FluXOR: detecting and monitoring fast-flux service networks

Emanuele Passerini, Roberto Paleari, Lorenzo Martignoni, and Danilo Bruschi

Università degli Studi di Milano
{ema,roberto,lorenzo,bruschi}@security.dico.unimi.it

Abstract *Botnets* are large groups of compromised machines (*bots*) used by miscreants for the most illegal activities (e.g., sending spam emails, denial-of-service attacks, phishing and other web scams). To protect the identity and to maximise the availability of the core components of their business, miscreants have recently started to use *fast-flux service networks*, large groups of bots acting as front-end proxies to these components. Motivated by the conviction that prompt detection and monitoring of these networks is an essential step to contrast the problem posed by botnets, we have developed FluXOR, a system to detect and monitor fast-flux service networks. FluXOR monitoring and detection strategies entirely rely on the analysis of a set of features observable from the point of view of a victim of the scams perpetrated thorough botnets. We have been using FluXOR for about a month and so far we have detected 387 fast-flux service networks, totally composed by 31998 distinct compromised machines, which we believe to be associated with 16 botnets.

1 Introduction

A malware is a program written with malicious intents. Today, the main motivation behind malware writing and their use is the easy financial gain. Smart miscreants write malware and sell them in the wealthy underground market to other miscreants [1]. These malicious programs are "installed" on machines all around the world, without any permission of the users, and transform these machines into *bots*, i.e., hosts completely under to control of the attackers. Bots are then used to steal computational resources and confidential information, to relay spam email messages, to mount distributed denial of service (DDoS) and other attacks, to host phishing websites, and for other kinds of scams. To maximise the profit from these activities, multiple "infected" machines are grouped together in a *botnet* (a network of bots) and used simultaneously to achieve the same purpose [2]. With a single command, miscreants can control hundreds or even thousands of bots [3]. The botnet problem is so extensive nowadays that it has made headlines several times [4,5].

The most well known botnets are those related with the Warezov and the Storm worms [6,7]. These botnets are infamous for the huge amount of spam emails they have been generating, often containing links to malicious web servers hosting various frauds as well as malicious web pages able to infect the machines

of the visitors with malware. Of particular interest is the technique used by those botnets to masquerade the identity of the malicious web servers in order to maximise the availability of the service. If these web servers are difficult to identify, they are difficult to shutdown, and they can hit more and more victims. This technique, known as *fast-flux service network*, is very simple and consists in associating the canonical hostname of a malicious web server (e.g., `www.factvillage.com`) with multiple IP addresses corresponding to the addresses of a subset of the bots of the botnet. Each victims' request to visit the web server will thus reach one of the bots and the bot will proxy the request to the real server, making impossible to discover the identity of the malicious web server without having full control of one of these bots. The association between the hostname of the web server and the IP addresses of the bots acting as front-end proxies is updated very frequently such that newly compromised machines can immediately take part in the game and dead bots are excluded without affecting the availability of the service [8].

The impact that botnets using fast-flux service networks have on the Internet community is tremendous [9]. Although the average lifetime of domains used for malicious purposes, including the domains associated with fast-flux service networks, is very short, the lifetime of botnets using those domains is much longer. As the identity of the hosts associated with those domains is well protected and the bots that are part of the networks are difficult to track, botnets are difficult to eradicate. Authorities put a lot of efforts to take down the domains registered for malicious purposes, but these efforts are worthless because the bots are not isolated. Before the domain is suspended, a new one is registered and associated with the same set of bots, to replace the old one. Consequently, miscreants can continue their malicious activity through their botnets without interruption.

The natural approach to monitor and detect botnets activity and the bots involved is to passively analyse the network traffic. Unfortunately, that requires the access to a significant network segment [10,11,12,13,14,15,16]. Fast-flux service networks are interesting from the research point of view because they allow to "observe" the botnet phenomenon from a completely different prospective, the prospective of a victim of the botnet. In fact, the visibility a victim has on the botnet is quite significant. More precisely, imagine a recidivous victim that visits very frequently a malicious web site associated with a botnet and served through a fast-flux service network. At each visit the victim is likely to access the web site through a different bot (recall that the canonical hostname of the web server is resolved into the IP address of one of the bots). After a large number of visits, the recidivous victim will have discovered the IP addresses of the majority of the active bots of the botnet.

This paper presents FluXOR, the system we have developed to detect and monitor fast-flux service networks. Given a suspicious hostname, FluXOR, by behaving like a recidivous victim, tries to detect if the hostname conceals a fast-flux service network. Hostnames associated with fast-flux service networks are then continuously monitored to find out all the IP addresses of the compromised machines that are part of the botnet associated with the service network itself. FluXOR detection strategy is based on the combined analysis of nine distinguish-

ing features describing some properties of (I) the domain the suspicious hostname belongs to, (II) the degree of availability of the potential fast-flux service network, and (III) the heterogeneity of the potential hosts of the network.

We have been using FluXOR since the beginning of January 2008 to monitor potential fast-flux service network whose hostnames were collected from spam emails. So far the system correctly classified all the analysed hostnames (4961) and 7.8% of them (387) turned out to be associated with fast-flux service networks, involving 31998 distinct compromised machines located all around the world. Real-time results of the analysis are available on-line at `http://fluxor.laser.dico.unimi.it`.

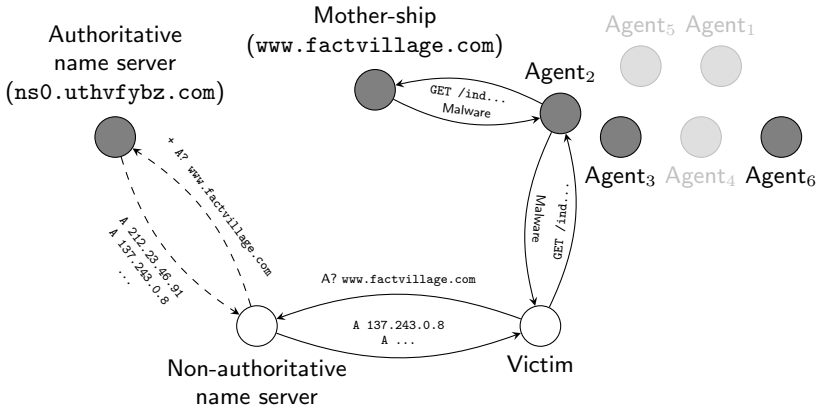
To summarise, this paper makes the following contributions:

- identification of the features that, combined together, allow to precisely detect whether or not a suspicious hostname conceals a fast-flux service network (Section 3 and 4).
- Implementation of a strategy to monitor a fast-flux service network and to detect the majority of the bots that are in the network (Section 5).
- Empirical analysis of the fast-flux service network phenomenon (Section 6).

2 Problem description and solution overview

A *fast-flux service network* is a network of compromised hosts that is used to carry out malicious activities, for example to deliver malware to users, to distribute illegal materials or to steal users' credentials [8]. The service network is identified by one or more fully qualified domain names (FQDNs) that are resolved to multiple (hundreds or even thousands) different IP addresses, belonging to unaware compromised hosts, the *fast-flux agents* (or *bots*). The fundamental characteristic of a fast-flux service network is *high availability*, which is provided by continuously updating the pool of agents serving the network. Newly compromised hosts are inserted into the network, inactive or unreliable hosts are removed, and victims are always redirected to the active and most reliable agents. The key is a combination of a very short time-to-live (TTL) of the DNS resource records that associate the canonical name of the service network with the set of IP addresses of the agents and a round-robin selection of these records [17,18]. In the common setup, the agents do not carry out the malicious activities, but they simply redirect received requests to the *fast-flux mother-ship*, the controlling element of the network, whose identity must be kept secret. With this setup, it is not possible to identify the mother-ship without having complete control of one of the agents.

Imagine that the fully qualified domain name `www.factvillage.com` conceals a fast-flux service network composed of hundreds of agents and that it is used to attract users, with the promise of very cheap drugs, and to infect their machines with malware. Figure 1 shows how our sample malicious contents provider leverages the fast-flux service network to serve the victims. A victim, wishing to visit the on-line drugstore, queries a name server (usually a non-authoritative name server which recursively queries the authoritative one) to resolve the hostname of the website. The name server returns the addresses of a subset of the agents



Average time-to-live of DNS resource records: 300 s.
Number of distinct IP addresses in the network: 253.

Figure 1. An example of the fast-flux service network used by our sample malicious web server `www.factvillage.com`, the entities involved and the communication between these entities (nodes in gray denote hosts under the control of the miscreants and shaded agents denote those that are not currently serving the network).

currently active in the network, and the victim connects to one of them. The agent then proxies the victim's requests to the mother-ship, which in turn delivers the malicious contents. In background, the mother-ship, or another entity controlled by the miscreants, continuously monitors the status of the agents and updates the resource records of the authoritative name server of the domain (in the example the authoritative name server is `ns0.uthvfybz.com`), to distribute the network across the reliable agents. The short time-to-live associated to the DNS resource records prevents non-authoritative name servers to cache for too long the records that define the subset of agents currently serving the network. When the cache expires the name server contacts again the authoritative name server for the domain and gets the new list of agents serving the network. These agents are selected from the set of all active agents in a round-robin fashion, to balance their load.

Our goals are, given a fully qualified domain name, to verify whether it conceals a fast-flux service network and, in such a case, to identify all the agents that are part of the network. The prompt identification and isolation of all the agents is important because if the service network is shutdown but the agents remain under the control of miscreants, a new service network of the same extent can be created by simply registering a new domain and reusing the same agents. Moreover, these agents can be used for other malicious purposes (e.g., they can be used as DDoS zombies, to steal personal information from the hosts they are running on, and to act as spam bots). FluXOR is the name of the system we have developed to accomplish these goals. The key idea behind the system is that a fast-flux service network has *multiple distinguishing features* that are not typically found in benign fully qualified domain names. Some of the most

characteristic features are (I) the time-to-live of DNS resources records, (II) the large number of IP addresses into which the canonical hostname is resolved, and (III) the heterogeneous set of organisations that own these addresses. Clearly, these features taken singularly are not enough to distinguish between benign and malicious hostnames. As an example let us compare our sample malicious hostname `www.factvillage.com` with the benign hostname `database.clamav.net`. The latter is a typical example of how DNS resources records with very small time-to-live and round-robin can be used to distribute the load across multiple mirrors (in this case the mirrors are used to distribute updates for the database of signatures of the ClamAV anti-virus [19]). Moreover, as mirrors are hosted by universities and companies, the hosts running a mirror belong to different networks, owned by different organisations, and are distributed around the world. Despite hosts like `database.clamav.net` have most of the characteristics of a fast-flux service network, FluXOR, by monitoring the suspicious hostname for a small period of time and by combining the extracted features using a naïve Bayesian classifier [20], can precisely distinguish between hostnames that are associated with fast-flux service networks from those that are not. It is worth noting that the chosen approach works well also when some of the selected features are not available.

When a fast-flux service network is detected, FluXOR continuously monitors the service network, behaving like a victim and periodically querying various DNS servers to resolve the canonical name of the network for the purpose of enumerating the IP addresses of the compromised hosts that, even for a small period of time, are used as agents.

A fast-flux service network, like the one described in this section, is known in the literature as a *single-flux* network. More complex setups are possible, an example is a *double-flux* network [8]. FluXOR handles indifferently any kind of fast-flux service network, but unfortunately the current implementation does not distinguish between the various types.

For the remaining of the paper, for conciseness, we will refer to the FQDNs associated with a fast-flux service network as malicious and to all the others as benign, although what in the paper is considered benign could be an hostname created for other malicious purposes but not associated with a fast-flux service network. Moreover, we will refer to any hostname whose maliciousness has not been established yet as suspicious.

3 Characterising fast-flux service networks

The features used by FluXOR to distinguish between benign and malicious hostnames are summarised in Table 1 and discussed in detail in the remaining of the section. The features are grouped in three categories: (I) features characterising the domain name to which the suspicious hostname belongs to, (II) features characterising the degree of the availability of the network that is potentially associated with the suspicious hostname, and (III) features characterising the heterogeneity of the potential agents of the network. Some of the features might appear similar initially, but, as shown later, each of them tells us something im-

Category	#	*Description*
Domain name	F_1	Domain age
	F_2	Domain registrar
Availability of the network	F_3	Number of distinct DNS records of type "A"
	F_4	Time-to-live of DNS resource records
Heterogeneity of the agents	F_5	Number of distinct networks
	F_6	Number of distinct autonomous systems
	F_7	Number of distinct resolved qualified domain names
	F_8	Number of distinct assigned network names
	F_9	Number of distinct organisations

Table 1. Summary of the features used to distinguish between benign and malicious hostnames, grouped by category.

portant about the suspicious hostname, especially because some features might not be always available and because there is no well known convention about how some of them are attributed.

3.1 Features characterising the domain name

Domain age (F_1). Benign domains are usually characterised by a relatively long age. Domains used for malicious purposes instead are typically active only for short periods of time. As soon as they are identified, they are deactivated by the authority in charge of the corresponding top-level domain. Thus, miscreants have to register new domains and start to use them right away, to successfully achieve their malicious purposes. The average age of a benign domain is much older than the average age of malicious domain. Indeed, during our experiments, we have estimated that the average age of malicious hostnames is less than five weeks.

Domain registrar (F_2). We empirically observed that most of the domains used to implement fast-flux service networks are registered through a limited number of registrars, typically located in countries with a lax legislation against cybercrime. Our hypothesis is that these registrars perform almost no check when domains are registered. Miscreants can easily complete the registration process using false identities and paying with stolen credit card numbers, making impossible, for the authorities, to identify the person who has effectively registered a domain. On the other hand, the set of registrars used to register benign domains is more heterogeneous and is not likely to overlap with the set of registrars used by miscreants.

3.2 Features characterising the degree of availability of the network

Number of distinct DNS "A" records (F_3). Fast-flux service networks are generally composed by a large number of agents. The authoritative name server for the malicious domain, when queried, returns the set of active agents (i.e., the subset

of agents currently serving the network) by returning multiple DNS "A" records, each one containing the IP address of a specific agent. These resource records are periodically updated by the fast-flux mother-ship to put in the network newly compromised agents and to remove the faulty ones. Thus, after a reasonable long span of time, the number of distinct DNS records of type "A" (i.e., agents IP addresses) that had or have been associated with a malicious FQDN is rather large. The higher the number of distinct DNS records of type "A" associated to the same FQDN, the larger the number of potential agents, and the higher the probability that the FQDN conceals a fast-flux service network.

Time-to-live of DNS resource records (F_4). The fundamental characteristic of fast-flux service networks is the high frequency at which the set of active agents is updated. Most of the agents are end-user machines and consequently it is reasonable to expect that they will appear on-line and disappear very frequently. Thus, to guarantee the high availability of the service offered through the fast-flux network, the set of active agents has to be updated as soon as one of them changes its state. Moreover, the update must be promptly propagated across the Internet, down to the victims. To achieve this goal, the authoritative name server for the malicious domain associates a very short time-to-live to the DNS resource records of the domain. That forces non-authoritative name servers, used by the victims, to flush their cache and to query the authoritative name server very frequently, that in turn returns a different set of active agents every time. The higher the time-to-live associated to the various DNS resource records of a domain, the lower the probability that the domain is malicious. Unfortunately the converse is not always true. Several authoritative name servers for benign domain names associate very short time-to-live to their records for various purposes.

3.3 Features characterising the heterogeneity of the agents

Number of distinct networks (F_5). Fast-flux agents are usually randomly compromised hosts scattered all around the globe. Thus, a malicious FQDN is resolved to many different IP addresses belonging to hosts that very likely belong to different networks. On the other hand, when a benign FQDN encompasses multiple hosts, for load-balancing purposes, these hosts often belong to the same network because they are owned by the same company and physically very close to each other. The higher the number of distinct networks associated to the same FQDN, the more scattered the hosts are, and the more likely these hosts have been compromised and have been used as fast-flux agents. As an example compare the networks associated with the benign FQDN `hp.com` with those associated with the malicious FQDN `www.factvillage.com`, reported respectively in Table 2(a) and Table 2(c). The IP addresses associated with the former all belong to the same network (15.0.0.0/8), while the addresses associated with the latter belongs to completely different networks. As shown in the example of Table 2(b) where each IP address associated with `www.avast.com` belongs to a separate network, this is not always the case.

IP address	F_5	F_6	F_7	F_8	F_9
15.216.110.140	**15.0.0.0/8**	**AS9218**	**polyserve.com**	**HP-INTERNET**	**Hewlett-Packard**
15.192.45.22	**15.0.0.0/8**	**AS9218**	**polyserve.com**	**HP-INTERNET**	**Hewlett-Packard**
15.200.30.24	**15.0.0.0/8**	**AS9218**	**polyserve.com**	**HP-INTERNET**	**Hewlett-Packard**

(a) `hp.com` (benign)

IP address	F_5	F_6	F_7	F_8	F_9
67.228.112.196	67.228.0.0/16	**AS36351**	**avast.com**	SOFTLAYER-4-5	**SoftLayer Tech.**
216.12.205.130	216.12.192.0/19	AS36420	**avast.com**	EVRY-BLK-4	Everyone Internet
74.86.245.119	74.86.0.0/16	**AS36351**	**avast.com**	SOFTLAYER-4-4	**SoftLayer Tech.**

(b) `www.avast.com` (benign)

IP address	F_5	F_6	F_7	F_8	F_9
61.18.66.?	61.18.0.0/16	AS9908	hkcable.com.hk	HKCABLE-HK	HK Cable TV
218.47.195.?	218.47.0.0/16	AS4713	ap.plala.or.jp	PLALA	Plala Net. Inc.
81.173.151.?	81.173.151.0/24	AS8422	netcologne.de	NC-DIAL-IN-POOL	NetCologne

(c) `www.factvillage.com` (malicious)

Table 2. Comparison of the host specific features (F_5 to F_9) characterising two benign and one malicious FQDNs (the entries in bold are those common to multiple IP addresses).

Number of distinct autonomous systems (F_6). An autonomous system (AS) is a connected group of one or more IP prefixes run by one or more network operators with a single and clearly defined routing policy [21]. Thus, distinct networks, but physically very close, might be connected to the Internet through the same AS. As with the previous feature, the majority of benign FQDNs are mapped to hosts located in a circumscribed geographical area and are all part of the same autonomous system. On the other hand, as the agents of a fast-flux network are scattered across all the countries, they typically belong to distinct autonomous systems. As an example let us compare the autonomous systems associated with the benign FQDN `www.avast.com`, with those associated to `www.factvillage.com` (Tables 2(b) and 2(c) respectively). In the first case we have three distinct networks but only two autonomous systems. In the second case, each host, as located in a different country, is part of a different AS.

Number of distinct resolved qualified domain names (F_7). Even if a FQDN is associated with multiple hosts scattered around the globe and part of distinct networks and autonomous systems, the hosts might still be owned by the same company or organisation and thus they can share the same qualified domain name. As an example let us compare the benign FQDNs of Tables 2(a) and 2(b) with the malicious `www.factvillage.com` of Table 2(c). In the first two cases both hostnames are resolved into multiple IP addresses, but these addresses are in turn resolved into canonical hostnames belonging to the same domain (i.e., `polyserve.com` and `avast.com` respectively). The example of `www.avast.com` clearly indicates that all the IP addresses found are legitimate. Unfortunately, that is not completely evident in the case of `hp.com` because the domain name (`polyserve.com`) does not match the domain name of the suspicious FQDN under analysis. Nevertheless, all the IP addresses found are part of the same

domain, which is not common for malicious FQDNs. Indeed, fast-flux agents are compromised hosts belonging to distinct organisations, and the canonical hostnames associated with their IP addresses are solely under the control of the respective owners of the networks and the attacker cannot control in any way these information. In the case of `www.factvillage.com`, each of the three IP addresses found, probably used by dial-up hosts, is resolved into a hostname with a distinct qualified domain, corresponding to that used by the ISP providing the service.

Number of distinct assigned network names (F_8). The network name is the name assigned to a network by the registration authority. Multiple network addresses can be logically grouped under the same network name. This is often the case when the different network addresses are owned by the same company or organisation. Like the other three previous features, the number of distinct network names is an indication of the degree of scattering of the hosts associated with the suspicious FQDN.

Number of distinct organisations (F_9). Each network is assigned to an organisation, but as with network names, same organisation can own multiple networks with one or multiple names. As an example let us consider the benign domain `avast.com` analysed in Table 2(b). Each network is assigned a distinct network name, but two of these networks belong to the same organisation (i.e., SoftLayer Technologies Inc.). Clearly, fast-flux agents randomly distributed around the world share a limited number of organisations.

4 Combining the features for detection

FluXOR initially monitors suspicious hostnames for a short period of time, after which the selected features are analysed to determine whether the domain is malicious or not. The number of domains is incredibly growing. Indeed, it has been estimated that several hundreds of thousands of generic second-level domains (e.g., .com, .org, .net) are registered daily [22]. Consequently, the number of suspicious hostnames to monitor can be very large and it is essential that a precise classification can be accomplished in the shortest period of time, to reduce the workload of the system, but also to promptly intervene to mitigate the damage fast-flux service networks and their bots can cause to the Internet community.

Table 3 shows a comparison of the features of three FQDNs, associated with as many distinct fast-flux service networks, with those of three benign hostnames. Note that the features reported in the table were extracted after only three hours of monitoring. From a quick glance at the numbers in the table it should be clear that each of the selected features effectively tells us something important about the maliciousness of a hostname. Although it is easy to spot by hand benign and malicious hostnames, the numbers in the table show a high variability in the most intuitive features (e.g., F_3 and F_4). For all the analysed hostnames reported in the table it was possible to extract all the selected features. In the general case

	FQDN	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
Benign	`www.avast.com`	539	NetworkSolutions	12	3600	5	3	1	5	2
	`adriaticobishkek.com`	65	Melbourne IT	21	1200	1	1	1	1	1
	`google.com`	542	MarkMonitor	3	300	2	1	1	1	1
	Mean	**493.27**	**N/A**	**2.86**	**4592.53**	**1.27**	**1.11**	**1.08**	**1.21**	**1.07**
	Standard dev.	**289.27**	**N/A**	**3.89**	**7668.74**	**0.65**	**0.36**	**0.74**	**0.58**	**0.25**
Malicious	`www.eveningher.com`	18	PayCenter	127	300	83	49	33	71	54
	`www.factvillage.com`	2	PayCenter	117	300	81	46	34	67	54
	`www.doacasino.com`	2	NameCheap	33	180	19	14	11	19	14
	Mean	**4.85**	**N/A**	**98.13**	**261.49**	**63.75**	**38.36**	**27.98**	**53.58**	**41.47**
	Standard dev.	**4.9**	**N/A**	**37.27**	**59.64**	**23.91**	**12.34**	**8.5**	**18.73**	**15.41**

Table 3. Comparison of three sample benign and malicious FQDNs using the selected features (F_1 is measured in weeks) and comparison of the features of the average benign and malicious FQDNs (computed from a set of about 75 benign and 215 malicious hostnames monitored for about three hours).

some of these features might be missing, but nevertheless the system must be able to correctly discern between malicious and benign hostnames. Furthermore, hosts associated with malicious hostnames tend to be rather scattered, but the degree of the scattering and the number of fast-flux agents might depend on the amount of time the fast-flux service network has been active. If hosts are compromised and turned into agents using a self-propagating malware (e.g., that identifies targets using weak random scanning), it is reasonable to believe that, in the early stage, the agents are rather localised and limited in number. Our goal is to be able to detect if a hostname is malicious as soon as possible, even when the number of agents involved is very small.

For these reasons the detector tries to achieve the best accuracy by combining the selected features using a naïve Bayesian classifier [20]. Given the features of a suspicious hostname, the classifier returns the class (i.e., benign or malicious) to which the hostname is most likely to belong to. The classifier was trained with a set of malicious and benign FQDNs that we manually classified, with the help of data obtained after a week of monitoring. The set of malicious hostnames was composed of hostnames found in spam emails. The set of benign hostnames was composed of hostnames found in spam and non-spam emails. Furthermore, the latter set was extended, to make it more heterogeneous, by adding the address of some randomly selected websites we recently visited. The assumption that the features are completely independent, made by this type of classifier, might appear to simplistic (e.g., features like F_5, F_6, F_8, and F_9 could be correlated). Nevertheless, this approach turned out to have very good performance in many real-world situations and the work of Zhang has shown that the efficacy of naïve Bayesian classifiers has some theoretical foundations [23]. In our context, as discussed later in Section 6, this approach gives very accurate results (for this reason we decided not to evaluate other classifiers). Our hypothesis is that, in practise, no real correlation between the alleged correlated features (F_5, F_6, F_8, and F_9) exists because no convention regulates how ISPs should partition their address space. For example the network associated with a single autonomous system (F_6) could be divided into sub-networks and multiple sub-networks (F_5) can be assigned to the same organisation (F_9).

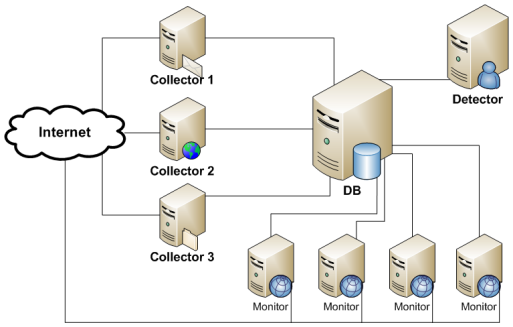


Figure 2. Typical deployment of the system. Multiple collectors and monitors can be used to distribute the workload and to uniformly blend the system in the victims.

5 Architecture and implementation of the system

The architecture of FluXOR is very simple. The system is divided in three components and each one accomplishes a very specific task: (I) one or more *collectors* of suspicious hostnames, (II) one of more *monitors* of suspicious and malicious hostnames, and (III) a *detector* of fast-flux service networks. Figure 2 shows the typical deployment of the system.

FluXOR is entirely developed in Python and consists of about 2150 LOC, without including the code of the web interface used to display the results of the analysis.

5.1 Collector

The *collector* harvests from various sources hostnames that could be associated with fast-flux service networks. Examples of sources are unsolicited emails, instant messages and post in public web forums and blogs. The current implementation of FluXOR only supports harvesting of suspicious hostnames from emails. In the future this component will be extended to support other sources, for example using web crawlers and honeypots. Newly collected hostnames are flagged as suspicious and are considered as such and monitored until the detector classifies them.

5.2 Monitor

The *monitor* is responsible for monitoring suspicious and malicious hostnames. Benign FQDNs, instead, do not need to be monitored (recall that benign hostnames are those already monitored in the past and classified as such). The distinguishing features used by FluXOR to detect fast-flux service networks are extracted from data obtained by querying two different sources: (I) non-authoritative name servers and (II) WHOIS servers. Once a malicious hostname is detected, instead, it is sufficient to perform a subset of the queries used to monitor suspicious hostnames, that is, those used to extract features describing the heterogeneity

of the agents. For statistical and analysis purposes other information about the agents are also collected (e.g., the country in which the hosts are located and their geographical location). A description of the queries performed follows.

Features characterising the domain name (F_1 and F_2). Given a FQDN like `www.factvillage.com`, the age of the domain and the registrar in charge for the domain are determined through WHOIS queries on the name of the second-level domain (e.g. `factvillage.com`). Although the query is conceptually trivial, it presents a serious challenge from the practical point of view. The WHOIS protocol does not define the format in which replies to queries have to be formatted and registries are free to choose the format they like more [24]. Moreover, some registration authorities omit to publish part of the information needed by our analysis. Today the entire IPV4 address space is assigned to 10 different registries. Things are more and more complicated for top-level domains because each domain is assigned to a different registry[1]. Currently we are using a custom WHOIS client that is able to parse the format used by the most common registration authorities. To deal with the registries not currently supported by our client, we rely on a commercial service, that extracts WHOIS information and convert them in XML and offers a free limited number of queries per day. In the future we will extend our client to make the system completely independent from third parties.

Features characterising the degree of availability of the network (F_3 and F_4). The natural approach to enumerate all the resource records of type "A" associated with a particular FQDN (i.e., the IP addresses of the potential fast-flux agents) and the time-to-live of the various records would be to query directly the authoritative name server for the suspicious domain. Although at each query we would always obtain "fresh" records and we would have the highest chance to see previously unseen records (i.e., in the ideal case records are rotated at each query and always have the highest time-to-live), the malicious authoritative name server could easily correlate the high number of queries with a system like FluXOR and consequently fool the analysis by returning fake resource records. The solution currently adopted by FluXOR is to collect the information by issuing recursive queries through multiple public non-authoritative name servers, such that FluXOR queries are blended in the victims' queries. To estimate the maximum time-to-live of the resource records, to maximise the number of agents seen, and to minimise the network traffic, non-authoritative name servers are queried immediately after the cached records have expired.

Features characterising the heterogeneity of the agents (F_5 to F_9). The remaining features are specific to the IP addresses into which a suspicious FQDN is resolved to. The number of distinct networks (F_5) associated with the same

[1] Obviously, a malicious registrar returning (directly or indirectly) fake answers to our WHOIS queries could fool our system. However, in our opinion, that is very improbable: top-level domain registrars are accredited directly by ICANN and they risk to compromise their entire business if they are found to be malicious.

FQDN is computed by enumerating the distinct networks associated with the IP addresses of the potential fast-flux agents. This information can be obtained through a WHOIS query, one for each IP address, directed the respective registry. Similarly, the number of distinct autonomous systems associated with the same FQDN (F_6), is obtained by querying the databases of the regional registries for the AS to which each IP address belongs to. The number of distinct domain names associated with the IP addresses of the potential fast-flux agents (F_7) are obtained by querying name servers for pointer (PTR) resource records associated with each IP address (this kind of query is commonly known as "reverse lookup"). The hostnames obtained are subsequently split to extract the domain name. The network name and the organisation owning the network (F_8 and F_9) are obtained through WHOIS queries. Unfortunately some of the information from which we extract the features of interest are not always available. An example are PTR records associated with the IP addresses of the potential agents.

5.3 Detector

The *detector* of malicious hostnames feeds the set of collected features of the suspicious hostname to the naïve Bayesian classifier for the classification. The classifier is built on top of Weka [25], using the classification algorithm called "NaiveBayesSimple", which models numeric attributes by a normal distribution.

6 Experimental results

We have been running FluXOR since the beginning of January, but unfortunately the system has been working without interruption only since mid January. Currently the monitor and the detector are located on the same machine, an AMD Athlon XP 1.8GHz with 384Mb of RAM, running GNU/Linux and using MySQL for the persistent storage. The detector has been trained with three different data-sets, containing features extracted after one, two, and three hours of monitoring respectively. The three training sets were composed by 50 benign and 75 malicious FQDNs manually analysed and classified. The collector was located on the mail server of our laboratory and processed all the spam emails forwarded by the mail server of our department. Malicious FQDNs were all extracted from spam emails, while benign hostnames were extracted from emails (both spam and non-spam) and from the history of our browsers.

Table 4 summarises the most important numbers of our experiments: the volume of spam email messages processed, the number of URLs extracted, the number of FQDNs active at the time the emails were received, the number of fast-flux service networks detected, the number of distinct fast-flux agents, and the number of hypothetical botnets the detected fast-flux agents were part of. About 7.8% of the active FQDNs turned out to conceal fast-flux service networks served by 31998 distinct fast-flux agents, which we believe to belong to 16 distinct botnets (we considered two fast-flux service networks associated with the same botnet if they were pointing to the same website).

Description	#
Processed spam email messages	44804
Extracted URLs	15281
Active FQDNs (whose hostname could be resolved)	4961
Fast-flux service networks	**387**
Fast-flux agents	**31998**
Botnets	**16**

Table 4. Summary of the results obtained using FluXOR to monitor the suspicious hostnames found in spam emails. Note that the number of agents is the number of distinct IP addresses. Dial-up hosts using dynamically assigned addresses might use multiple addresses and multiple hosts might share some addresses.

We evaluated the detection accuracy automatically, before training the classifier, and manually by comparing the output of the detector with our belief. Although during the manual analysis we found some corner case benign and malicious hostnames, the detector always classified the suspicious hostnames correctly. That is, we had *zero* false-positives. We also tried to correlate the data collected in the last month to understand the botnet phenomenon by observing botnets activity from the prospective of a victim, starting from hostnames associated with fast-flux service networks found in spam emails.

6.1 Detection accuracy

We evaluated the accuracy of our detection strategy following two different strategies: (I) an automatic cross-validation with the three training data-sets and (II) a manual analysis of a random subset of the active FQDNs extracted from the emails.

Part of our training data-set was used to estimate the accuracy of the model using cross-validation, with 5 and 10 folds [26]. No hostname was misclassified. The manual analysis was performed by comparing the response of the detector with our belief about the maliciousness of the hostnames. Hostnames whose maliciousness was difficult to attest were monitored for a day. The detector was invoked three times on each sample, the first time with the features extracted after one hour of monitoring and the corresponding model, the second and the third time with the features extracted after two and after three hours of monitoring, and the corresponding model, respectively. Note that the amount of active hostnames processed were rather large and impossible to analyse manually in its entirety. Thus, we pruned the set using a filter to identify all the hostnames that were undoubtedly benign (i.e., those, after three hours of monitoring, associated with only two or less IP addresses and classified as benign). The manual analysis confirmed the correctness of our classifier, no hostname was misclassified.

During the manual analysis of the accuracy of the detector we came across some peculiar benign hostnames that had some of the characteristic of malicious hostnames. Two examples of these hostnames are `imageshack.us` and `database.clamav.net`. These hostnames are associated with very small time-

to-live and are resolved in multiple IP addresses, 129 and 21, respectively[2]. All the 129 distinct IP addresses associated with the first hostname belong to the same network. That makes us believe that the hosts are hosted in a server farm somewhere and that load-balancing is implemented using DNS round-robin. On the other hand, the IP addresses associated with `database.clamav.net` (see the discussion in Section 2) are located in 12 distinct networks, because mirrors are voluntarily hosted by companies and universities. Both hostnames belong to domains registered several years ago through registrars that are not commonly used by miscreants. In both cases FluXOR correctly classified the hostnames, even when the detection was performed using the features collected during one hour of monitoring only. Other examples of correctly classified benign hostnames that share some of the features of hostnames used for fast-flux service networks are `pool.ntp.org` and `en.wikipedia.org.nyud.net` (Wikipedia mirrored through Coral Content Distribution Network).

We also identified several very young (or not very active) fast-flux service networks for which, after an hour of monitoring, we only saw from three to five distinct agents. After three hours of monitoring the size of the network was still very small and reached only seven or eight agents. Despite the small number of agents, the hostnames were always classified as malicious, even when detection was performed using the data collected in an hour of monitoring. Not completely convinced of the response of the detector, we continued to monitor the hostnames. After several days the service networks encompassed hundred of hosts.

Three observations are worth mentioning. First, the detector is surprisingly precise. Second, in less than three hours we can precisely tell if a FQDN is malicious or not. Third, the current status of the fast-flux service network might not reflect the status of the network in the future (e.g., a hostname can be used for any kind of purpose at the beginning and then associated with a fast-flux service network in the future). The detector can only classify the current status of the hostname and, in order to detect a change of the status, the hostname must be monitored and classified again.

6.2 Empirical analysis of the fast-flux service networks phenomenon

Although we collected suspicious hostnames from a single source only and the number of hostnames collected was rather small, the number of detected fast-flux service networks and the number of their agents is unexpectedly very large. About 7.8% of the hostnames analysed were malicious. In the following paragraphs we briefly summarise some results we believe are interesting. Real-time and complete results of the analysis can be found on-line at `http://fluxor.laser.dico.unimi.it`.

Figure 3 shows the number of fast-flux agents, belonging to six distinct networks, detected during the time. The number of agents detected depends on

[2] The hostname `database.clamav.net` is resolved into different IP addresses according to the country from which the request comes from. During our experiments we used a public DNS located in the U.S., which is the country with the highest number of IP addresses associated with the hostname.

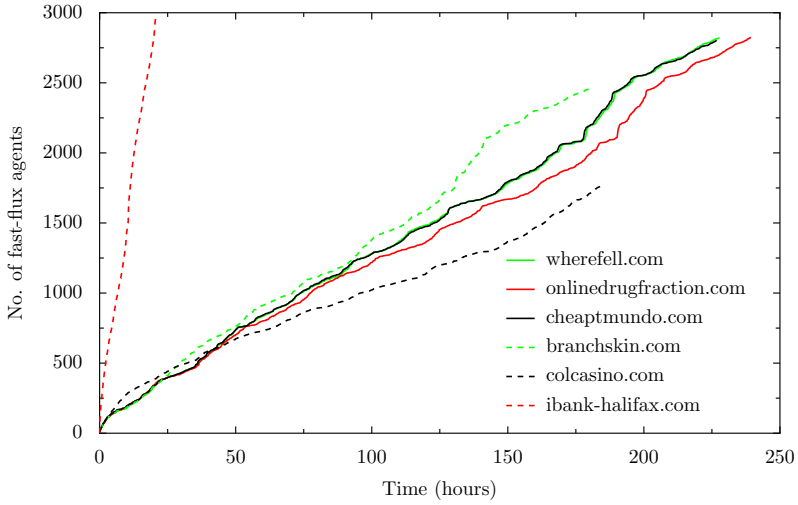


Figure 3. Number of fast-flux agents, serving some representative fast-flux service networks, detected during the time.

many factors. For example the time-to-live of the DNS resource records, the number of records returned at each query, and the frequency at which the set of active agents is updated. The case of `ibank-halifax.com` is very impressive. In less than a day we detected about 3000 agents. The turnaround of agents in the average fast-flux service network is much smaller. The average number of new agents detected daily was about 122.

We visited some of the websites served thorough the detected fast-flux service networks and found out that several FQDNs were associated with the same website. The networks were probably pointing to the same mother-ship. Our hypothesis is that, to improve the availability of the system, miscreants registered multiple domains. If a domain was shutdown, victims could still be served through the other domains. Thus, it is more difficult for the authorities to eradicate the scam. Besides the common website, this hypothesis is further corroborated by the fact that multiple fast-flux service networks are served by the same set of agents. Figure 3 shows that the number of agents detected during the time for the FQDNs `wherefell.com` and `cheaptmundo.com` is growing symmetrically. We also observed that the two domains share the same authoritative name servers and also about 81% of the agents. We believe it is reasonable to assume that all the fast-flux networks pointing to the same website, and thus used for the same fraud, are served by agents belonging to the same botnet. Table 5 shows some of the hypothetical botnets associated with the detected fast flux service networks, and their extent in number of agents (the name assigned to the botnet is derived from the title of the main page of the website).

Figure 4 shows the geographical distribution of the detected agents. Their heterogeneous geographical distribution testifies that the scale of the problem is world-wide.

Botnet (Website)	*# networks*	*# agents*
Halifax scam	1	13958
Canadian Pharmacy	312	4773
EuroPrimeCasino	7	3242
Cheap EOM Software	1	2371
PosteItaliane scam	1	50

Table 5. Some of the fast-flux service networks detected, grouped by botnet.

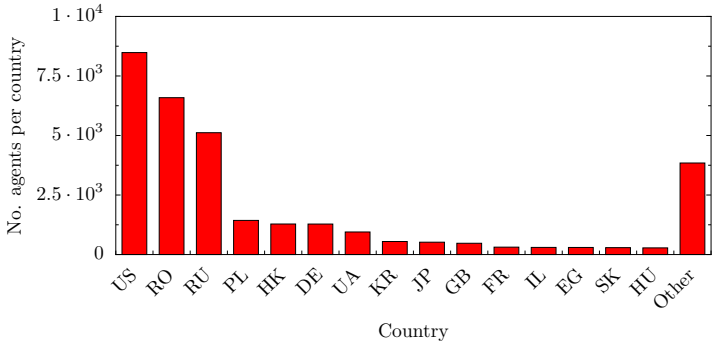


Figure 4. Geographical distributions of the detected fast-flux agents.

7 Related work

The botnet problem has been studied by the research community mainly from two different prospectives: from the prospective of the bot, to study its code and its behaviours, and from the prospective of the network, to study the traffic generated by these bots. The approach proposed in this paper studies the phenomenon from the prospective of a victim of the scams perpetrated by these botnets.

The first analysis and characterisation of fast-flux service networks was presented by the HoneyNet project [8]. The report analysed the two types of networks seen so far (i.e., single-flux and double-flux service networks) and analysed the behaviour of a malware with the capabilities of a fast-flux agent. The problem of detecting and mitigating fast-flux service networks was concurrently addressed by Holz *et al.* [27]. Our work and theirs are very similar. They also propose a detection method based on the observation of some features common in fast-flux service networks. However, while we employ 9 different features, Holz *et al.* focus on just three features (i.e., the number of DNS "A" records, the number of DNS "NS" records, and the number of distinct autonomous systems fast-flux agents belongs to), one of which (the second) does not seem to be used at all for the classification. We believe such a limited set of distinguishing features could lead to several false-positives, mostly because such features are also typical of domains that employ some DNS load-balancing techniques (e.g., such as `pool.ntp.org`). Our extensive evaluation has shown that FluXOR is undoubtedly robust and very

efficient. The work of Rajab *et al.* differs from ours in term of techniques and goals. However, the point of view from which botnets are observed is similar to ours [28]. We detect and monitor fast-flux service networks by performing simple DNS and WHOIS queries. Similarly, in their work, Rajab *et al.* tracked botnets by infiltrating in IRC channels and by measuring the cache-hit rate of the DNS servers queried by the bots to contact their control centre.

The analysis of the network traffic generated by compromised machines transformed into bots and the traffic generated by bot "management" open several opportunities for understanding the phenomenon and for detection. Rishi, by monitoring the network traffic for unusual IRC communications like connection to uncommon servers and ports and use of suspicious nicknames, detects machines infected with bots [11]. Karasaridis *et al.* developed a transport and application layer traffic analyser to detect IRC based bots on wide-scale [13]. Cooke *et al.* studied the effectiveness of detecting botnets by directly monitoring IRC communications and other command and control activities. Unfortunately their work demonstrated that a more comprehensive approach, based on the correlation of data coming from multiple sources, is required to precisely detect botnets. BotHunter correlates alerts coming from different types of sensors to identify the communication sequences that occur during the infection process (i.e., target scanning, infection exploit, binary egg download, and outbound scanning) [12]. Dagon *et al.* used DNS redirection to detect machines part of specific botnets and to understand how time and geographical location affect the spread dynamics of these botnets [14]. The problem of understanding how challenging is to estimate the size of botnets were addressed by Rajab *et al.* [3]. A similar problem was subsequently addressed by Dagon *et al.* [10]. They proposed several metrics to measure the utility of botnets for various activities and presented a taxonomy of botnets based on these metrics and on the topological structure of the networks.

Many researchers have studied botnets by studying how bots behave and how they are implemented. These bots can be analysed using dynamic, static, or a hybrid dynamic and static analysis. BotSwat characterises and detect the typical behaviours of bots using dynamic taint analysis [29]. Barford *et al.* statically analysed the codebase of four of the most common IRC bots to understand their propagation methods, the mechanism used for their remote control, the delivery and the obfuscation mechanisms used [30]. Other specific bots have been thoroughly analysed to understand the new techniques used and the best method to block them [31,32,7].

8 Conclusion

Botnets represent one of the major threats for the Internet community. In this paper we have presented FluXOR, the system we have developed to detect and monitor fast-flux service networks. Fast-flux service networks are used by the miscreants, controlling the biggest and most powerful botnets, to hide and to maximise the availability of the core components of their business. Fast-flux service networks offer researchers the possibility to observe a botnet from the out-

side, by simply observing what a victim of these botnets could observe. Through FluXOR we have demonstrated that, by tracking fast-flux service networks with very simple queries any end-user can perform, we were able to detect, in a very short period of time, more than thirty thousands compromised machines remotely controlled by miscreants and used for various on-line frauds.

9 Acknowledgements

We would like to thank our shepherd, John McHugh, and the anonymous reviewers for their useful comments and suggestions.

References

1. Franklin, J., Perrig, A., Paxson, V., Savage, S.: An inquiry into the nature and causes of the wealth of internet miscreants. In: Proceedings of the 14th ACM conference on Computer and communications security (CCS'07), New York, NY, USA, ACM (2007) 375–388
2. Ször, P.: The Art of Computer Virus Research and Defense. Addison Wesley Professional (2005)
3. Rajab, M.A., Zarfoss, J., Monrose, F., Terzis, A.: My Botnet is Bigger than Yours (Maybe, Better than Yours): Why Size Estimates Remain Challenging. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07, Berkeley, CA, USA, USENIX Association (2007)
4. Furst, M.: Expert: Botnets No. 1 Emerging Internet Threat. CNN Technology (2006)
5. Markoff, J.: Attack of the Zombie Computers Is a Growing Threat, Experts Say. The New York Times (January 2007)
6. Corporation, F.S.: Malware Information Pages: Warezov (2006) `http://www.f-secure.com/v-descs/warezov.shtml`.
7. Porras, P., Saidi, H., Yegneswaran, V.: A Multi-perspective Analysis of the Storm (Peacomm) Worm. Technical report, SRI International (October 2007)
8. The Honeynet Project & Research Alliance: Know Your Enemy: Fast-Flux Service Networks (2007)
9. Gaudin, S.: Storm Worm Erupts Into Worst Virus Attack In 2 Years (2007)
10. Dagon, D., Gu, G., Lee, C., Lee, W.: A taxonomy of botnet structures. In: Proceedings of the 23 Annual Computer Security Applications Conference (ACSAC'07). (December 2007)
11. Goebel, J., Holz, T.: Rishi: identify bot contaminated hosts by irc nickname evaluation. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07, Berkeley, CA, USA, USENIX Association (2007)
12. Gu, G., Porras, P., Yegneswaran, V., Fong, M., Lee, W.: BotHunter: Detecting malware infection through ids-driven dialog correlation. In: Proceedings of the 16th USENIX Security Symposium (Security'07). (August 2007)
13. Karasaridis, A., Rexroad, B., Hoeflin, D.: Wide-scale botnet detection and characterization. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07, Berkeley, CA, USA, USENIX Association (2007)

14. Dagon, D., Zou, C., Lee, W.: Modeling botnet propagation using time zones. In: Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS'06). (2006)
15. Ramachandran, A., Feamster, N., Dagon, D.: Revealing botnet membership using dnsbl counter-intelligence. In: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06), Berkeley, CA, USA, USENIX Association (2006)
16. Cooke, E., Jahanian, F., Mcpherson, D.: The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In: Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI). (June 2005) 39–44
17. Mockapetris, P.: Domain names – concepts and facilites. RFC 1034, Internet Engineering Task Force (November 1987)
18. Mockapetris, P.: Domain names – implementation and specification. RFC 1035, Internet Engineering Task Force (November 1987)
19. Kojm, T.: Clam AntiVirus (`http://www.clamav.net`).
20. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann (1995) 338–345
21. Hawkinson, J., Bates, T.: Guidelines for creation, selection, and registration of an autonomous system (as). RFC 1930, Internet Engineering Task Force (March 1996)
22. DomainTools.com: Domain Counts & Internet Statistics (`http://www.domaintools.com/internet-statistics/`).
23. Zhang, H.: The Optimality of Naïve Bayes. In: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA, AAAI Press (2004)
24. Daigle, L.: WHOIS protocol specification. RFC 3912, Internet Engineering Task Force (March 2004)
25. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edition edn. Morgan Kaufmann, San Francisco (2005)
26. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1995) 1137–1145
27. Holz, T., Gorecki, C., Freiling, F., Rieck, K.: Detection and Mitigation of Fast-Flux Service Networks. In: Proceeding of the 15th Annual Network & Distributed System Security Symposium (NDSS'08). (February 2008)
28. Rajab, M.A., Zarfoss, J., Monrose, F., Terzis, A.: A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06), New York, NY, USA, ACM (2006) 41–52
29. Stinson, E., Mitchell, J.C.: Characterizing Bots' Remote Control Behavior. In: Proceedings of the Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer (2007) 89–108
30. Paul, B., Vinod, Y.: An Inside Look at Botnets. In: Malware Detection. Volume 27 of Advances in Information Security. Springer (2007)
31. Chiang, K., Lloyd, L.: A case study of the rustock rootkit and spam bot. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07, Berkeley, CA, USA, USENIX Association (2007)
32. Daswani, N., Stoppelman, M.: The anatomy of clickbot.a. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07, Berkeley, CA, USA, USENIX Association (2007)