as phishing, spam, and click fraud (Al-Bataineh & White, 2012; Al-Duwairi & Al-Hammouri, 2014; Almomani, 2016; Emre, 2011). All of these facts show the need to unify the efforts to stop the threats of botnets, especially those supported with fast-flux techniques.

4. CLASSIFICATION OF DETECTION APPROACHES AGAINST FAST-FLUX BOTNETS

Multiple studies have investigated the detection of a botnet, particularly the detection of the fast-flux botnet. The detection of FFSNs or malicious fast-flux domains, the one that represent the main element of the fast-flux botnet strategy, was discussed in most prior research. The previous fast-flux works discussed fast-flux in terms of what is fluxed or what method was used to identify Fast-flux domain. However, the current research is the first to explore fast-flux botnet methods based on the solution scope of detection methods to the best of our knowledge.

Figure 5 presents the solution scope of fast-flux botnet detection based on previous studies. In the present work, we classify fast-flux botnet approaches according to the solution scope. Therefore, number 1 in the graph refers to host-based methods, number 2 refers to router-based methods, and number 3 refers to DNS-based methods. Moreover, we discuss the mode of each detection technique and identify whether it is active, passive, or real-time, as depicted in Figure 6.

Figure 5. Solution scope of FF botnet detection methods

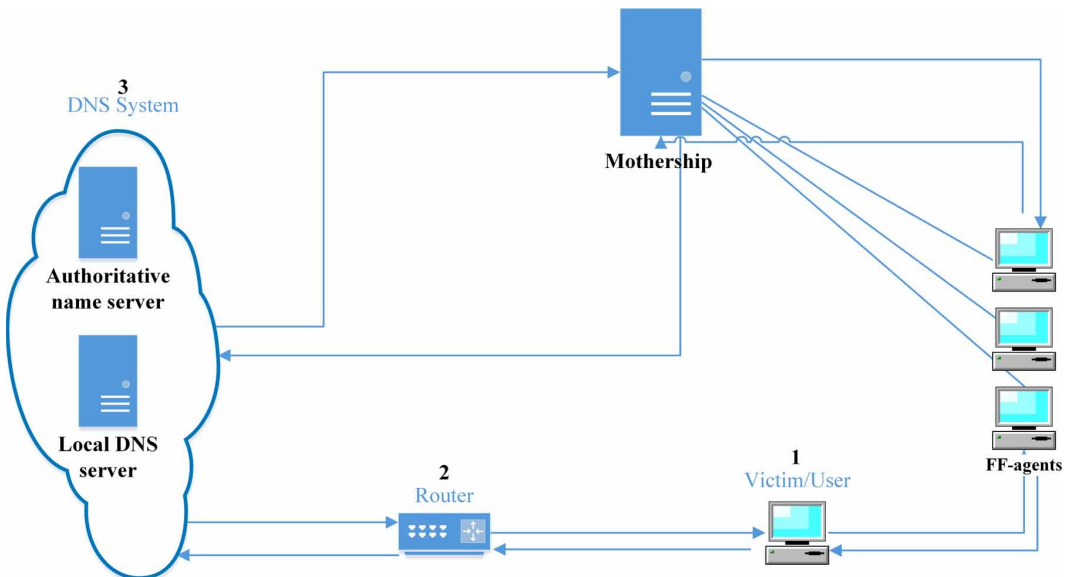
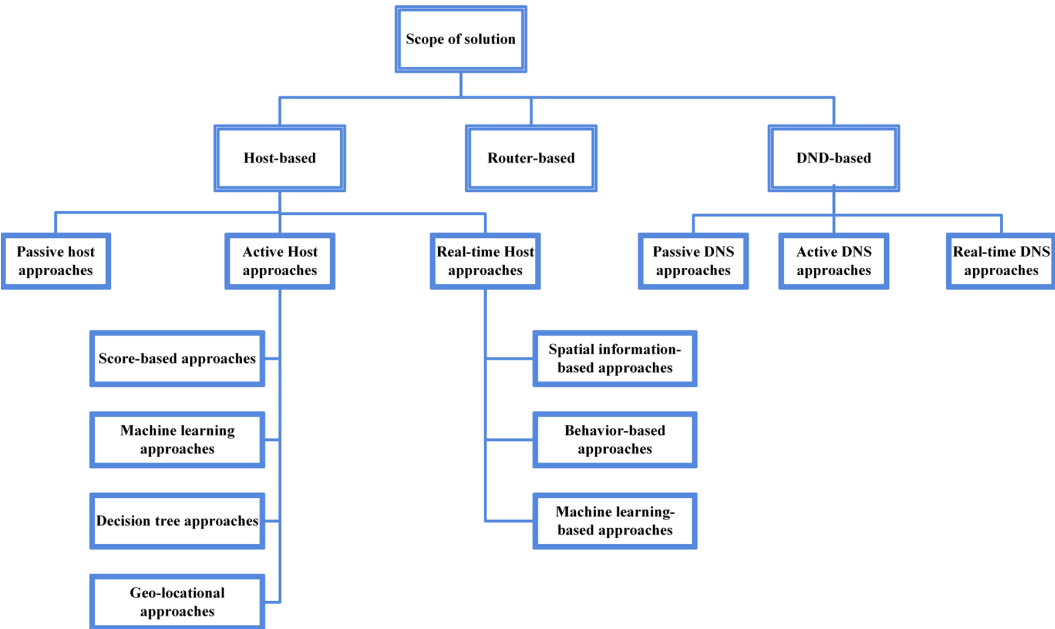


Figure 6. Chart of the solution scope



4.1 Host-Based Detection Methods

Host-based implies the proposed approach is implemented to a host computer or a set of machines. These methods are split into three subsections: passive, active, and real-time approaches, according to the job performed, the majority was a host-based detection approach.

4.1.1 Passive Host-Based Approaches

The idea behind passive approaches relies on the monitoring part of a specific network area for a while. The collected data are then analyzed to prove predefined propositions. Passive monitoring provides the detection methods with the advantage of not being noticed by attackers and adds no extra traffic flows to the network traffic.

A Bayesian method is proposed to detect bots based on DNS traffic similarity (Villamarín-Salomón et al., 2009). The proposed system relies on the idea that a bot at the same botnet has the same traffic similarity as the other botnets. One bot should be known at the beginning; then, the search for other bots with the same traffic similarities in the DNS traffic is initiated. However, the poor tuning of parameters generates low false positive (FP) values (Villamarín-Salomón & Brustoloni, 2009).

Another method of using decision trees to identify malicious FFSNs was proposed in (D. Zhao et al., 2012). The classifier begins to classify malicious domains and then monitors the suspicious ones for a longer period. The proposed system may be able to identify legitimate and malicious FFSNs, but it may not easily classify them based on malicious website behaviours. The proposed system is also unable to detect unknown FFSNs, as well as unknown zero-day domains. Also, the author suggests generating a new system that can develop its classifier while running based on an existing dataset and newly generated data, which would enable the system to identify new threats (D. Zhao & Traore, 2012). Table 2 summarizes the passive approaches.

Overall, fast-flux botnets still need to be detected in a short time because of the quick change in the IP addresses of motherships that hampers the easy tracking of their locations. Thus, detecting this type of “Fast-flux zero-day” domains as quickly as possible is important. Moreover, passive

Table 2. Summary of passive approaches

Authors	Algorithm	Mechanism	Advantages	Disadvantages
(Zhao & Traore, 2012)	Decision tree	Monitoring malicious domains to detect FFSNs	Low computational complexity	- Classification problem - Unable to detect unknown zero-day domains
(Villamarín-Salomón & Brustoloni, 2009)	Bayesian method	Detecting bots based on DNS traffic similarities and known bot traffic	Effective and robust	Parameter tuning causes FP

approaches deal with a huge amount of data and are thus unsuitable for fast processing in a short time with few resources.

4.1.2 Host-Based Active Approaches

In contrast to passive approaches, active approaches require assistance from third-party data sources, such as the WHOIS or GeoIP database. Such third parties provide the additional necessary information (e.g., IP address registrar name and creation date). The following subsections describe related works that applied host-based active approaches.

4.1.2.1 Score-Based Approaches

Many fast-flux domain detection approaches are based on the flux score calculation of a set of features adopted by (Holz et al., 2008; Hsu et al., 2014; Karim et al., 2014; Otgonbold, 2014; Yukonhiatou et al., 2014). (Holz et al., 2008) proposed a system that measures and detects a FFSN based on the calculated flux score. Their proposed system takes malicious domains from spam emails and then uses the Dig tool to generate DNS lookups and reverse DNS lookups and thereby obtain necessary information about a feature set (number of A records, number of autonomous system numbers (ASNs), and number of NS). Thus, the flux score calculation is fed for use later in distinguishing between malicious FFSNs and legitimate ones. Their results showed that the proposed system achieves a detection accuracy of 99.98%. However, the coefficients used in the score calculation require modification to ensure the highest possible accuracy of the detection system. Moreover, the set of features chosen cannot purely distinguish between FFSNs and CDNs.

(Hsu et al., 2014) proposed a fast-flux domain detector (FFDD) system, thus adding to Holz's source of malicious domains and taking unknown URLs from spam or social networks. The FFDD system is used to calculate the flux score based on the response time series between each of the two subsequent requests from a host to the FF-agent. The FFDD is a lightweight standalone system that does not need support from other parties. Consequently, the FFDD can accurately detect a fast-flux domain with 3% FP and 2% FN in less than 20 min. Therefore, this technique is not suitable for fast-flux detection.

(Sheng et al., 2010) proposed two metrics, namely, the average online rate (AOR) and the minimum availability rate (MAR) to detect fast-flux agents based on the agents themselves. The calculations of these two methods are initiated from the beginning of the monitoring process. The monitoring is extended for one h using the AOR and MAR calculation once a malicious domain is detected. The results show that most FFSNs have lower values than legitimate ones. Moreover, these methods are easy to implement and deploy and are useful for distinguishing between benign and malicious FFSNs but not for FFSN detection. However, the metrics may work incorrectly if the group of agents is small or a few agents are found (Sheng et al., 2010). According to Sheng et al., the metrics depend on the quality of the HTTP service, which may affect network accessibility and thus stop reaching agents.

The Google search engine has also been used as a technique to classify malicious domains by feeding the search process with IP addresses of suspicious domains (Losses, 2014). The number of hits is then observed. As expected, the number of hits comprising domains associated with FFSNs would be much less than the number of legitimate domains. The new legitimate domain could also mislead the classifier. The proposed system is still at its infancy and thus needs other features to confirm its detection accuracy.

Koo et al. (Karim et al., 2014) proposed a computed formula to detect malicious domains being used in FFSNs, with the domains obtained from a malware domain list. They explored the actual status of FFSNs employed in cyber-crimes and analyzed the distribution of compromised computers. Consequently, the detection accuracy is high. However, their data were not sufficient to estimate the scope of the FFSN. Thus, their proposed procedure may lead to misclassified domains.

(Yukonhiatou et al., 2014) proposed a probability formula to detect malicious fast-flux domains. The network behaviour of malicious domains is formalistic based on the time-space behaviour of malicious FF-domains. Besides, an analysis was proposed to reduce the time complexity of feature modelling. The results of this study show that the proposed solution performs better than blacklists. However, a threshold is still needed to compute the probability formula. Moreover, gathering information about domain names requires more time, which affects detection performance.

Otgonbold (Otgonbold, 2014) proposed a fast-flux formula to help detect fast-flux domains in the wild. The proposed ADAPT system takes inputs from the domain zone file to collect the DNS information needed in the detection system. The zone file is targeted because it contains domains scattered all around the globe using the Tor network (Figure 7). The system's clients gather suspicious domains from various DNS servers over the Tor network and then analyzes the collected information. Thus, the decision as to whether the domain needs further scanning to confirm its maliciousness is made. The results of this study indicate that the proposed system is capable of detecting malicious fast-flux domains in their infancy. However, the RDNS server should be queried to collect full DNS information, and such a requirement could affect detection performance. The current version of Grails also shows a memory leak problem, which causes out-of-memory exceptions and long-running tasks.

Another work by (Guo et al., 2018) to detect malicious domains that cloak fast-flux. To ensure their collection and evidence is anonymous and spread, they incorporated their systems into the Tor scheme. With the information gathered, they created algorithmic data analytics to extract the malicious fast-fluxing and web domains and classify them.

Figure 7. ADAPT system architecture (Otgonbold, 2014)

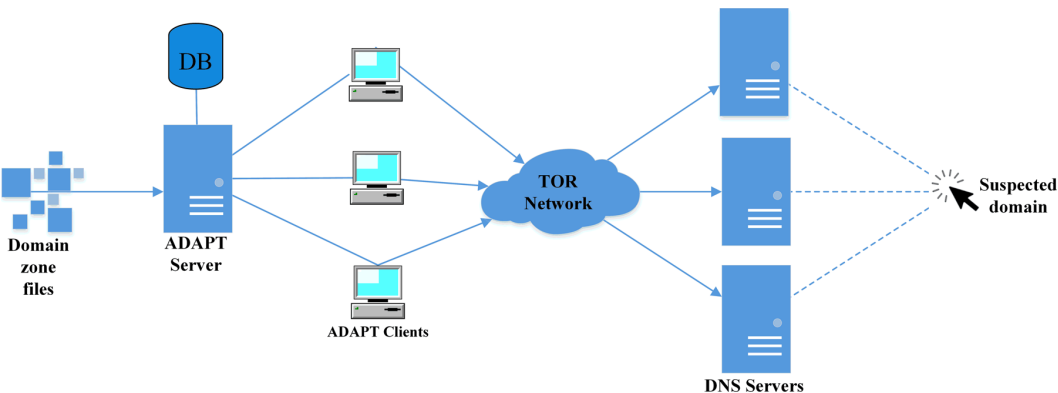


Table 3 summarizes the calculated score-based approaches.

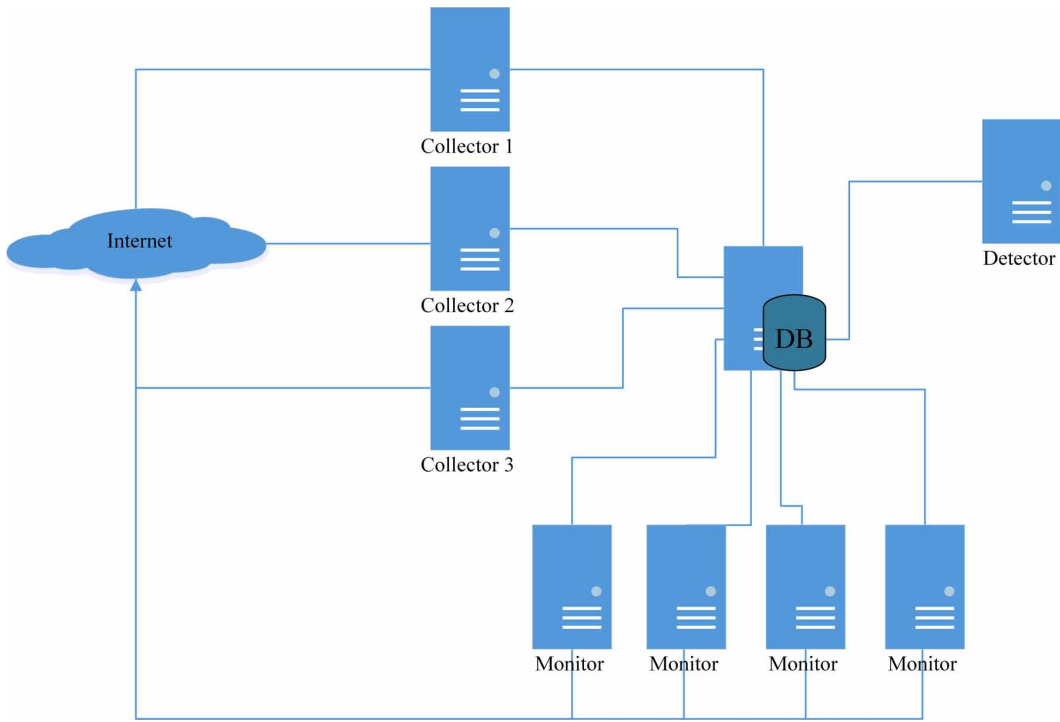
Table 3. Summary of score-based approaches

Authors	Algorithms	Mechanism	Advantages	Weakness
Holz et al. (Holz et al., 2008)	Flux score	The flux score is computed based on DNS records	Uses two consecutive DNS lookups	- Coefficients require periodic adjustment - The feature is not distinguishable
Hsu et al. (Hsu et al., 2014)	Flux score	The fast-flux score is computed based on the response time differences of subsequent requests of FF domains	Lightweight, stand-alone system	Long detection time
Sheng, Shijie, and Sha (Sheng et al., 2010)	AOR, MAR	Once the existence of a fast-flux domain agent is discovered, its activities are monitored every hour using calculations based on AOR and MAR	Easy to implement and deploy; metrics are time-saving	- Inaccurate result - Based on the quality of the HTTP service
Al-Duwairi et al. (Losses, 2014)	Number of hits in the Google search engine	Depending on the number of hits of query responses using the Google search engine	Lightweight approach	- Still in the development phase - Needs more features to confirm detection accuracy - Misclassifies new domains as malicious
Koo et al. (Karim et al., 2014)	Calculated formulas	Calculated formulas based on the actual status of the FFSN being employed	High detection accuracy	- Data problem - Misclassified domains
Chen et al. (Yukonhiatou et al., 2014)	Probability formula	- Time-space behaviour of malicious FF domains and network behaviour of domains are formalistic	Outperforms blacklists	- Threshold is needed - Long detection time
Otgonbold (Otgonbold, 2014)	Flux score formula	- Detection system collects domains from DNS zone files - Anonymously provides domains all around the globe in a short period with a little resource using the Tor network	Detects malicious fast-flux domains in their infancy	- RDNS servers should be queried, which could affect performance - Out-of-memory exception
(Guo & Guan, 2018)	TOR network	- They incorporated their systems into the Tor scheme. With the information gathered, they created algorithmic data analytics to extract the malicious fast-fluxing and web domains and classify them.	These are helpful indicators to detect malicious domains	Further data handling systems for storage and time efficiency

4.1.2.2 Machine Learning-Based Approaches

Several machine learning algorithms are used to classify domains as either malicious or benign (Chen et al., 2014; Passerini et al., 2008) (Table 4). In the naïve Bayes classifier proposed by Passerini et al. (Passerini et al., 2008), all malicious domains are collected from spam emails. Their detection and monitoring “FluXOR” system rely on the idea of a host being a victim to such a scam. The system begins to send requests and gathers the feature set information to feed the naïve Bayes classifier (Figure 8). The naïve Bayes classifier is a supervised algorithm, which is not suitable for detecting unknown attacks. FluXOR reduces the time of detection to 1–3 h, which is still relatively long; a domain with a TTL of more than three h is still considered legitimate (Huang et al., 2010).

Figure 8. FLUXOR system deployment (Passerini et al., 2008)



Chen et al. (Chen et al., 2014) proposed a Bayesian probability theory to distinguish between benign and malicious domains using dissimilar ASNs, reverse DNS lookups, and domain registration time features. They aimed to detect a fast-flux website based on its fluxed characteristics. The result of this proposed system presents its ability to identify possible threats. Nevertheless, their judgment was not perfect enough to reflect the good precision of the proposed system.

(Chen et al., 2011) used the k-nearest neighbor (KNN) and random forest (RF) as sampling techniques to solve the imbalanced problem, concerning FFSN detection. Besides, they proposed a sampling technique that is combined with feature extraction from datasets for use in fast-flux detection. The result showed that the TTL is an important feature to the classification of the proposed technique. However, its detection accuracy in the case of a long TTL is affected.

The support vector machine (SVM) was proposed by (Yu et al., 2012) to detect fast-flux botnets by analyzing the patterns of DNS queries from FF botnets. They extracted six features to build the weighted SVM classifier for use in distinguishing legitimate and FF botnet domains. They noted that using SVM to identify fast-flux botnets is effective and provides a satisfactory detection accuracy. Overall, the proposed method entails a long detection time because it waits for additional information from a third party. Moreover, such a supervised method does not help detect new and unknown zero-day attacks.

4.1.2.3 Decision Tree-Based Approaches

Celik and Oktug (Celik et al., 2013) proposed the C4.5 decision tree algorithm to evaluate various DNS feature sets and put forward a detection framework, which is a high-dimensional feature vector with various features, including timing network, spatial, and NS and DNS response information. C4.5 evaluates each feature set of previous vectors and decides which one is the best feature vector based on detection accuracy. Combining all features provides a detection accuracy of 98.9%. However,

Table 4. Summary of machine learning approaches

Authors	Algorithms	Mechanism	Advantages	Weakness
Passerini et al. (Passerini et al., 2008)	Naïve Bayesian classifier	Analyzes a set of features observed from the victim's point of view on botnet scams	Reduces detection delay	- Long detection delay - Unable to detect zero-day domains
Chen et al. (Chen et al., 2014)	Bayesian probability theory	Uses different characteristics to distinguish benign and malicious domains	Enhances detection accuracy of web-based botnets	- Achieves inaccurate precision
Chen et al. (Chen et al., 2011)	KNN and RF	Use the resampling technique to solve the imbalanced classification problem concerning FFSN detection	Solve the imbalanced dataset problem	- Long TTL affects detection accuracy
Yu, Zhang, Kang, and Chen (Yu et al., 2012)	Weighted SVM	Extracts six features to the weighted SVM by analyzing the patterns of DNS responses to FFSNs	Satisfies detection accuracy	- Earlier domains create FP - Unable to detect zero-day domains

the detection is unaffected in those timing and domain name feature sets. The C4.5 unsupervised algorithm depends on clustering and is good for detecting unknown attacks; however, it achieves a low accuracy level in most applications (Almomani, Gupta, et al., 2013).

(Zhao & Traore, 2012) proposed another method (REPTree) for botnet detection using a decision tree with reduced error pruning. This type of machine learning decision tree is used to classify and identify malicious FFSNs by defining and computing some of the network metrics captured from network flows. Although decision tree-based classifiers are considered as a well-known classification technique with low computational complexity, the authors were not sure of the result because some benign websites were misclassified as malicious websites. They also searched for other reliable evidence. Table 5 summarizes the approaches using the decision tree algorithm.

The classification and regression tree algorithm are used in the method proposed by (Zhao et al., 2015). This method uses a small dataset to quickly distinguish legitimate and malicious FFSNs. This method is mainly based on FFSN domains, DNS, and the process of HTTP visiting. The distinct domain features are shown in Figure 9. Another researcher used distinct mapping of features (Pa et al., 2015). The classification process needs less than a few days, and the detection accuracy is 90%. The detection time is also relatively long, and other detection methods exhibit higher accuracy and lower detection time. Moreover, this method cannot detect zero-day domains.

4.1.2.4 Geo-Informational Based Approaches

A constraint-based geolocation technique was employed in previous work (Castelluccia et al., 2009), and the proposed framework utilizes a geo-localized fast-flux hidden server. Thus, mean error distance is used in this framework to determine the physical location of the mothership server. As a result, their framework localizes the mothership with a mean error below 100 km. However, the system requires extensive resources, achieves low precision, and is incomplete. (Buhariwala, 2011) used the same technique and determined that the 100 km mean error is inaccurate; moreover, the result indicated that the right error value is 1,000 km from the mothership server. A virtual private proxy server was proposed to decrease the overhead of requesting data from the content server. The result

Figure 9. Process of visiting a domain (Zhao & Jin, 2015)

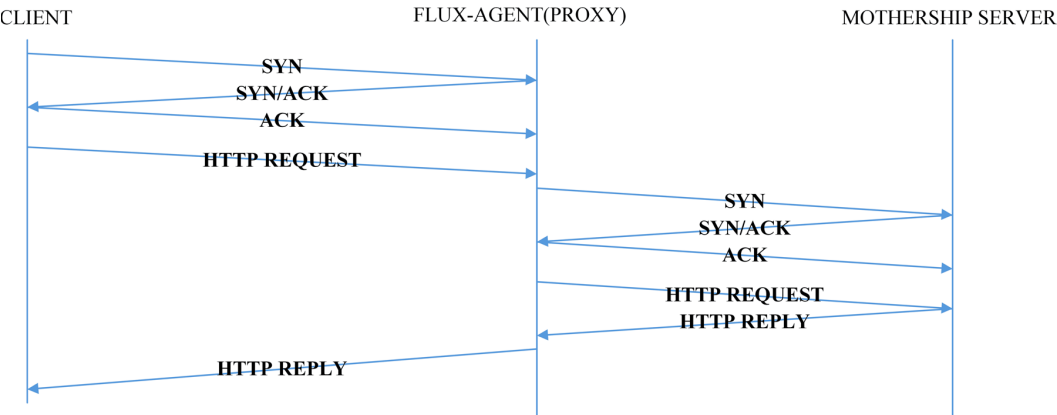


Table 5. Summary of approaches that use decision tree algorithm

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Celik and Oktug (Celik & Oktug, 2013)	C4.5 decision tree	Several feature sets are experimented on to detect FFSN.	Detect unknown attacks	- Unaffected by some of the feature sets - Low level of accuracy
D. Zhao and Traore (Zhao & Traore, 2012)	Decision tree using reduced error pruning (REPTree)	Computed metrics of captured network flows that are analyzed using REPTree.	Low computational complexity	-Misclassification -Needs additional discriminators
Y. Zhao and Jin (Zhao & Jin, 2015)	Regression tree algorithm	Detect FFSN domains based on the intrinsic features of DNS analysis and the process of HTTP visiting.	Ability to classify FFSN domains	-Long detection time -Low accuracy rate -Unable to detect zero-day domains

indicated a 300 km mean error, which is three times better than that obtained by Castelluccia et al. A large mean error rate for physically localizing the mothership server still exists. Table 6 summarizes the methods that utilize geo-information.

The system proposed by (Stalmans et al., 2012) used Moran's I and Geary's C to produce classifiers to detect the fast-flux C&C domain names of C2 servers. The proposed system can detect domain names based on the geographic locations of C2 servers. Moran's I assumes that close geographical C2 servers are similar, whereas Geary's C measures the spatial autocorrelations between C2 servers. Their system can reliably detect FF domains with a low FP rate. Moran's I measurement is influenced by the number of white spaces at a large scale (Stalmans et al., 2012).

(Stornig, 2013) employed another approach in which Moran's I of spatial autocorrelation and spatial service distance are used to classify legitimate and non-legitimate fast-flux domains. This approach is based on the geo-information of the distributed IP addresses of FF-agents. The spatial autocorrelation between two distant geographical points means that they are not similar, and close points share more similarities. The spatial service distance denotes the average distance between the geolocation of the IP addresses that are correlated with the same domain and the geolocation of the IP addresses of the name server. As a result, the author was convinced that the proposed approach is accurate and lightweight for detecting fast-flux domains with low FPs. However, botmasters could cause the detection approach to yield misclassified results by changing the distribution of the IP addresses of the agents.

Table 6. Summary of approaches using geo-information

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Castelluccia et al. (Castelluccia et al., 2009)	Constraint-based geolocation technique	Determines the physical location of the FF mothership based on network measurements	Can localize with a mean error distance below 100 km	-Requires extensive resources to set up -Less precise and less complete
Buhariwala (Buhariwala, 2011)	Constraint-based geolocation technique	Determines the physical location of the FF mothership based on network measurements	Decreases the overhead of requesting content servers	Inaccurate rate (300 km)
Stalmans et al. (Stalmans et al., 2012)	Time zone, UTM, MGRS	Identify fast-flux domains on the sole basis of the geographic locations of C2 servers	Only a small percentage of FPs	The classifier is affected by a large amount of whitespace
Stornig (Stornig, 2013)	Moran's I of spatial autocorrelation and spatial service distance	Utilizes methods of geo-information and spatial statistics	-Lightweight system -Avoids FPs	Could be misclassified by botmasters

The problem with active detection-based approaches is that they deal with minimal DNS traffic traces, which correspond to non-legitimate domain names in most cases. According to the nature of active approaches that mostly deal with malicious domains, they are unable to detect unknown zero-day domains.

4.1.3 Host-Based Real-Time Approaches

The previous methods involve passive and active approaches, which presented many detection techniques to detect malicious fast-flux botnet domains and FFSNs. Fast-flux detection needs a quick and precise strategy before changing their IP addresses to recognize malicious domains. Thus, to boost the power of detection methods, a fresh era of real-time methods was created. The main idea behind employing real-time approaches is to reduce the time needed to detect attacks to real-time processing.

4.1.3.1 Spatial Information-Based Approaches

(Caglayan et al., 2009) were the first to conduct a related empirical study. The authors presented a fast-flux monitor (FFM) that could detect and classify FFSNs in real-time within minutes. The FFM comprises active and passive DNS monitors, which reduce the long-term observation of FFSNs. Using active and passive monitoring can reduce observation duration, but the system still requires a few additional minutes. Obtaining extra information from a data center helps classify botnet domain names.

(Huang et al., 2010) proposed a real-time system called spatial snapshot fast-flux detection (SSFD). SSFD detects FFSNs by extracting the IP addresses of the hosts (agents) from the DNS responses and determining the geographical traffic patterns of these agents in a geographic coordinate system. Two spatial measures were used: spatial distribution estimation and spatial service relationship evaluation. A Bayesian network classifier was also employed to distinguish FFSNs from benign networks. The experimental results indicated that SSFD is effective (less than 0.5 s) and yields lower FP rates than flux score detection systems through their data sets. However, SSFD suffers from a single IP problem and missing geographical information problem, which may cause the system to malfunction. The experiments verify that the detection accuracy is 62% (Lin et al., 2013).

A Bayesian network classifier algorithm classifier was proposed by (Horng-Tzer et al., 2012) to detect FFSNs in real-time. The authors believed that the grid distribution of the localized spatial-locating capability is ideal for depicting the spatial relationship between the resolutions of IP addresses. To enhance the localized geo-locational characteristics, the proposed system incorporated ASNs, localized spatial geo-location detection (LSGD) system, and DNS to achieve the identification of potential FFSNs. The authors believed that the detection capability of the LSGD system is better than that of spatial or temporal detection approaches. The LSGD system exhibits a lower FP rate than the spatial snapshot system in real-time detection, which is completed within a few seconds. However, the highest FP rates are caused by CDNs, which have a similar localized spatial distribution signature that affects accuracy. Table 7 shows a summary of spatial informational approaches.

Table 7. Summary of spatial informational real-time approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
(Caglayan et al., 2009)	Bayesian belief network	-Bayesian classifier employs multiple active and passive DNS sensors -Generates a probabilistic assessment of the existence of FFSNs	Reduces the observation period	-Long-time -Datacenter help is needed
(Huang et al., 2010)	Bayesian network classifier and K2 algorithm	Determines the geographic traffic patterns of hosts and maps the IP address of a DNS response in a geographic coordinate system	Lower FP rate than flux score-based detection	-Single IP problem -Missing value problem
(Horng-Tzer et al., 2012)	Bayesian network classifier and K2 algorithm	Propose LSGD system for identifying FFSNs in real-time	Better detection capability than spatial or temporal detection approaches	-Misled by CDN service sites

4.1.3.2. Behavior-Based Approaches

Many researchers have studied the behaviour of the changes in fast-flux domains. (Caglayan et al., 2010) modelled the behaviour pattern of FF botnets based on DNS resource records using a Bayesian classifier. The authors determined that botnets exhibit common characteristics and form clusters according to botnet size, growth, and operations. Their findings show that a majority of fast-flux botnets operate in at least five countries and between 20 and 40 countries on average. Unfortunately, their approach is misled by benign servers, such as CDNs, thus resulting in a high number of FPs (Caglayan et al., 2010).

(B. Yu et al., 2014) addressed the behaviour of fluxed domain changes and proposed a novel time-series model based on carefully selected features. Their model uses network security and a semi-supervised training framework to overcome and identify difficulties in known supervised machine learning approaches. A horizontal scalable online system was proposed to deal with a large amount of data that pass through a network in a real deployment. Their system can identify flux domains despite the presence of long TTLs or a limited number of mapped IP addresses. Actual latency is determined by an online system (10 min) given a domain name, whereas most active threats can be

Table 8. Summary of behavior-based approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Caglayan et al. (Caglayan et al., 2010)	Bayesian classifier	Modelling the behavioural patterns of fast-flux botnets using DNS records	-Botnets operate in 20 to 40 countries - < 250 ASNs	- Misled by CDNs -The high number of FPs
B. Yu et al. (B. Yu et al., 2014)	Time-series model	-A time-series model -A horizontally scalable online system	Captures fast-flux domains	-Long detection time. -FN rate is not considered

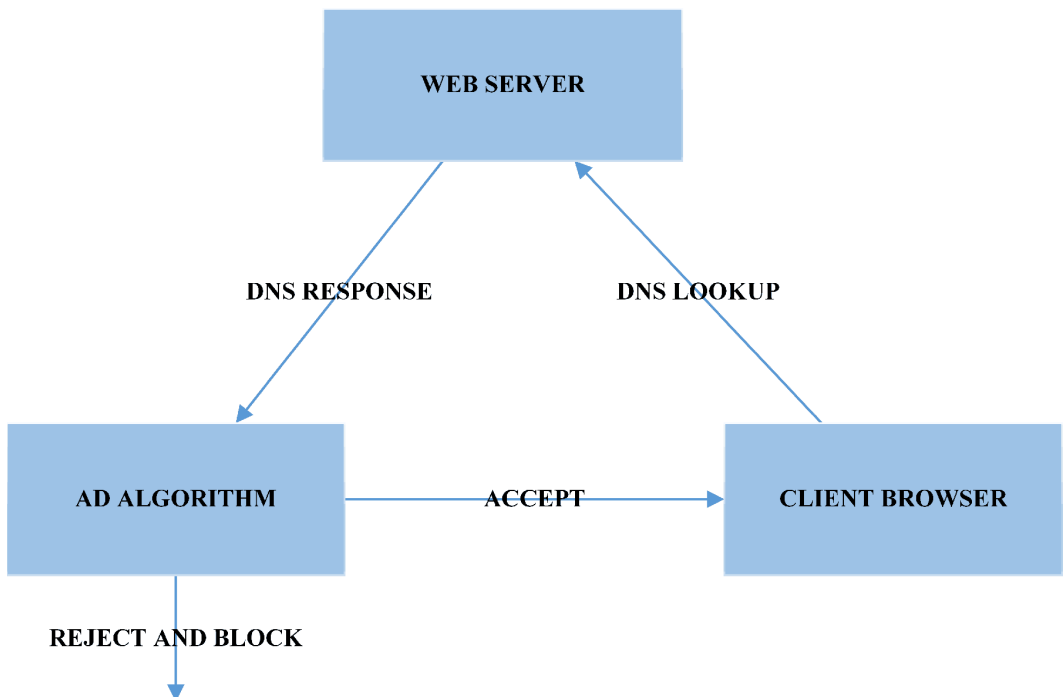
detected in less than 10 min. Their approach does not address the FN rates in the evaluation of results. Table 8 shows the behaviour-based approaches.

4.1.3.3 Machine Learning-Based Approaches

(Qassrawi et al., 2012) used the algorithm of an alternative decision tree (Gothai et al., 2012) to determine whether a domain is an FF domain or not. Figure 10 shows that only one DNS response resource record is needed to achieve fast detection in real-time. Previous studies show that DNS information is insufficient to detect FF botnets (Martinez-Bea et al., 2013).

Unlike Qassrawi, (Hsu et al., 2010) proposed a real-time system to measure the delay for HTTP responses by relaying user requests from an FF-agent to back-end servers. Thus, a long delay means that a host (FF-agent) relayed a request to another server. The authors proposed this real-time system to reduce detection time to a few seconds without affecting detection accuracy (96% with FP and FN rates below 5%). The authors carried out a classification based on supervised learning using SVM

Figure 10. Alternative decision tree detection framework (Qassrawi & Zhang, 2012)

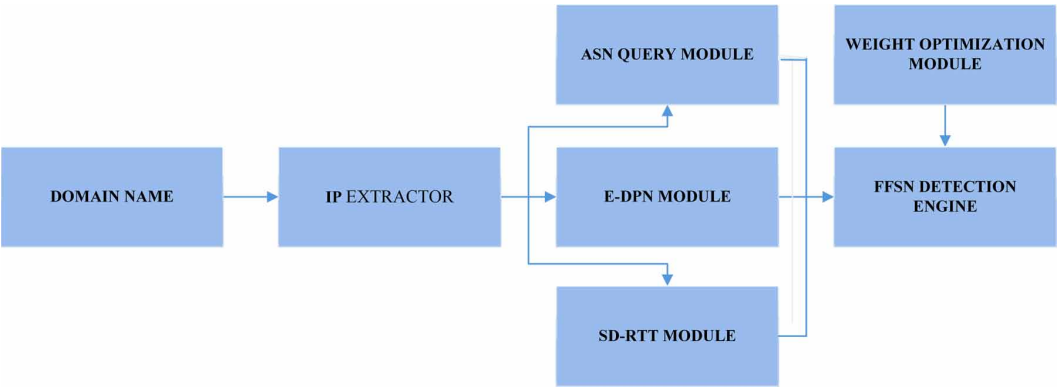


trained on six features. The delays in the relaying request are counted because of the limited power and bandwidth of the relaying hosts (FF-agents). However, extracting the six features from this volume is time-consuming. Thus, keeping the detection time within the real-time range is difficult. The proposed system cannot effectively detect fast-flux domains with long TTLs, and the detection accuracy is 67% (Lin et al., 2013). The proposed detection system cannot detect zero-day domains.

SVM was built by D. Kevin McGrath to detect fluxed phishing domains (D. Kevin McGrath, 2009). The classifier was trained based on the features extracted from the DNS responses, such as the number of IP addresses related to one domain, ASNs, number of different prefixes, and number of countries of an IP address. The main limitation of the classifier based on the nature of its features is that it can be misled by botmasters. The previous classifier proposed in (Hsu et al., 2010) can be misled by a benign server, such as a CDN or RRDNS. A new SVM classifier proposed by (Martinez-Bea et al., 2013) was trained on real features from both domains and bots. Combining the two feature sets from the two previous approaches (McGrath, 2009; Hsu et al., 2010) increased the TP and TN rates and decreased the FP and FN rates. However, the author mentioned that the proposed method for detecting fast-flux domains may still be evaded theoretically and that the proposed detection system cannot detect zero-day domains.

The genetic-based real-time approach for FFSN detection (GRADE) was proposed by Lin et al. (Lin et al., 2013). The authors assumed that fast-flux bots are distributed arbitrarily in many places in the world. Thus, the distances between bots (FF-agent) and users differ. The fast-flux domains would result in significant differences in the round-trip time between the user and the agents. The GRADE system architecture is depicted in Figure 11. GRADE can more effectively detect FFSNs (within a few seconds) than flux scores and is more accurate (98%) than fast-flux bot detection and SSFD. However, GRADE suffers from the single IP problem, in which only one point in the geographic coordination system may cause GRADE to malfunction. Table 9 summarizes real-time machine learning approaches.

Figure 11. GRADE system architecture (Lin et al., 2013)



In most cases, malicious domain names and malicious FFSNs can be detected by the proposed real-time approaches. The above methods, however, have some constraints that cast doubt on their outcomes (precision, TP, TN, FP, and FN). We still lack a stable method in an acceptable period with elevated detection precision that can detect malicious domains, especially zero-day domains.

Table 9. Summary of real-time machine learning approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Qassrawi and Zhang (Qassrawi & Zhang, 2012)	Alternative decision tree	One DNS response RR is needed to achieve FF detection in real-time	One DNS response RR is needed	Insufficient features to conduct classification
Hsu et al. (Hsu et al., 2010)	Linear SVM algorithm	Observes longer delays for HTTP responses as a result of relaying the requests via fast-flux agents	-Real-time -Robust -Lightweight	-Long detection time -Cannot detect long TTL domains -Unable to detect zero-day domains
Martinez-Bea et al. (Martinez-Bea et al., 2013)	Linear SVM algorithm	Builds an SVM classifier trained via real features extracted from domains and bots to differentiate malicious FFNs	-Increased TP and TN -Reduced FP and FN	Unable to detect zero-day domains
Lin et al. (Lin et al., 2013)	Genetic algorithm	The distances between clients and flux bots vary significantly	Outperforms other systems, such as flux score, FFBD, and SSFD	Single IP problem

4.2 Router-Based Detection Methods

Many researchers have extracted various information from network traffic to solve several network issues for the fast-flux botnet issue in general and in particular. DNS traffic and non-DNS traffic include network traffic. Recent studies (Paul et al., 2014) did not rely on DNS data traffic.

A previous work (Paul et al., 2014) aimed to cluster similar packets in data traffic from both router sides, assuming that the C&C servers had to change their IP addresses automatically. The approach assembles all packets, as shown in Figure 12, between the C&C server and the host for analysis and obtains the malicious pattern in each cluster. The detection accuracy of this approach to malicious traffic is 95.8%, and its low FP rate is 1.6% in the worst case. However, the approach suffers from a scalability problem. Thus, when data traffic is insufficient, the malicious packet sensitivity decreases.

A new online Botnet detection classification technique based on the characteristics of DNS traffic that make Botnet different from CDN-based traffic is presented in (Cafuta et al., 2018). Botnet characteristics are categorized in an embedded system according to their usability and execution. As a powerful candidate for online detection, traffic responses are analyzed. Its inconvenience resides in certain regions in which CDN operates as a botnet. Table 10 summarizes the router-based approaches.

The table above clearly indicates that constructing router-based systems to detect fast-flux botnets may produce acceptable results for the authors. However, the speed and high quantity of information that passes the router cause three key issues in systems building: high false rates based on the notion of quick detection of Fast-flux botnets, data-based issues in memory and scalability. Thus, fast-flux domains and especially zero-day domains detection methods are ineffective in this portion of the network.

4.3 DNS-Based Detection Methods

Researchers studied DNS traffic in their home nation. Their research was therefore concentrated on surveillance and analysis of data traffic and malignancy, such as fast-flux botnets. Some researchers used passive, active, and real-time methods, as outlined in the subsections below.

Figure 12. System architecture of the proposed detection method (Paul et al., 2014)

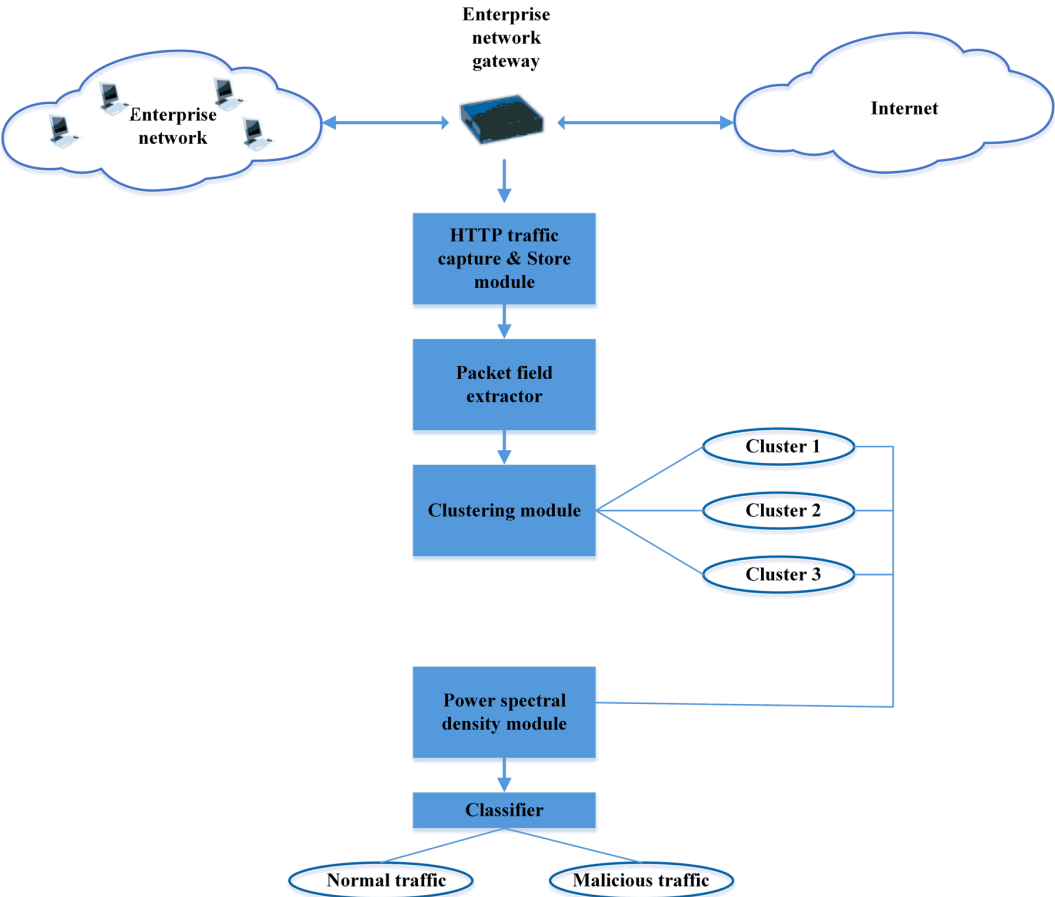


Table 10 .Summary of router-based approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Paul et al. (Paul et al., 2014)	PSD value used as a classifier	Computes the power spectral density (PSD) for each cluster and feeds it to the classifier, which examines the PSD data for significant peaks	Detects traditional HTTP and fast-flux botnets	-Scalability problem -Malicious packet detection sensitivity problem
(Cafuta et al., 2018)	An embedded system	Botnet characteristics are categorized in an embedded system according to their usability and execution.	Improve the FP rates	Misclassifying CDN and FFSN

4.3.1 Passive Approaches

Researchers monitored DNS servers and analyzed data traffic passively to detect malicious activities. (Gržnić et al., 2014) presented a detection system called CROFlux that detects fast-flux domains relying on a passive DNS replication method. Their system aims to reduce FP rates and detect unknown fast-flux domains with flux characteristics, which are usually used to share malware. Thus, the approach avoids the reporting of legitimate domains with similar characteristics. The proposed system suffers from a design problem because it does not utilize active DNS requests to feed the system, and many IP addresses can enhance fast-flux detection (Gržnić et al., 2014). The proposed system cannot detect zero-day fast-flux domains because the classification process depends on the comparison of the number of malicious domains in the candidate fast-flux cluster with predefined fixed malicious domains.

A scalable and fast approach proposed by Kwon et al. (Kwon et al., 2016) detects fast-flux botnets based on large-scale DNS traffic. This approach analyzes the collected large-scale DNS data traffic to extract malicious behaviours. A signal processing technique, namely, PSD analysis, is leveraged to determine the main frequencies from the periodic DNS queries initiated by botnets. Their system detection accuracy is 95%, given its detection of 23 unknown and 26 known botnet groups with 0.1% FP. However, the proposed method relies on the number of hosts. Thus, increasing the number of hosts should decrease speed and detection efficiency. A threshold number should be assigned according to the circumstances of DNS servers; such threshold number differs for all DNS servers (Kwon et al., 2016).

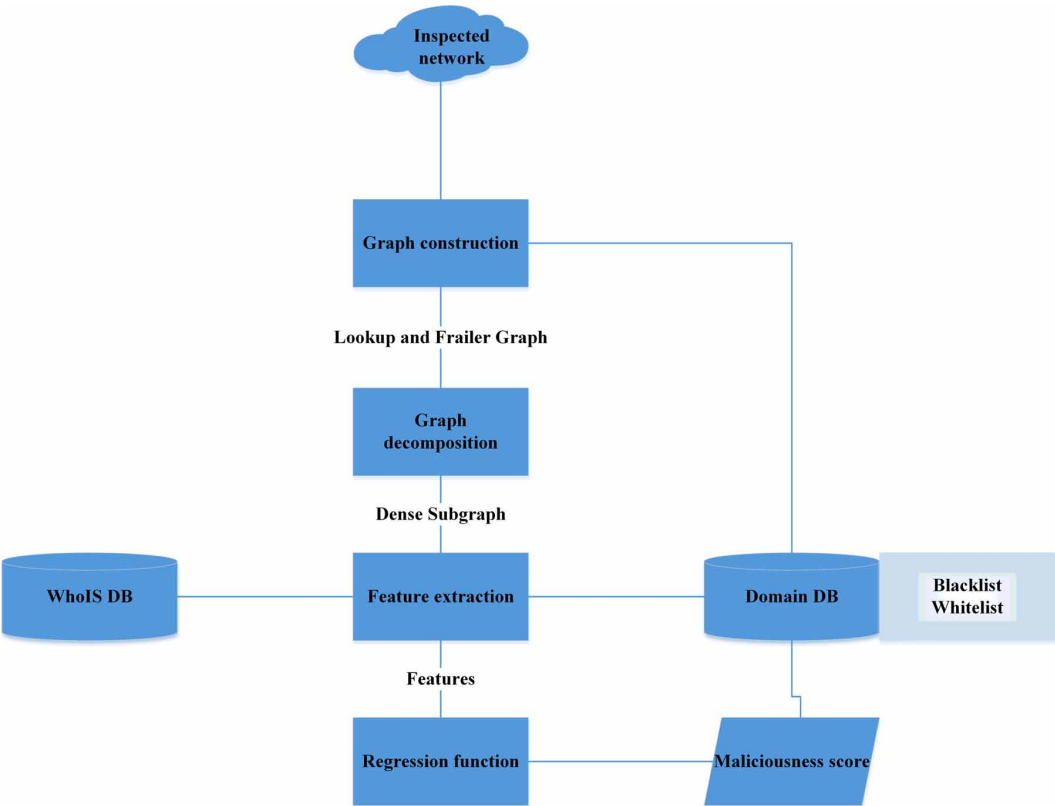
Some decision tree algorithm versions have been used for many detection techniques, such as the system proposed by (Perdisci et al., 2009), which passively collects recursive DNS queries and responses by deploying multiple sensors in front of RDNS servers in two ISP networks. They analysed the extracted features to detect malicious FFSNs using a C4.5 decision tree. Their experiments showed that they accurately distinguished malicious and legitimate FFSNs. They used a statistically supervised learning approach to build a service classifier. Thus, the classifier cannot detect malicious zero-day flux services.

Similarly, (Perdisci et al., 2012) proposed a novel passive DNS system called FluxBuster using C4.5 decision tree as a classifier; this system analyzes DNS traffic for malicious FFN detection and blocking. Their approach gathers DNS traffic generated from hundreds of RDNSs, which are scattered in many networks around the world. A large-scale analysis is carried out based on the resultant traffic. Thus, FluxBuster can detect unknown FFNs before they are reported in a public blacklist. However, the detection system waits for a user to click on a domain name to initiate a request and detect a domain. Furthermore, more IP addresses are needed to set the threshold value of their classifier.

An anomaly-based technique using a decision tree with AdaBoost algorithm was proposed in previous work (Vu Hong, 2012). This approach depends on the passive analyses of extracted DNS data traffic to detect fast-flux botnets. Two graphs were constructed, namely, the lookup and failure graphs, from the extracted DNS traffic. The resulting graphs were distributed into clusters, as depicted in Figure 13. These clusters exhibited a strong correlation between traffic elements (domain, host, and IP addresses). The related features of DNS traffic were extracted from these clusters to feed the classification module in the detection system and identify the existence of a fast-flux botnet. The authors believed that they succeeded in detecting a fast-flux botnet from traffic analysis. However, the system produces FP rates when the number of domain names in a malicious subgraph was small and produced FN rates when a benign subgraph included a large number of random-looking domain names. The malicious characteristics exhibited by the subgraph were not sufficiently distinctive for the technique to obtain.

(Soltanaghaei et al., 2015) Proposed a method for passive network DNS analysis that involves a background for each domain to be evaluated, their method achieved 94.44% identification and 0.001% false-positive levels in their best test. Moreover, (Lombardo et al., 2018) 's Algorithm uses a similar

Figure 13. Analysis procedure (Vu Hong, 2012)



strategy, but by choosing parameters more carefully, better results are obtained when performing an experiment in a near-real-term way. Table 11 summarizes passive DNS-based detection approaches.

4.3.2 Active Approaches

An active approach (Zhou et al., 2009) adopts a collaborative detection system based on a decentralized correlation model called large-scale intrusion detection to detect fast-flux phishing domains by analyzing the relationship between the number of IP addresses and DNS requests from different networks. Figure 14 shows the combination of different DNS server responses to quantify the probable time to be saved. The results indicated that combining evidence from multiple DNS servers would speed up the process of fast-flux detection. No significant time was saved, which leads to fast detection of fast-flux domains. Table 12 summarizes the active approaches.

4.3.3 Real-Time Approaches

A real-time approach (Futai et al., 2013) was used to develop a fast-flux botnet detection method. This approach employs the J48 decision tree algorithm as a classifier in a hybrid system, which combines real-time detection and long-term monitoring, as depicted in Figure 15. Their approach can achieve a higher real-time detection rate compared with flux score-based methods. However, the proposed approach cannot detect fast-flux domains with high TTL values. Table 13 summarizes the real-time approaches.

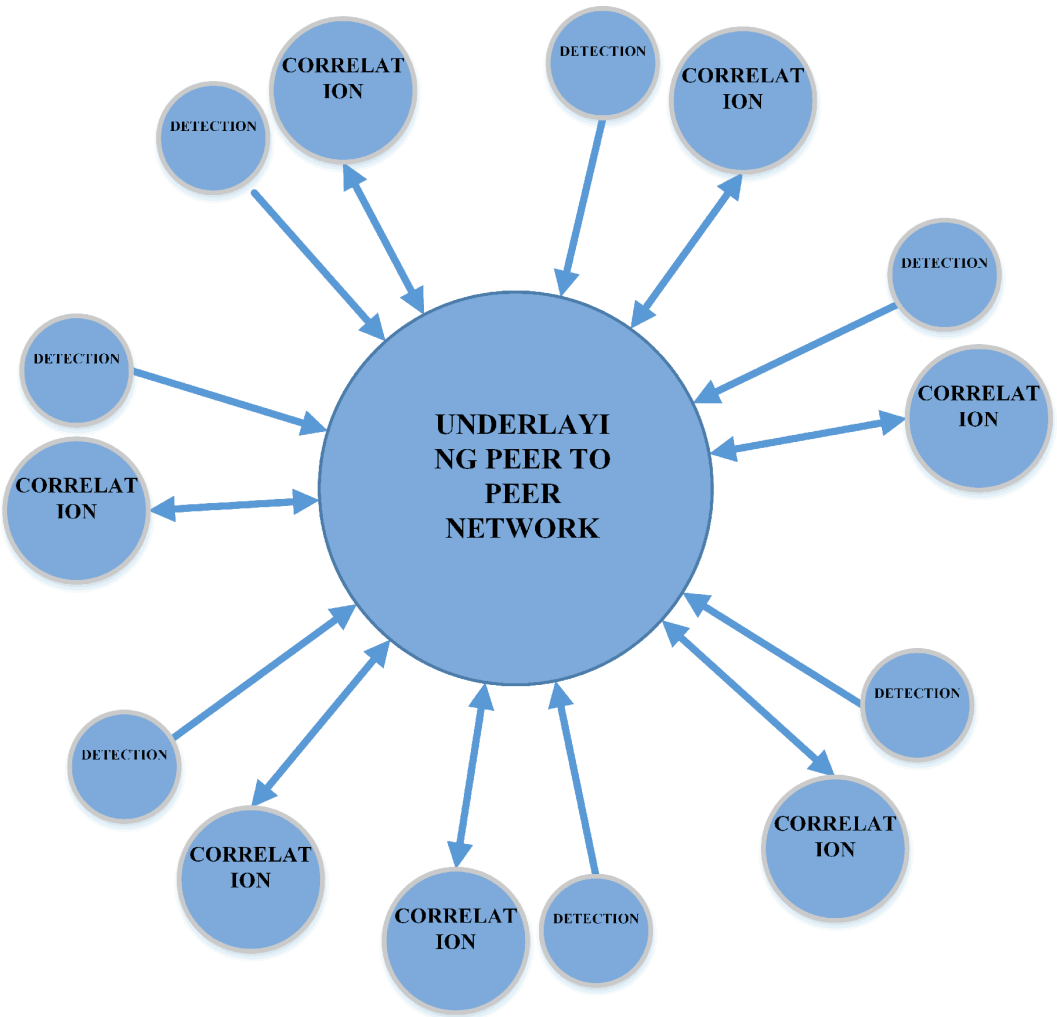
Table 11. Summary of passive dns-based detection approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Gržnić et al. (Gržnić et al., 2014)	Publicly available and private malware lists	Relies on the passive DNS replication method to detect suspicious fast-flux domains	Reduces FP	-Design problem -Unable to detect zero-day fast-flux domains
Kwon et al. (Kwon et al., 2016)	PSD	Leverages a signal processing technique to discover the major frequencies of periodic DNS queries of botnets	Detection of 23 unknown and 26 known botnet groups	-Increases in the number of hosts decrease the efficiency -The fixed threshold for all DNS servers
Perdisci et al. (Perdisci et al., 2009)	C4.5 decision tree	Detects malicious flux service networks through passive analysis of recursive DNS traces	Accurate classification	Cannot detect zero-day malicious flux services
Perdisci et al. (Perdisci et al., 2012)	C4.5 decision tree	A passive DNS traffic analysis system for detecting and tracking malicious flux networks	Detects unknown flux networks before blacklisting	-wait for a user click -The threshold needs sufficient IP addresses to be set
Vu Hong (Vu Hong, 2012)	Decision tree with AdaBoost algorithm	-Constructs a lookup graph and a failure graph from captured DNS traffic -Decomposes these graphs into clusters with a strong correlation between their domains, hosts, and IP addresses	Helps detect botnets through traffic analysis	-Produces FN and FP -Insufficient distinctive features
(Soltanaghaei & Kharrazi, 2015)	Sequential Probability Ration Testing (SPRT)	Two phases: 1. includes two sub-processes of traffic Parser” and \History Saver”. Then 2. the extracted domains are compared with those stored in the Bloom filter	Passively detect fast-flux domains with low false-positive rates	Not able to detect zero-day domains
(Lombardo et al., 2018)	Formula (static and history-based metrics)	Has been evaluated over the LAN of a company, with the injection of 47 pcaps associated with nine different malware campaigns that leverage FFSNs and cover a wide variety of attack scenarios.	Passively detect fast-flux domains with low false-positive rates	Not able to detect zero-day domains

Table 12. Summary of active approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Zhou et al. (C. V. Zhou et al., 2009)	Decentralized correlation model called LarSID	Correlation of multiple responses of DNS servers to increase detection time	Reduces query time up to 30%	Detection time is long

Figure 14. LarSID architecture (Zhou et al., 2009)



There are no network time delays in the detection systems launched via a DNS server. Many researchers found that DNS-driven systems cannot deliver an exact detection rate for fast-flux domains (Martinez-Bea et al., 2013).

The primary issue with fast-flux botnet detection is to detect an evasion screening mechanism before the attack, especially if zero-day domains are detected without any previous information on

Figure 15. Hybrid detection system (Futai et al., 2013)

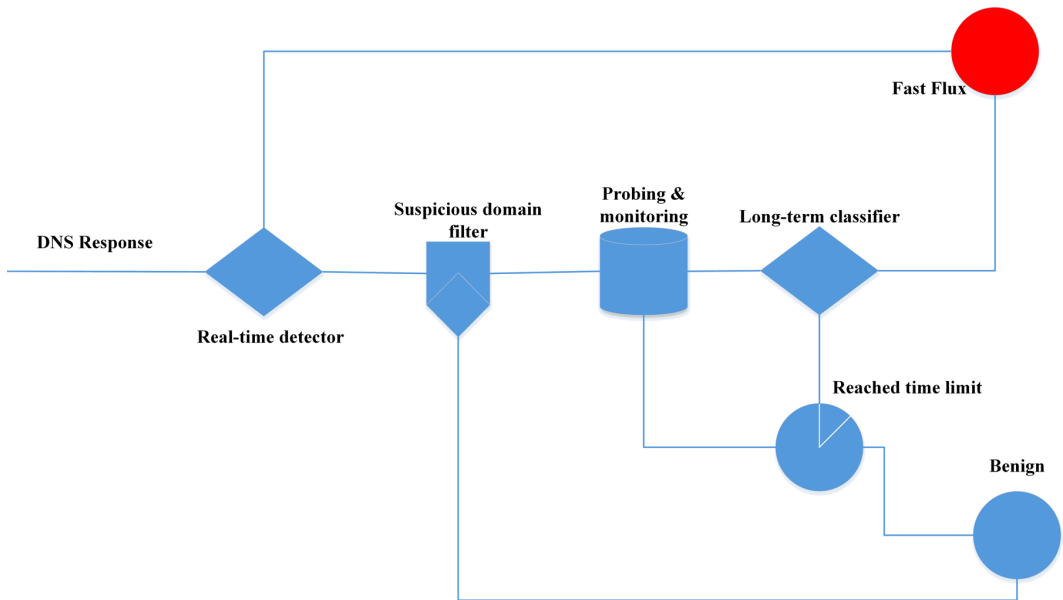


Table 13. Summary of real-time approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
Futai et al. (Futai et al., 2013)	J48 decision tree	Combines real-time detection and long-term monitoring	Higher detection rate compared with flux score-based algorithms	-Cannot detect FF domain with a high TTL value -Cannot detect unknown FFSNs

the new domain name that is available for malicious web sites / C2 servers/motherships. Detection precision and low error rates of detection are tracked simultaneously. The detection scheme must create innovative long-lasting and adaptive systems on the grounds of attackers ‘ approaches to enable their future functional changes. Detection systems should continuously learn by analyzing new system inputs as new data instead of training old data.

4.4 Hydra Flux Service Network

Hydra fast-flux networks and SQL injection attacks are the main advanced features of Asprox botnets. (Al-Bataineh & White, 2012) studied the design and structure of Asprox botnets, in which communication protocols are used to download malicious codes, propagate malicious codes, and employ hydra FFSNs. The authors mentioned that SQL injection attacks are responsible for the recruitment of new bots and social engineering ruses to spread malware binaries. Hydra FFSNs prevent the disruption of the communication channel between bots and the C&C server.

The crucial aspect of the hydra fast-flux is the possibility of bots contacting other C&C servers when the original C&C server is taken down. Hydra is an advanced double fast-flux that refluxes the name server and the host IP addresses, making the prevention of massive disruption impossible.

4.5 Neural Network Approaches

ANNs were used with other machine learning algorithms by (Statement before the Senate Judiciary Committee, 2014) to model a novel system of botnet detection that employs evasion techniques, such as fast-flux. This model is aimed at classifying malicious and non-malicious traffic. All the used algorithms are supervised. Thus, the proposed system cannot detect new and unknown malicious traffic.

Besides, neural networks, as one of the five machine learning algorithms, were used by (Wijesinghe et al., 2015) for botnet detection. The author proposed techniques to detect botnets with different bot families, such as HTTP botnets. The analyses of network traffic identified relevant attributes for botnet detection. The results of the analysis of fast-flux concepts indicate that bots carry out many DNS lookup instances to identify the C&C server, which produces patterns. These patterns were used in their techniques to track botnets. The authors believed that their techniques could detect different bot families at a low cost. However, their technique cannot detect bots, and their model is not formally trained; thus, their method cannot detect new and unknown threats (Hands et al., 2015).

A fast-flux detection method was proposed by (Al-Nawasrah et al., 2018), authors build an approach that detects fast-flux domains in real-time mode. Their approach was based on the adaptive dynamic evolving spiking neural network algorithm. The detection accuracy was above 98%, but there were too many parameters needed to be set before running the algorithm.

Another botnet detection framework (Bottazzi et al., 2015) is software-defined networking (SDN) (Wijesinghe et al., 2015). SDN was used to separate the control part from the content part of traffic at networking devices. A neural network with algorithms was employed to identify the best attribute based on flow similarities, which help in flow clustering. The authors used IPFIX for evolving customized templates to detect botnets. These templates can be customized to add new features from the network flow, such as DNS flows, to help fast-flux detection. This template must be configured manually. Thus, it cannot detect new zero-day domains, which exhibit features that are different from those of templates.

Neural networks with Bayesian regularization were proposed in (UK, 2015), with the goal of implementing a novel hybrid framework to detect P2P botnets. The authors mentioned the possible attacks of these botnets, such as fast-flux botnets. Integrating neural networks with Bayesian regularization enables the detection of botnet activities, as well as those that were not used in neural network training. Thus, the detection of new and unseen botnets in live network traffic is enhanced. Their tests indicate that the detection accuracy is high. ANNs have well-known limitations, such as the difficulties in selecting network structure. These networks do not monitor previously learned knowledge after additional training (Watts, 2004).

4.6 Approaches Applied in a Cloud Computing Environment

Researchers discussed different approaches while studying fast-flux defending proposed solutions. Could computing environment has part of researches effort (Almomani et al., 2019; B. Gupta et al., 2019; B. B. Gupta et al., 2017; S. Gupta et al., 2015), the current section presents the work have done based on application in a cloud computing environment.

(Al-Duwairi & Al-Hammouri, 2014) compared incoming and outgoing data traffic at a leaf router of stub networks to find matches between incoming and outgoing SYN packets. This online approach efficiently detects malicious fast-flux agents within stub networks. However, installing the system on all the leaf routers of stub networks is difficult to achieve (scalability problem), and the utilized data traffic traces do not have fast-flux traffic (Al-Duwairi & Al-Hammouri, 2014).

According to (Jiang et al., 2017), authors deployed Support Vector Machine, Naïve Bayes and K-Nearest Neighbors in their approach; they tried to identify a new class of FFSN architecture known as N-flux. This is happened by collecting the information by the digger module then store it in a database and is then processed by a data mining algorithm. However, as their system works passively; thus, it is not suitable for real-time applications.

(Jia et al., 2014) proposed a cloud-enabled, shuffling-based, moving target mechanism to mitigate DDoS attacks against open Internet services. New algorithms are implemented for optimizing runtime reassignment plans. The method uses the set of existing bots, which provides the intelligence of the migrating servers as the main measure for calculating the optimized' shuffling' pattern. However, it does not portray how the proxy node detects the attack. Overheads are based on the number of shuffles required and the size of the covered geographical region, which can be global.

A method of cloud nodes in a swarm network for DDoS mitigation presented in (Lua et al., 2011). The client needs a fully qualified domain name to reach the server. The client is sent to the server and sends its request via the community exit node to the identified server. The server responds, and the outcome is transmitted back to the client via the swarm network. The swarm network is available via fast-flux hosting and therefore highly robust. A parallel optimization algorithm like the Intelligent Water Drop system allows for continuous reconfiguration of the swarm network. This is all used to extend the cloud service's serviceability under DDoS attack. However, depending on the volatility of the network, rapid swarm-composition modifications can lead to listed servers of inactive names.

(Shetty, 2013) introduced cloud computing techniques to analyze network traffic. IP geolocation, IP router analysis and on-line data mining techniques. Technology. The three techniques are interconnected and necessary for evaluating the security of outsourced cloud data. The cloud data security is dependent on a trusted cloud and network computer system. The techniques described in this section provide insights into the cloud data security effect of the cloud network. IP geolocation enables the geographical positioning of routers in the network route between the cloud users and the providers to be determined. Analysis of Router IP offers a mechanism to analyze software risk on such routers. Finally, internet information mining provides a cloud traffic analysis strategy, while restricting misplaced favourable information from notion drifting streams. However, the size of cloud logs always poses challenges in log analysis (Kumar et al., 2019).

5. SUMMARIES AND DISCUSSIONS

This paper presents a comprehensive survey of fast-flux botnet detection approaches. Current detection techniques unsuccessfully detect fast-flux domains, particularly zero-day fast-flux domains. The summaries and comments on various detection measures from the literature survey according to the solution scope are as follows.

- DNS-based approaches

This type of technique focuses on detecting malicious fast-flux domains based on the monitoring of DNS servers and DNS responses. The problem with these approaches is that the features extracted from DNS features cannot provide high detection accuracy.

- Router-based approaches

Router-based detection techniques compare the DNS network traffic on both sides of the router to find matches between incoming and outgoing messages. They also identify malicious fast-flux agents, cluster network traffic, and identify malicious clusters. The problem with these approaches is the need for high detection speed and high memory storage based on the high volume and speed on the network traffic passing the router.

- Host-based approaches

Table 14. Summary of cloud computing approaches

Authors	Algorithms	Mechanism	Advantages	Disadvantages
(Al-Duwairi & Al-Hammouri, 2014)	FF-watch algorithm	Correlates incoming TCP connection requests to flux agents within a stub network with outgoing TCP connection requests from the same agents to the point-of-sale website	Eliminates the need for large DNS traffic	-An old dataset which may not contain FF traces -Scalability problem
(Jiang & Li, 2017)	Support Vector Machine, Naïve Bayes and K-Nearest Neighbors	The information collected by the digger module is stored in a database and is then processed by a data mining algorithm.	identify a new class of FFSN architecture known as N-flux	Not suitable for real-time applications
(Jia et al., 2014)	Maximum Likelihood Estimation (MLE) algorithm	This approach dynamically instantiates replica servers in the cloud and intelligently re-maps and migrates client sessions to new and un-advertised server locations.	Enables planned client-to-server shuffling operations to segregate the persistent attackers.	-Don't tell how the proxy node detects the attack. -The overheads of implementation depend on the number of shuffles required and the size of the covered geographical region.
(Lua & Yow, 2011)	fast-flux swarm network/ Water Drop mechanism	They presented a DDoS mitigation technique consisting of cloud nodes in a swarm network.	extend the serviceability of a cloud service under DDoS attack	-About the network's volatility. -fast swarm modifications may lead to listed inactive name servers
(Shetty, 2013)	SVM	presents technologies for network traffic analysis in a cloud computing environment	Limiting false positive from concept drifting streams.	Size of cloud logs always pose challenges in log analysis.

According to the literature, most of the works conducted to detect fast-flux botnets are host-based. Different approaches, such as passive, active, and real-time approaches, are used to detect fast-flux domains. However, fast-flux botnets must be detected in a short time because of the quick-changing IP addresses of the motherships. Thus, tracking their location is difficult. Detecting this type of “FF zero-day” domains as quickly as possible is crucial. These approaches deal with a large amount of data, which are not suitable for efficient processing with a few resources. Active detection-based approaches deal with less DNS traffic traces that correspond to non-legitimate domain names in most cases.

The proposed real-time approaches can detect malicious domain names and malicious FFSNs in most cases. However, the techniques mentioned above have certain limitations that cast doubt over their results (accuracy, TP, TN, FP, and FN). A stable technology is lacking for detecting malicious domains and malicious FFSNs, especially zero-day domains with a high detection precision over an acceptable period.

6. CONCLUSION

Botnets are the foundation for many security threats in cloud applications around the globe. C&C servers are the backbone of botnet communication. Botnets, such as IRC and P2P, are also categorized according to their C&C protocols. As DNS special method known as fast-flux is employed to cover malicious botnet activities and increase the lifetime of malicious servers (websites) by quickly changing the IP addresses of the domain names over time. Although several methods have been suggested for detecting fast-flux domains, these methods have shown the weakness of the detection accuracy, particularly for zero-day domains, exhibit long detection time, and consume high memory storage.

In this survey, an overview has been conducted of the various techniques that are used to detect fast-flux domains at different solution scopes whether they are implemented at Host, DNS, or Router side. This survey provides an understanding of the problem and its current solution space. So, give the researchers a solid background of the proposed solution scope advantages and disadvantages. As a result, Fast-flux detection approaches should be improved. In addition, the detection systems have to be enhanced to develop long-term and adaptive new technologies that have modifiable features. Such systems should also continuously learn from new inputs instead of training on old data. These steps are crucial to overcome future challenges, particularly zero-day cloud domains. Finally, the authors believe that this paper succeeded in perfectly exploring each of the mentioned solution scopes in details, so researchers could target the proper scope of the solution to improve intelligent new ideas.

REFERENCES

- Al-Bataineh, A., & White, G. (2012). Analysis and detection of malicious data exfiltration in web traffic. *Paper presented at the 2012 7th International Conference on Malicious and Unwanted Software*. Academic Press. doi:10.1109/MALWARE.2012.6461004
- Al-Duwairi, B. N., & Al-Hammouri, A. T. (2014). Fast Flux Watch: A mechanism for online detection of fast flux networks. *Journal of Advanced Research*, 5(4), 473–479. doi:10.1016/j.jare.2014.01.002 PMID:25685515
- Al-Fayoumi, M., Alwidian, J., & Abusaif, M. (2019). Intelligent Association Classification Technique for Phishing Website Detection. *The International Arab Journal of Information Technology*, 17.
- Al-Nawasrah, A., Al-Momani, A., Meziane, F., & Alauthman, M. (2018). Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm. *Paper presented at the 2018 9th International Conference on Information and Communication Systems (ICICS)*. Academic Press. doi:10.1109/IACS.2018.8355433
- Alauthman, M., Aslam, N., Zhang, L., Alasem, R., & Hossain, M. A. (2018). A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks. *Neural Computing & Applications*, 29(11), 991–1004. doi:10.1007/s00521-016-2564-5 PMID:29769759
- Alauthman, M., Aslam, N., Al-kasassbeh, M., Khan, S., Al-Qerem, A., & Raymond Choo, K.-K. (2020). An efficient reinforcement learning-based Botnet detection approach. *Journal of Network and Computer Applications*, 150, 102479. doi:10.1016/j.jnca.2019.102479
- Alieyan, K., Almomani, A., Anbar, M., Alauthman, M., Abdullah, R., & Gupta, B. B. (2019). DNS rule-based schema to botnet detection. *Enterprise Information Systems*. doi:10.1080/17517575.2019.1644673
- Alieyan, K., AlMomani, A., Manasrah, A., & Kadhum, M. M. (2015). A survey of botnet detection based on DNS. *Neural Computing & Applications*, 28(7), 1541–1558. doi:10.1007/s00521-015-2128-0
- Almomani, A. (2016). Fast-flux hunter: A system for filtering online fast-flux botnet. *Neural Computing & Applications*, 29(7), 483–493. doi:10.1007/s00521-016-2531-1
- Almomani, A., Alauthman, M., Albalas, F., Dorgham, O., & Obeidat, A. (2018). An Online Intrusion Detection System to Cloud Computing Based on Neucube Algorithms. *International Journal of Cloud Applications and Computing*, 8(2), 96–112. doi:10.4018/IJCAC.2018040105
- Almomani, A., Alauthman, M., Alweshah, M., Dorgham, O., & Albalas, F. (2019). A comparative study on spiking neural network encoding schema: Implemented with cloud computing. *Cluster Computing*, 22(2), 419–433. doi:10.1007/s10586-018-02891-0
- Almomani, A., Gupta, B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys and Tutorials*, 15(4), 2070–2090. doi:10.1109/SURV.2013.030713.00020
- Almomani, A., Obeidat, A., Alsaedi, K., Obaida, M. A.-H., & Al-Betar, M. (2015). Spam E-mail Filtering using ECOS Algorithms. *Indian Journal of Science and Technology*, 8(S9), 260–272. doi:10.17485/ijst/2015/v8iS9/55320
- Almomani, A., Wan, T.-C., Manasrah, A., Altaher, A., Baklizi, M., & Ramadass, S. (2013). An enhanced online phishing e-mail detection framework based on evolving connectionist system. *International Journal of Innovative Computing, Information, & Control*, 9(3), 169–175.
- Alomari, E., Manickam, S., Gupta, B., Anbar, M., Saad, R. M., & Alsaleem, S. (2016). A survey of botnet-based ddos flooding attacks of application layer: Detection and mitigation approaches *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security* (pp. 52–79). IGI Global. doi:10.4018/978-1-5225-0105-3.ch003
- Alomari, E., Manickam, S., Gupta, B., Singh, P., & Anbar, M. (2014). Design, deployment and use of HTTP-based botnet (HBB) testbed. *Paper presented at the 16th International Conference on Advanced Communication Technology*. Academic Press. doi:10.1109/ICACT.2014.6779162

- Barford, P., & Yegneswaran, V. (2007). *An inside look at botnets*. In *Malware Detection* (pp. 171–191). Springer. doi:10.1007/978-0-387-44599-1_8
- Bottazzi, G., & Me, G. (2015). *A Survey on Financial Botnets Threat*. In *Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security* (pp. 172–181). Springer.
- Buhariwala, K. (2011). Geo-locating Hidden Servers Behind Fast-Flux Proxies.
- Burghouwt, P. (2015). Detection of Botnet Command and Control Traffic in Enterprise Networks: TU Delft, Delft University of Technology.
- Cafuta, D., Sruk, V., & Dodig, I. (2018). Fast-Flux Botnet Detection Based on Traffic Response and Search Engines Credit Worthiness. *Tehnicki Vjesnik (Strojarski Fakultet)*, 25(2), 390–400.
- Caglayan, A., Toothaker, M., Drapaeau, D., Burke, D., & Eaton, G. (2010). Behavioral Patterns of Fast Flux Service Networks. *Paper presented at the 2010 43rd Hawaii International Conference on System Sciences (HICSS)*. IEEE Press. doi:10.1109/HICSS.2010.81
- Caglayan, A., Toothaker, M., Drapeau, D., Burke, D., & Eaton, G. (2009). Real-Time Detection of Fast Flux Service Networks. *Paper presented at the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*. Academic Press. doi:10.1109/CATCH.2009.44
- Castelluccia, C., Kaafar, M. A., Manils, P., & Perito, D. (2009). Geolocalization of proxied services and its application to fast-flux hidden servers. *Paper presented at the 9th ACM SIGCOMM conference on Internet measurement*. ACM Press. doi:10.1145/1644893.1644915
- Celik, Z. B., & Oktug, S. (2013). Detection of fast-flux networks using various dns feature sets. *Paper presented at the 2013 IEEE Symposium on Computers and Communications (ISCC)*. IEEE Press. doi:10.1109/ISCC.2013.6755058
- Chen, C.-M., Cheng, S.-T., & Chou, J.-H. (2013). Detection of Fast-Flux Domains. *Journal of Advances in Computer Networks*, 1(2), 148–152. doi:10.7763/JACN.2013.V1.30
- Chen, C. M., Huang, M. Z., & Ou, Y. H. (2014). Detecting hybrid botnets with web command and control servers or fast flux domain. *Journal of Information Hiding and Multimedia Signal Processing*, 5(2), 262273.
- Chen, Z., Wang, J., Zhou, Y., & Li, C. (2011). An improvement for fast-flux service networks detection based on data mining techniques. *Paper presented at the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Academic Press. doi:10.1007/978-3-642-21881-1_47
- Committee. (2014). Subcommittee on Crime and Terrorism.
- Emre, Y. (2011). *A literature survey about recent botnet trends* (pp. 1–14). GEANT.
- HP Enterprise. (2015). 2015 Cost of Cyber Crime Study: Global. Retrieved from http://engage.hpe.com/LP_510004609_HPSW-ESP_WW_EN-US_PonemonGate
- Fabian, M., & Terzis, M. A. (2007). My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. *Paper presented at the 1st USENIX Workshop on Hot Topics in Understanding Botnets*. Academic Press.
- FFA UK. (2015). *Fraud the Facts 2015: The definitive overview of payment industry fraud and measures to prevent it*. London: UK Cards Association.
- Futai, Z., Siyu, Z., & Weixiong, R. (2013). Hybrid detection and tracking of fast-flux botnet on domain name system traffic. *Communications, China*, 10(11), 81–94. doi:10.1109/CC.2013.6674213
- Gasster, L. (2008). GNSO issues report on fast flux hosting. Retrieved from https://gnso.icann.org/sites/default/files/filefield_5868/gnso-issues-report-fast-flux-25mar08.pdf
- Gothai, E., & Balasubramanie, P. (2012). An Efficient Way for Clustering Using Alternative Decision Tree. *American Journal of Applied Sciences*, 9(4), 531–534. doi:10.3844/ajassp.2012.531.534
- Grizzard, J. B., Sharma, V., Nunnery, C., Kang, B. B., & Dagon, D. (2007). Peer-to-peer botnets: Overview and case study. *Paper presented at the 1st USENIX Workshop on Hot Topics in Understanding Botnets*. Academic Press.

Gržnić, T., Perhoč, D., Marić, M., Vlašić, F., & Kulcsar, T. (2014). CROFlux—Passive DNS method for detecting fast-flux domains. *Paper presented at the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Academic Press.

Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008). BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. *Paper presented at the USENIX Security Symposium*. Academic Press.

Guo, Z., & Guan, Y. (2018). Active Probing-Based Schemes and Data Analytics for Investigating Malicious Fast-Flux Web-Cloaking Based Domains. *Paper presented at the 2018 27th International Conference on Computer Communication and Networks (ICCCN)*. Academic Press. doi:10.1109/ICCCN.2018.8487410

Gupta, B. (2011). *An Introduction to DDoS Attacks and Defense Mechanisms: An Analyst's Handbook*. Lap Lambert Academic Pub.

Gupta, B., & Agrawal, D. P. (2019). *Handbook of Research on Cloud Computing and Big Data Applications in IoT*. Hershey, PA: IGI Global. doi:10.4018/978-1-5225-8407-0

Gupta, B. B., Gupta, S., & Chaudhary, P. (2017). Enhancing the browser-side context-aware sanitization of suspicious HTML5 code for halting the DOM-based XSS vulnerabilities in cloud. *International Journal of Cloud Applications and Computing*, 7(1), 1–31. doi:10.4018/IJCAC.2017010101

Gupta, S., & Gupta, B. (2015). BDS: browser dependent XSS sanitizer. In *Handbook of Research on Securing Cloud-Based Databases with Biometric Applications* (pp. 174–191). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-6559-0.ch008

Hands, N. M., Yang, B., & Hansen, R. A. (2015). A Study on Botnets Utilizing DNS. *Paper presented at the 4th Annual ACM Conference on Research in Information Technology*. ACM Press. doi:10.1145/2808062.2808070

Holz, T., Gorecki, C., Rieck, K., & Freiling, F. C. (2008). Measuring and Detecting Fast-Flux Service Networks. *Paper presented at the NDSS*. Academic Press.

Horng-Tzer, W., Ching-Hao, M., Kuo-Ping, W., & Hahn-Ming, L. (2012). Real-Time Fast-Flux Identification via Localized Spatial Geolocation Detection. *Paper presented at the 2012 IEEE 36th Annual Computer Software and Applications Conference*. IEEE Press.

Hsu, C.-H., Huang, C.-Y., & Chen, K.-T. (2010). Fast-Flux Bot Detection in Real Time. *Paper presented at the International Workshop on Recent Advances in Intrusion Detection*. Academic Press. doi:10.1007/978-3-642-15512-3_24

Hsu, F.-H., Wang, C.-S., Hsu, C.-H., Tso, C.-K., Chen, L.-H., & Lin, S.-H. (2014). Detect fast-flux domains through response time differences. *IEEE Journal on Selected Areas in Communications*, 32(10), 1947–1956. doi:10.1109/JSAC.2014.2358814

Huang, S.-Y., Mao, C.-H., & Lee, H.-M. (2010). Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection. *Paper presented at the 5th ACM Symposium on Information, Computer and Communications Security*. ACM. doi:10.1145/1755688.1755702

Jia, Q., Wang, H., Fleck, D., Li, F., Stavrou, A., & Powell, W. (2014). Catch me if you can: A cloud-enabled DDoS defense. *Paper presented at the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. Academic Press. doi:10.1109/DSN.2014.35

Jiang, C.-B., & Li, J.-S. (2017). Exploring Global IP-Usage Patterns in Fast-Flux Service Networks. *JCP*, 12(4), 371–379. doi:10.17706/jcp.12.4.371-379

Jiang, N., Cao, J., Jin, Y., Li, L. E., & Zhang, Z.-L. (2010). Identifying suspicious activities through dns failure graph analysis. *Paper presented at the 2010 18th IEEE International Conference on Network Protocols (ICNP)*. IEEE Press. doi:10.1109/ICNP.2010.5762763

Kalige, E., & Burkey, D. (2012). A case study of eurograbber: How 36 million euros was stolen via malware. Versafe.

Karasaridis, A., Rexroad, B., & Hoeflin, D. (2007). Wide-scale botnet detection and characterization. *Paper presented at the Usenix Workshop on Hot Topics in Understanding Botnets*. Academic Press.

- Karim, A., Salleh, R. B., Shiraz, M., Shah, S. A. A., Awan, I., & Anuar, N. B. (2014). Botnet detection techniques: Review, future trends, and issues. *Journal of Zhejiang University Science C*, 15(11), 943–983. doi:10.1631/jzus.C1300242
- Kevin McGrath, D., & Minaxi Gupta, A. K. (2009). Phishing Infrastructure Fluxes All the Way. *IEEE Security and Privacy*, 7(5), 21–28. doi:10.1109/MSP.2009.130
- Konings, M. (2009). Final report of the gnso fast flux hosting working group: Internet Corporation for Assigned Names and Numbers–Generic Names Supporting.
- Kumar, R., & Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. *Computer Science Review*, 33, 1–48. doi:10.1016/j.cosrev.2019.05.002
- Kwon, J., Lee, J., Lee, H., & Perrig, A. (2016). PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks*, 97, 48–73. doi:10.1016/j.comnet.2015.12.008
- Lee, W., Wang, C., & Dagon, D. (Eds.). (2007). *Botnet detection: countering the largest security threat*. Springer Science & Business Media.
- Levy, E., & Arce, I. (2005). A Short Visit to the Bot Zoo. *IEEE Security and Privacy*, 3(3), 76–79. doi:10.1109/MSP.2005.58
- Lin, H.-T., Lin, Y.-Y., & Chiang, J.-W. (2013). Genetic-based real-time fast-flux service networks detection. *Computer Networks*, 57(2), 501–513. doi:10.1016/j.comnet.2012.07.017
- Lombardo, P., Saeli, S., Bisio, F., Bernardi, D., & Massa, D. (2018). Fast Flux Service Network Detection via Data Mining on Passive DNS Traffic. *Paper presented at the International Conference on Information Security*. Academic Press. doi:10.1007/978-3-319-99136-8_25
- Losses, N. (2014). *Estimating the Global Cost of Cybercrime*. McAfee.
- Lua, R., & Yow, K. C. (2011). Mitigating ddos attacks with transparent and intelligent fast-flux swarm network. *IEEE Network*, 25(4), 28–33. doi:10.1109/MNET.2011.5958005
- Marcus, R.S., & D., (2012). *Dissecting operation high roller*. McAfee.
- Martinez-Bea, S., Castillo-Perez, S., & Garcia-Alfaro, J. (2013). Real-time malicious fast-flux detection using DNS and bot related features. *Paper presented at the PST*. Academic Press. doi:10.1109/PST.2013.6596093
- Trend Micro. (2014). New Zeus Gameover Employs DGA and Fast Flux Techniques. Retrieved from <https://www.trendmicro.com/vinfo/us/threat-encyclopedia/spam/578/new-zeus-gameover-employs-dga-and-fast-flux-techniques>
- Otgonbold, T. (2014). ADAPT: An anonymous, distributed, and active probing-based technique for detecting malicious fast-flux domains. Iowa State University. Retrieved from <https://lib.dr.iastate.edu/etd/14225>
- Pa, Y. M. P., Yoshioka, K., & Matsumoto, T. (2015). Detecting malicious domains and authoritative name servers based on their distinct mappings to IP addresses. *Journal of information processing*, 23(5), 623–632.
- Pappas, V., Wessels, D., Massey, D., Lu, S., Terzis, A., & Zhang, L. (2009). Impact of configuration errors on DNS robustness. *IEEE Journal on Selected Areas in Communications*, 27(3), 275–290. doi:10.1109/JSAC.2009.090404
- Passerini, E., Paleari, R., Martignoni, L., & Bruschi, D. (2008). *Fluxor: Detecting and monitoring fast-flux service networks*. In *Detection of intrusions and malware, and vulnerability assessment* (pp. 186–206). Springer.
- Paul, T., Tyagi, R., Manoj, B., & Thanudas, B. (2014). Fast-flux botnet detection from network traffic. *Paper presented at the 2014 Annual IEEE India Conference (INDICON)*. IEEE Press. doi:10.1109/INDICON.2014.7030393
- Perdisci, R., Corona, I., Dagon, D., & Lee, W. (2009). Detecting malicious flux service networks through passive analysis of recursive DNS traces. *Paper presented at the 2009 Annual Computer Security Applications Conference*. Academic Press. doi:10.1109/ACSAC.2009.36
- Perdisci, R., Corona, I., & Giacinto, G. (2012). Early detection of malicious flux networks via large-scale passive DNS traffic analysis. *IEEE Transactions on Dependable and Secure Computing*, 9(5), 714–726.

- Qassrawi, M. T., & Zhang, H. L. (2012). Detecting Malicious Fast Flux Domains. *Paper presented at the Applied Mechanics and Materials*. Academic Press. doi:10.4028/www.scientific.net/AMM.157-158.1264
- Rajab, M. A., Zarfoss, J., Monroe, F., & Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. *Paper presented at the 6th ACM SIGCOMM conference on Internet measurement*. ACM. doi:10.1145/1177080.1177086
- Scharrenberg, P. (2008). *Analyzing Fast-Flux Service Networks* [Dissertation]. RWTH Aachen University, Germany.
- Shaikh, A., Tewari, R., & Agrawal, M. (2001). On the effectiveness of DNS-based server selection. *Paper presented at the 20th IEEE International Conference on Computer Communications (INFOCOM 2001)*. IEEE Press. doi:10.1109/INFCOM.2001.916678
- Sheng, Y., Shijie, Z., & Sha, W. (2010). Fast-flux attack network identification based on agent lifespan. *Paper presented at the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*. IEEE Press. doi:10.1109/WCINS.2010.5541861
- Shetty, S. (2013). Auditing and analysis of network traffic in cloud environment. *Paper presented at the 2013 IEEE Ninth World Congress on Services*. IEEE Press. doi:10.1109/SERVICES.2013.42
- Soltanaghahi, E., & Kharrazi, M. (2015). Detection of fast-flux botnets through DNS traffic analysis. *Scientia Iranica. Transaction D, Computer Science & Engineering, Electrical*, 22(6), 2389-2400.
- Sood, A. K., & Enbody, R. J. (2013). Crimeware-as-a-service—A survey of commoditized crimeware in the underground market. *International Journal of Critical Infrastructure Protection*, 6(1), 28–38. doi:10.1016/j.ijcip.2013.01.002
- SSAC. (2008). SAC 025: SSAC Advisory on Fast Flux Hosting and DNS.
- Stalmans, E., Hunter, S. O., & Irwin, B. (2012). Geo-spatial autocorrelation as a metric for the detection of Fast-Flux botnet domains. *Paper presented at the Information Security for South Africa (ISSA)*. Academic Press. doi:10.1109/ISSA.2012.6320433
- Stevanovic, M., & Pedersen, J. M. (2013). Machine learning for identifying botnet network traffic. Aalborg University.
- Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., . . . Vigna, G. (2009). Your botnet is my botnet: analysis of a botnet takeover. *Paper presented at the 16th ACM conference on Computer and communications security*. ACM. doi:10.1145/1653662.1653738
- Stornig, F. (2013). *Detection of Botnet Fast-Flux Domains by the aid of spatial analysis methods*. Carinthia University of Applied Sciences.
- Su, M.-Y., & Tsai, C.-H. (2012). A prevention system for spam over Internet telephony. *Applications of Mathematics*, 6(2S), 579S–585S.
- Villamarín-Salomón, R., & Brustoloni, J. C. (2009). Bayesian bot detection based on DNS traffic similarity. *Paper presented at the 2009 ACM symposium on Applied Computing*. ACM Press. doi:10.1145/1529282.1529734
- Vu Hong, L. (2012). *DNS Traffic Analysis for Network-based Malware Detection*. Sweden: KTH Royal Institute of Technology. Retrieved from <http://www.diva-portal.org/smash/get/diva2:524298/FULLTEXT01.pdf>
- Watts, M. J. (2004). *Evolving connectionist systems: Characterisation, simplification, formalisation, explanation and optimisation [PhD dissertation]*. University of Otago. Retrieved from <https://ourarchive.otago.ac.nz/bitstream/handle/10523/1489/MikeWattsthesis.pdf?sequence=5>
- Wijesinghe, U., Tupakula, U., & Varadharajan, V. (2015). Botnet detection using software defined networking. *Paper presented at the 2015 22nd International Conference on Telecommunications (ICT)*. Academic Press. doi:10.1109/ICT.2015.7124686
- Xu, W., Wang, X., & Xie, H. (2013). New Trends in FastFlux Networks. *Paper presented at the 16th BlackHat*. Academic Press.

- Yadav, S., Reddy, A. K. K., Reddy, A., & Ranjan, S. (2010). Detecting algorithmically generated malicious domain names. *Paper presented at the 10th ACM SIGCOMM conference on Internet measurement*. ACM Press.
- Yu, B., Smith, L., & Threefoot, M. (2014). Semi-supervised Time Series Modeling for Real-Time Flux Domain Detection on Passive DNS Traffic. *Paper presented at the International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Academic Press. doi:10.1007/978-3-319-08979-9_20
- Yu, S. (2014). *Malicious Networks for DDoS Attacks*. In *Distributed Denial of Service Attack and Defense* (pp. 15–29). Springer. doi:10.1007/978-1-4614-9491-1_2
- Yu, X., Zhang, B., Kang, L., & Chen, J. (2012). Fast-Flux Botnet Detection Based on Weighted SVM. *Information Technology Journal*, 11(8), 1048–1055. doi:10.3923/itj.2012.1048.1055
- Yukonhiatou, C., Kittitornkun, S., Kikuchi, H., Sisaat, K., Terada, M., & Ishii, H. (2014). Temporal behaviors of Top-10 malware download in 2010–2012. *Paper presented at the 2014 International Electrical Engineering Congress (iEECON)*. Academic Press.
- Zhang, L., Yu, S., Wu, D., & Watters, P. (2011). A survey on latest botnet attack and defense. *Paper presented at the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE Press. doi:10.1109/TrustCom.2011.11
- Zhao, D., & Traore, I. (2012). P2P botnet detection through malicious fast flux network identification. *Paper presented at the 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*. Academic Press.
- Zhao, Y., & Jin, Z. (2015). Quickly Identifying FFSN Domain and CDN Domain with Little Dataset. *Paper presented at the 2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering*. Academic Press. doi:10.2991/icmmce-15.2015.386
- Zhou, C. V., Leckie, C., & Karunasekera, S. (2009). Collaborative detection of fast flux phishing domains. *Journal of Networks*, 4(1), 75–84. doi:10.4304/jnw.4.1.75-84
- Zhou, S. (2015). A Survey on Fast-flux Attacks. *Information Security Journal: A Global Perspective*, 24(4-6), 79-97.
- Zou, F., Li, L., Wu, Y., Li, J., Zhang, S., & Jiang, K. (2018). Detecting domain-flux malware using DNS failure traffic. *International Journal of Software Engineering and Knowledge Engineering*, 28(02), 151–173. doi:10.1142/S0218194018400016

Ahmad Al-Nawasrah is an Assistant professor in Computer Science/ Information Security. I have graduated from the University of Salford, UK, 2018. Currently, I'm working at Taibah University. My research interest is in cyber security, Internet security, IoT Security, and Information security.

Almmar ali Almomani received a PhD degree from Universiti Sains Malaysia (USM) in 2013. He has published more than 60 research papers in international journals and conferences of high repute. Currently, he is an assistant professor and senior lecturer at the Dept. of Information Technology, Al-Huson University College, Al-Balqa Applied University, Jordan. His research interest includes advanced Internet security and monitoring cloud computing.

Samer Atawneh received his PhD Degree in information security from Universiti Sains Malaysia (USM) in 2015. Currently, Dr. Atawneh is an Assistant Professor at College of Computing and Informatics, Saudi Electronic University, Saudi Arabia. He has published several research papers in International Journals and Conferences with high reputation, where some of these publications are tracked by Thomson Reuters (ISI) and Scopus. His research interests lie in information security, steganography, software engineering, and mobile applications.

Mohammad Alauthman received his PhD degree from Northumbria University Newcastle, UK in 2016. He received a B.Sc. degree in Computer Science from Hashemite University, Jordan, in 2002, and received M.Sc. degrees in Computer Science from Amman Arab University, Jordan, in 2004. Currently, he is an Assistant Professor and senior lecturer at Department of Computer Science, Zarqa University, Jordan. His main research areas cyber-security, cyber forensics, advanced machine learning and data science applications.