A Collaborative Approach to Botnet Protection

Matija Stevanovic, Kasper Revsbech, Jens Pedersen, Robin Sharp, Christian Jensen

A Collaborative Approach to Botnet Protection

Matija Stevanovic[1], Kasper Revsbech[1], Jens Myrup Pedersen[1], Robin Sharp[2] and Christian Damsgaard Jensen[2]

[1] Department of Electronic Systems
Aalborg University
{mst,kar,jens}@es.aau.dk
[2] Department of Informatics and Mathematical Modelling
Technical University of Denmark
{robin,Christian.Jensen}@imm.dtu.dk

Abstract. Botnets are collections of compromised computers which have come under the control of a malicious person or organisation via malicious software stored on the computers, and which can then be used to interfere with, misuse, or deny access to a wide range of Internet-based services. With the current trend towards increasing use of the Internet to support activities related to banking, commerce, healthcare and public administration, it is vital to be able to detect and neutralise botnets, so that these activities can continue unhindered. In this paper we present an overview of existing botnet detection techniques and argue why a new, composite detection approach is needed to provide efficient and effective neutralisation of botnets. This approach should combine existing detection efforts into a collaborative botnet protection framework that receives input from a range of different sources, such as packet sniffers, on-access anti-virus software and behavioural analysis of network traffic, computer sub-systems and application programs. Finally, we introduce ContraBot, a collaborative botnet detection framework which combines approaches that analyse network traffic to identify patterns of botnet activity with approaches that analyse software to detect items which are capable of behaving maliciously.

Keywords: Botnets, Botnet Detection, Collaborative Framework, Correlation Analysis

1 Introduction

During the last few decades, the Internet and applications based on it have experienced a tremendous expansion to the point at which they have become an integral part of our lives, supporting a wide range of services such as banking, commerce, healthcare, public administration and education. The growing reliance on the Internet introduces a number of security challenges that require sophisticated and innovative solutions. The main carrier of malicious activities on the Internet is malicious software, i.e., malware, which includes vira, trojans, worms, rootkits and spyware.

Botnets represent a state of the art deployment of malware that combines many existing advanced malware techniques. A bot is a computer which has been infected with some form of malware which can provide a remote attacker with total, unconditional

and imperceptible control over the compromised computer. A botnet is a (usually) large collection of such compromised computers that are infected with the specific malware instance which enables them to be controlled by the malicious third party ("botmaster"). Botnets may range in size from a couple of hundred to millions of bots, spanning over home, corporate and educational networks covering different parts of the world. Botnets provide a collaborative and highly distributed platform for a wide range of malicious and illegal activities such as sending spam, launching distributed denial of service (DDOS) attacks, malware distribution, click fraud, distribution of illegal content, collection of confidential information and attacks on industrial control systems and other critical infrastructure.

Some recent cybesecurity studies [10, 28], claim that more than 40 percent of computers world-wide are infected with some kind of bot malware, thus being actively or passively involved in the malicious activities of a botnet. Additionally, these studies have shown that the average size of botnets is growing and that the biggest botnets can easily involve several million bots. The size of such botnets indicates the great potential in terms of processing power and available bandwidth. Pairing this with collaborative and coordinated action makes botnets rightfully regarded as one of the biggest threats to cybersecurity up to date.

Neutralisation of botnets involves technical, legal and political issues, and therefore requires an inter-disciplinary collaboration to achieve a successful result. At the same time, there is the challenge of raising computer users' awareness of the dangers posed by botnets as well as the challenge of persuading them to take the necessary steps to hinder the spread of botnets. Although complex in its nature, the problem of neutralisation and mitigation of botnets primarily relies on the ability to detect them. Extensive research efforts have therefore been made during the last decade to find ways to efficiently detect botnets. Many experimental systems have been reported in the literature, based on numerous technical principles and varying assumptions about bot behaviour and bot traffic patterns. However due to the dynamic nature of botnets and constant improvement of the malicious techniques, the success of the proposed detection and mitigation approaches has been limited. In this paper we present an overview of existing botnet detection techniques and we analyse their ability to cope with the challenges posed by modern botnets. We elaborate on the need for a more comprehensive detection approach in order to provide efficient and effective neutralisation of botnets. Finally, we introduce ContraBot, a collaborative botnet detection framework that combines approaches that analyse network traffic to identify patterns of botnet activity with approaches that analyse software to detect items which are capable of behaving maliciously.

The rest of this paper is organized in the following way: Section 2 examines the threat of botnets and identifies some trends in the development of botnets. A survey of earlier work on botnet detection is presented in Section 3. The need for a systematic approach to botnet protection is discussed in Section 4, which argues why combining existing approaches to botnet detection will provide better results in the fight against botnets. Section 5 presents the architecture of the ContraBot platform, which defines a collaborative framework for botnet detection and neutralization. Finally, a discussion of the ContraBot architecture and directions for future work are outlined in Section 6.

2 Threats from Botnets

As a state of the art form of malware, bots are taking advantage of multiple malicious techniques and evolve at an unprecedented speed, presenting a considerable challenge to existing botnet defence systems. Current botnets are characterized by diversity of protocols and structures, usage of advanced code obfuscation techniques and a tendency to spread to new platforms.

The essential component, and at the same time the main carrier of botnet functionality, is the C&C (Command and Control) channel that is established between the botmaster and the infected computers. Moreover, the C&C channel represent the main characteristic that distinguish bots from the other malware forms. Botmasters rely on the C&C channel to issue commands to their bots and receive information from the compromised machines. C&C infrastructure has been evolving in recent years, so that today several control mechanisms in terms of protocols and structures are used for the realization of the C&C channel.

The earliest botnets utilized a centralized C&C network architecture, where all bots in a botnet contact one (or a few) C&C server(s) owned by the botmaster. The centralized C&C channels are usually based on the IRC or HTTP protocols. IRC-based botnets are realized by deploying IRC servers or by using an IRC server in a public IRC network. The botmaster specifies a channel, which bots connect to and listen on to receive commands from the botmaster. HTTP-based botnets are similar to the IRC-based ones. After infection, bots contact a web-based C&C server and notify the server with their system-identifying information via HTTP, while the server sends back commands via HTTP responses. IRC- and HTTP-based C&C have been widely used in botnets, but both of them are vulnerable to a single point of failure. That is, once the central IRC or HTTP servers are identified and disabled, the entire botnet will be disabled. Some examples of IRC and HTTP botnets that have been observed ”in the wild” (Agobot, SDbot, Zeus, etc.) have had more than a million bots and have been successfully used for malicious actions such as DDoS attacks, identity theft, etc.

In order to be more resilient to counter-measures, the attackers have recently started to build botnets using decentralized C&C infrastructures such as P2P [6] or advanced hybrid P2P structures [33], where bots belonging to a P2P botnet form an overlay network in which any of the nodes (i.e. bots) can be used by the botmaster to distribute commands to the other peers or collect information from them. In these botnets, a botmaster can join, publish commands and leave at any time at any place. While more complex, and perhaps more costly to manage compared to centralized botnets, P2P botnets offer higher resiliency, since even if a significant portion of a P2P botnet is taken down (by law enforcement or network operators) the remaining bots may still be able to communicate with each other and with the botmaster to pursue their malicious purpose.

One of the first well-known examples of a P2P botnet was the Storm botnet [18] from 2008. Storm was estimated to run on over a million compromised computers and was primarily used for sending spam emails. It utilized Kademlia [6], a decentralized Distributed-Hash-Table (DHT) protocol. Other noteworthy recent P2P botnets include Waledac [25] and Conficker [19]. Although similar to the Storm botnet they employ self-defined communication protocols based on HTTP and fast-flux techniques. This illustrates the fact that modern botnets often use additional techniques for improving

resilience and robustness of communication, such as obfuscation of existing communication protocols or development of new ones, encryption of communication etc. We can conclude that botnet detection approaches designed specifically to detect and mitigate centralized botnets will be less effective for such novel, highly decentralized botnets. Also, given the range of different C&C infrastructures that the botmaster can employ, a detection method targeting only specific C&C infrastructure cannot provide sufficiently effective detection of modern botnets.

Besides the diversity in structure and protocols, current botnets commonly employ code obfuscation techniques such as polymorphism and metamorphism. These enable the bot code to mutate without changing the functions or the semantics of its payload. Usually, in the same botnet, bot binaries are different from each other. Since signature-based detection schemes look for specific data patterns within binaries, they require constant update of signatures in order to be able successfully detect bots. However even then these techniques are limited to detecting known bots.

Whereas almost all modern botnets have targeted personal computers (PCs), attackers are constantly searching for new ways of disseminating their product, such as finding new platforms to host botnets. One of the trends noticed at the beginning of 2012 was the occurrence of botnets on mobile telephones. The first mobile-based bot was Android.Counterclank, developed for the Android platform. This could be downloaded via the Android Market through several application packages. According to Symantec reports [29], this bot could carry out commands from a remote server and was capable both of stealing information from, and displaying ads on, infected Android handsets. Although promptly detected and taken down, this bot showed that botnets are slowly spreading to the smartphone domain. The popularity of smartphones and the fact they have a lot of processing power and bandwidth at their disposal will certainly continue to attract the attention of botmasters. The unique features of mobile devices such as communication via multiple technologies, like Bluetooth and NFC (Near Field Communication) in addition to the conventional IP network, could also provide mobile botnets with more stealthy and robust functioning. As existing detection techniques only cover PC-based botnets, further work is needed to counter the innovations seen in these novel botnets on smartphones and other potential new platforms.

3 Earlier Work on Botnet Detection

Botnet detection systems go back to the middle 2000s, and many experimental systems have been reported in the literature, with various aims in mind, and based on diverse technical principles and varying assumptions about bot behaviour and traffic patterns. Aims may include features such as automated operation, independence of communication topology and protocol, independence from payload content and real time detection.

Depending of the point of deployment, the detection approaches can be classified as client-based or network-based. In client-based approaches, the detection system is deployed within the client computer, and examines the computer's internal behaviour and/or traffic visible on the computers external network interfaces. Network-based detection, on the other hand, is deployed at the edge of the network (usually in routers or firewalls), providing botnet detection by passive monitoring of network traffic.

Like intrusion detection systems in general, botnet detection systems may be based on recognising characteristic patterns of code or data ("signatures") or patterns of behaviour. Likewise, they may be based on misuse detection – i.e. recognising signatures or behaviour patterns known to be associated with undesirable activities – or anomaly detection, where the idea is to detect noticeable deviations from normal behaviour. Misuse detection often gives fewer false positives, but plainly cannot be used to detect new bots or obfuscated variants of already known bots; anomaly detection is able to detect new forms of malicious activity, but may give many false positives if the pattern of normal activity changes. Thus there is a very wide range of potential combinations of approaches.

3.1 Client-based detection

Several client-based detection systems have been proposed. One of the earliest was BotSwat [26], which was based on a taint tracking system developed to discover programs that take advantage of received network data from an unreliable external source and to identify the potential remote control behaviour of bots. The main idea behind BotSwat is that a bot installed on the host computer has a specific pattern of behaviour that can be recognized by monitoring execution of an arbitrary executable binary, and tracing the traffic to its external source. This approach is directed at detection of individual bots by misuse detection and is independent of botnet topology and communication protocol.

The approach used by Masud et al. [15] illustrates a method for botnet traffic detection based on the assumption that bots have a different response pattern from humans and that is possible to detect them by correlating multiple network flow log files on the hosts. The approach utilizes data mining techniques to extract relevant features from these log files and detect C&C traffic. The method offers several advantages such as real-time operation, and independence from communication protocol and topology. However, the approach has two major limitations. Firstly, it requires access to payload content, so it cannot detect botnets which use encrypted communication. Secondly, as the approach relies on the assumption that the response pattern of bots differs from that of humans, it is vulnerable to evasion techniques that include mimicking of human response patterns.

EFFORT [24] is one of the most recent client-based detection approaches, and is based on intrinsic characteristics of bots from both client and network aspects. The detection framework uses a multi-module approach that correlates bot-related information gathered by inspection of client computer internals, by monitoring the computer's interaction with the human user (key strokes and mouse input) and by monitoring traffic on the computer's external interfaces. The method has a number of advantages such as independence of topology and communication protocol, and the ability to detect encrypted and obfuscated protocols. The major limitation is that each detection module within the framework that detects specific bot-related occurrences can be evaded by suitably chosen evasion techniques. Nor can this technique provide real-time detection.

3.2 Network-based detection

Network-based detection is a more common principle used for detecting botnets and is primarily realised by passive network monitoring. Some of the earlier approaches of this type, such as Rishi [5], Snort [21] and BotHunter [9] were based on misuse detection, using signatures of botnet malicious activity and C&C communication in order to detect them.

Rishi [5] was one of the first detection techniques to tackle the problem of IRC botnets. It uses a signature-based detection algorithm that matches the IRC nickname with typical nickname patterns of IRC bots. Rishi is based on passive traffic monitoring for suspicious IRC nicknames (Layer 7), IRC servers, and uncommon server ports (Layer 4). It uses a specially developed scoring system and n-gram analysis to detect bots that use uncommon communication channels that are commonly not detectable by classical intrusion detection systems. However this approach has not had much impact due to the fact that it does not have the ability to detect IRC botnets that use encrypted or obfuscated communication.

Snort [21] is an open source network intrusion detection system (NIDS) based on misuse detection. Snort monitors network traffic, and is configured with a set of signatures and rules for logging traffic which is deemed suspicious. This method has several advantages, such as immediate detection and the impossibility of false positives, but of course can only detect known botnets. Moreover, as it performs deep packet inspection it can easily be defeated by encryption or obfuscation of payload content.

BotHunter [9] was the first open source botnet detection system available for broad public use. It was developed as an extension of Snort, by the addition of two anomaly detection plug-ins on top of Snorts existing signature database. BotHunter defines a model of the botnet infection dialogue process, which is intended to match the life-cycle of contemporary botnets, and uses it as a guideline to recognise infection processes within the network. However, this approach suffers from many shortcomings, primarily inherited from Snort, such as the inability to detect encrypted traffic, vulnerability to various evasion techniques and attacks directed at the content of the correlation matrix. Additionally BotHunter can only identify bots whose life-cycle follows the chosen model of infection.

Other network-based detection approaches are based on detection of statistical anomalies in network traffic, such as high network latency, high volume of traffic, traffic on unusual ports and unusual system behaviour, which are exhibited as a consequence of botnet communication. Important examples are BotSniffer [7] and the approach of Karasaridis et al. [12].

BotSniffer [7] is a network-based anomaly detection approach developed to identify botnet C&C channels in a local area network without any prior knowledge of botnet signatures. It is based on the observation that, because of the pre-programmed activities related to C&C communication, bots within the same botnet will likely demonstrate spatial-temporal correlation and similar behaviour. The technique captures behavioural patterns of botnet traffic and utilizes statistical algorithms on them to detect botnets. BotSniffer was primarily developed to detect centralized IRC and HTTP based botnet C&C channels, and cannot cope with modern P2P botnets. Furthermore, the method is vulnerable to evasion techniques such as misusing whitelists, encryption, using very

long or random response delays, injecting random noise packets, and evasion of the protocol matcher and etc.

Karasaridis et al. [12] used an anomaly-based botnet detection method aimed at detecting botnet controllers by monitoring transport layer data. The method was developed to detect IRC botnet controllers, i.e. IRC servers within large Tier-1 ISP networks. The approach is entirely passive and does not depend on botnet behaviour signatures or particular application layer information, so it is able to detect bots using encrypted and obfuscated protocols. However, the approach relies on an IDS to provide an indication of suspicious hosts, so it cannot detect unknown botnets or bots, and it cannot handle modern HTTP and P2P botnets.

In parallel with these efforts based on network traffic in general, a subgroup of botnet detection approaches directed at detecting anomalies of DNS traffic emerged. DNS-based detection approaches rely on detection of patterns within DNS traffic that can indicate the presence of a bot or botmaster within the network. Some of the most prominent DNS-based approaches realize botnet detection by detecting anomalies of DNS traffic as in [30], performing DNSBL (DNS Black List) counter intelligence [20], capturing DNS group behaviour [2] or building a reputation system for DNS queries [1]. Many novel classes of botnets (P2P, hybrid P2P), however, do not require a DNS service for their functioning so these approaches have a significantly limited detection scope.

Some more recent approaches to botnet detection have attempted to detect patterns of botnet traffic by employing sophisticated machine learning techniques. Machine learning is used because it offers the possibility of automated, real-time recognition of patterns within traffic without a need for traffic exhibiting specific anomalies. Several detection approaches that employ machine learning have been proposed over the years such as in Strayer et al. [27], Botminer [8], Lu et al. [13], Saad et al. [22], and Zhang et al. [36], providing more or less efficient botnet detection.

Strayer et al. [27] developed one of the first approaches that employed machine learning to detect patterns of botnet traffic within the network. Several machine learning approaches were utilized and their performance in classifying IRC traffic flows evaluated. The approach provides a real-time detection framework which has the ability to detect botnets even before a cyber-attack occurs. However, it only has the ability to detect IRC botnets with centralized topology and it requires external judgment, either by humans or machines, in order to generate reliable alarms for the existence of a botnet. This limits its practical usability.

BotMiner [8] uses an approach based on data mining, and was developed in order to successfully identify modern botnets, which can significantly differ in size, structure, communication technology and purpose. The technique assumes that bots within the same botnet will be characterized by similar malicious activity and similar C&C communication patterns. BotMiner employs clustering techniques in order to detect similarities within different hosts in the network. This technique is entirely independent of the C&C protocol, structure, infection model of botnets and it does not require prior knowledge of botnet specific signatures. However, by design it essentially targets groups of compromised machines within a monitored network, so it may not be effective at detecting individual compromised hosts. Moreover, the technique is exposed to various

evasion techniques, and performs poorly in situations where stealthy P2P botnets, that mask their traffic within non-malicious P2P traffic, are present in the network.

A recent study in the field of botnet detection by Saad et al. [22] considers the problem of detecting P2P botnets by using machine learning techniques. The study evaluates the ability of commonly used machine-learning techniques to meet on-line botnet detection requirements such as adaptability, novelty detection and early detection. The study shows that machine learning algorithms have a great potential for detecting patterns of botnet traffic. However it also indicates that the performance of these techniques is highly dependent on the features selected for classification or cluster analysis and that they often have high computational requirements.

Zhang et al. [36] describe a novel botnet detection system that can identify stealthy P2P botnets, even when malicious activities may not be observable. Their approach focuses on identifying P2P bots within a monitored network by detecting their characteristic C&C communication patterns, regardless of how they perform malicious activities. To accomplish this, the system derives statistical fingerprints of the P2P communications generated by P2P hosts, and uses them to distinguish P2P bots from hosts that are part of legitimate P2P networks. This system can detect stealthy P2P botnets even when the underlying compromised hosts are running legitimate P2P applications (e.g. Skype). However, it targets only P2P bots, so it cannot cope with botnets based on IRC or HTTP. Moreover, as the method relies on numerous assumptions regarding P2P communication and P2P bot traffic patterns, it is vulnerable to evasion techniques such as using a legitimate P2P network, randomizing traffic patterns, using a malicious DNS server, or injecting P2P control messages.

Although each of the methods described above has a certain range of application, none of them can provide comprehensive botnet detection, fulfilling all of the detection requirements and providing a foundation for successful defence against modern botnets. Evidently the dynamic nature of bots and botnets requires an approach to botnet detection that would consider not just one characteristic of botnets but a variety of them, covering every aspect of the botnet life cycle. Some of the latest research efforts have therefore been directed at the development of novel combinations of detection approaches, which we look at in the next section.

4 Collaborative Botnet Detection

Faced with the many challenges of detecting modern botnets, researchers turned their efforts toward development of novel collaborative classes of detection approaches that integrate multiple principles of botnet detection. The main hypothesis behind these methods is that it is possible to provide higher efficiency and effectiveness of detection by correlating the findings of independent detection entities.

The general approach for correlating aspects of behaviour observed by various sensors is based on ideas presented by Strayer et al. [27], Oliner et al. [17] and Flaglien et al. [4], which extend older proposals made by Cuppens & Miège [3] and Ning et al. [16] for correlation of alerts in IDS systems. Using these or similar concepts, several authors have proposed botnet detection systems that correlate alerts from several detection enti-

ties. Wang and Gong have proposed frameworks for collaborative [31] and fusion [32] detection while Zeng et al. [35] proposed a combined botnet detection system.

To counteract the weaknesses of existing botnet detection architectures, Wang and Gong proposed several collaborative detection frameworks. The first framework [31] represents a hierarchical collaborative model that incorporates several independent detection systems, that use bot-related information from multiple sources such as network traffic, client computer internals and deployed honeypots. The second proposed framework [32] introduced the idea of combining multi-source information fusion with a collaborative detection framework. The framework envisions combining bot-related information originating from several sources (traffic monitors, IDSs, other botnet detection systems, firewalls) in order to determine the presence of a botnet within the monitored network. However the authors have not deployed or experimentally evaluated the proposed frameworks.

Zeng et al. proposed a hybrid detection approach [35] that achieves botnet detection by combining host- and network-level information. The approach is based on the assumption that two sources of bot-related information will complement each other in making accurate detection decisions. Their system first identifies suspicious hosts by discovering similar behaviours among hosts using network-flow analysis, and then validates the identified suspects to be malicious or not by scrutinizing their in-host behaviour. The approach promises independence from the C&C protocol, topology and content of transmitted data. The main limitation is that it operates in time windows, which prevents it from providing real-time detection and makes it vulnerable to time-based evasions. Additionally the approach did not use the opportunity of including bot-related information originating from other sources, leaving space for further improvements.

The systems reported on by these research groups and others, although mainly proofs of concept, demonstrated that systems which combine information from multiple sources can achieve significantly increased accuracy in recognising malicious behaviour on a network wide scale. This sets a milestone for a new direction in the field of malware detection informally known as a collaborative detection.

5 The ContraBot Framework

The ContraBot framework represents a novel systematic approach to the detection and mitigation of botnets. Following the principles considered by the research groups presented in the previous section [27, 17, 4, 31, 32], ContraBot belongs to the emerging class of collaborative botnet detection approaches that integrate multiple principles of botnet detection, in order to provide more efficient and effective detection. The basic scientific hypothesis behind our method is that correlating the observations and analyses from client and network entities combined with in-depth analysis of harvested code will significantly improve the botnet classification ability, in comparison to todays state-of-the-art methods.

The ContraBot framework utilises several functional entities, as illustrated in Figure 1. A set of network sniffers placed within the network collect and pre-process network traffic data, while a set of activity monitors within the clients collect and pre-

process information about client activity. The pre-processing is necessary in order to reduce the amount of data and also to allow selective analysis of particular traffic and/or client behaviour patterns. The output of this pre-processing is passed to a set of one or more Correlators, where it is analysed to reveal patterns of similar behaviour in different hosts and different parts of the network. Unusual patterns of activity, which may indicate an attack, will lead to the harvesting of portions of code from the hosts (and the associated network traffic), so that these can be further analysed by entities which investigate the code for malicious effects. In a similar way, Client distribution analysis entities analyse modules, apps and other forms of software fetched by the Clients from the network, so that well-known malicious software can be disabled or removed as in a traditional AV (anti-virus) system.

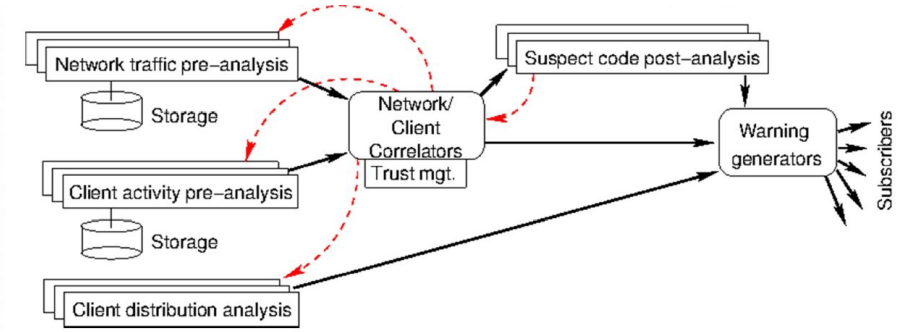

Fig. 1. Architecture of the proposed analysis system.

The ContraBot framework will require input from a wide range of sources, including sensors installed by end users, ISPs and backbone network providers. The producers and consumers of this input often belong to different, possibly competing, organisations that employ different types of sensors, so it is important that all parties can evaluate the trustworthiness of the input they receive. The Correlation Framework will therefore include a trust management component that aims to establish the trustworthiness of input based on both direct experiences with the individual input provider and reputation ratings exchanged between the different Correlators in the Correlation Framework.

If the Correlators, distribution analysis entities or code analysis entities detect signs of malicious software, they pass this information to a sub-system which generates Warnings for distribution to Subscribers of the anti-botnet service. This allows the subscribers to initiate various counter measures, e.g. walled-gardens. In addition to warning messages, information (indicated by the dashed red arrows) is architecture of-passed back through the system, so that the analysis can be adjusted to focus more accurately on recognisable malicious activity. Both the network sniffers and client analysis entities have access to a (possibly distributed) storage system, to facilitate recording and subsequent in-depth analysis of detected threats.

5.1 Network Traffic Sniffing and Pre-analysis

The network sniffers that monitor and pre-analyse the traffic are entities operating in core networks, e.g. in an ISP backbone, and must therefore be capable of handling traffic at high data rates. For the realization of these sniffers FPGA (Field-programmable gate array) based network interface cards will be used. Such cards can guarantee high-speed packet capture with no packet loss, and they can also be configured to pre-filter/pre-process the packets according to some predefined parameters.

Detection of malicious traffic in core networks is not a new concept. Many approaches, ranging from signature based methods to more sophisticated methods based on machine learning, have been proposed and can be reused to a large extent. However, as described above, we plan to make the pre-analysis adaptive in the sense that it will receive information from the Correlator about threats observed elsewhere. Hence a substantial part of the research and performance optimisation regarding the network sniffers will focus on how existing principles can be used in an adaptive setup. In addition, we will investigate if further performance and/or precision can be achieved by utilising the available FPGAs.

5.2 Client Activity Monitoring

Client activity will be monitored by recording features related to changes in the file system, registry (or similar configuration database) and use of network connections in the individual hosts. This type of monitoring is comparable to what many client IDS systems already do. The collected features will be used as the basis for clustering analysis, and the results of this local analysis will be sent to the Correlators for further investigation of possible correlated activities in separate hosts. Research into performance optimisation of this process will be needed in order to reduce the computational load and storage requirements on the clients as far as possible, while retaining sufficient information for our analysis.

5.3 Client Distribution Analysis

A significant problem in many of the newest botnets is that malware may be distributed by non-traditional routes which will not be detected by traditional on-access AV scanning of mail and web access. Within the last few years, for example, distribution via Facebook spam, YouTube and on-line markets providing apps for smartphones has been observed, and there is every reason to believe that this trend will continue, expanding to cover other platforms. In addition, designers of malware now have access to almost unlimited computing power via cloud computing. This enables them to produce huge numbers of variants of each item of malware, so as to avoid malware collection and analysis by ensuring that each variant only appears very rarely and thus avoids rousing suspicion. We believe that correlation analysis, combined with in-depth post-analysis of harvested code, can mitigate this problem significantly, as we expect to be able to detect behavioural similarities among code items which appear not to be significantly related.

5.4 Correlation Framework

The Correlation Framework is one of the most important elements of our framework, as it should realize correlation of observation generated by various sensors. The Correlation Framework will be realized as two components: A Behavioural Analysis Component and a Trust Management Component.

The behavioural analysis component will analyse the filtered and pre-processed feature data provided by the client and network sensors, in order to search both for network-network and client-network correlations. It will further develop ideas proposed by a variety of research groups, as reported in Oliner et al. [17], Wang et al. [31, 32], and Flaglien et al. [4], amongst others. We intend to extend this previous work to cover more scenarios and platforms.

The trust management component has a task of assessing the trustworthiness of data and alerts from the different sensors within the system. Proposals for assessing the information received from the different sensors have been based on trust evaluation of the individual data sources [14], collaborative filtering techniques employing robust statistics [23] or extended with trust metrics [34], or filtering to eliminate outliers in the received data [37]. None of these techniques, however, have been developed for the environment envisaged in the ContraBot infrastructure, where sensor data and alerts are shared among separate collaborating organisations. We therefore need to investigate hybrid techniques, where the results of the different filtering techniques are incorporated with different weights, e.g. the weight of the trust evaluation of the data provider is higher when evaluating input from end-users, because less can be assumed about their motives and competence levels.

The trust management component will assess the trustworthiness of the different sensors using both direct experience and indirect experience through a reputation system. Direct experience will be based on both content based filtering using data contained in the observations, such as the make, model and version of the sensor, and collaborative filtering where the output from one sensor is compared with output from other sensors in order to determine whether the sensor agrees with the majority. The trust management component will monitor the results of this trustworthiness assessment over time and apply the results to a trust evolution function, such as the one proposed for the Wikipedia Recommender System [11]. Moreover, the trust management component will build indirect trust in sensors by exchanging direct trust assessments (reputation scores) with the other trust management components in the Correlation Framework. This will significantly reduce the "cold start" problem and accelerate trust formation among components in the framework.

5.5 Testing

To facilitate tests of algorithms and architectural design, a closed and controlled Internet-like test network, in which experiments can be repeated, is needed. Based on the sniffing equipment, a testbed emulating a subset of the Internet will be designed and implemented. The testbed will facilitate botnet life cycle analysis with real and artificial bots, where spread patterns, infection times, etc. can be studied, both from a client and a network perspective. Furthermore, the testbed will facilitate the development of the

client and network entities, as it will facilitate tests in realistic (emulated) conditions. It will also be used to test the scalability of given system architecture proposals. It may here prove possible to reuse substantial parts of the already existing emulab testbed system (http://www.emulab.com) with modifications. We believe that having a testbed on which the framework can be implemented and tested will allow us to achieve a precise, reliable and performance optimised product. Furthermore, as the threats evolve continually, it is important to have a test setup where updated mechanisms and parameters can be tested and evaluated.

6 Discussions and Future Work

In this paper we have presented an overview of existing botnet detection techniques and we have analysed their ability to cope with challenges of detecting modern botnets. Furthermore, we have shown that there is a need for new and more comprehensive detection approaches in order to provide efficient and effective neutralisation of botnets. Finally we introduced ContraBot a novel collaborative botnet detection approach.

To our best knowledge the ContraBot framework could possibly be the first extensive attempt to take counter botnet research to a systematic level, providing the basis for a more comprehensive botnet defence system. The botnet defence system envision by the ContraBot framework will aggregate simultaneous observations from different types of sensors, such as network sniffers and client monitors to identify suspect activities and possibly initiate appropriate counter measures.

The Contrabot framework is partly based on principles similar to existing collaborative botnet detection approaches such as [27, 17, 4, 31, 32], but it has a number of advantages: First, the ContraBot will employ traffic analysis in the core network, providing protection for a broader set of end-users. Secondly, our proposed set-up will combine information not only from network and client levels but also from in depth analysis of harvested code in order to improve the detection accuracy even further. Third, the proposed system will provide flexibility of including diverse end-user platforms through development of appropriate client-based analysis entities. Fourth, our system will also introduce the feed-back mechanism. This will provide adaptivity of network- and client-based pre-analysis entities to the bot-related information generated by correlating findings from other sources. This information allows the system to dynamically adapt to changes in behaviour of bots and botnets.

An important future step for testing and evaluating the framework is the development of a prototype system to demonstrate the technical approaches for reliable detection and elimination of botnets. Such a prototype system needs to be systematically evaluated using a suitable testbed. We believe a testbed should be developed specifically for this purpose, capturing salient features of large-scale networks. The prototype could be a first step twoards the development of a full-scale botnet defence platform.

References

1. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for DNS. In: Proceedings of the 19th USENIX Security Symposium (Security'10). USENIX Association (Aug 2010)

2. Choi, H., Lee, H.: Identifying botnets by capturing group activities in DNS traffic. Journal of Computer Networks 56, 20–33 (2011)
3. Cuppens, F., Miège, A.: Alert correlation in a cooperative intrusion detection framework. In: Proceedings of IEEE Symposium on Security and Privacy. pp. 202–215 (May 2002)
4. Flaglien, A., Franke, K., Årnes, A.: Identifying malware using cross-evidence correlation. In: Peterson, G., Shenoi, S. (eds.) Advances in Digital Forensics VII, IFIP ACIT, vol. 361, chap. 13, pp. 169–182. IFIP (2011)
5. Goebel, J., Holz, T.: Rishi: Identifying bot-contaminated hosts by IRC nickname evaluation. In: HotBots'07: Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, Mass. USENIX Association (Jun 2007)
6. Grizzard, J.B., Sharma, V., Nunnery, C., Kang, B.B., Dagon, D.: Peer-to-peer botnets; Overview and case study. In: HotBots'07: Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, Mass. USENIX Association (Jun 2007)
7. Gu, G., Zhang, J., Lee, W.: BotSniffer: Detecting botnet command and control channels in network traffic. In: NDSS'08: Proceedings of the 15th Annual Network and Distributed System Security Symposium, San Diego. Internet Society (Feb 2008)
8. Gu, G., Perdisci, R., Zhang, J., Lee, W.: Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proceedings of the 17th conference on Security symposium. pp. 139–154 (2008)
9. 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, San Jose, California. pp. 167–182. USENIX Association (Jul 2007)
10. Hogben, G. (ed.): Botnets: Detection, measurement, disinfection and defence. Tech. rep., ENISA (2011)
11. Jensen, C., Korsgaard, T.: Dynamics of trust evolution: Auto-configuration of disposiional trust dynamics. In: Proceedings of the International Conference on Security and Cryptography (SECRYPT 2008), Porto, Portugal. pp. 509–517 (Jul 2008)
12. Karasaridis, A., Rexroad, B., Hoeflin, D.: Wide-scale botnet detection and characterization. In: HotBots'07: Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, Mass. USENIX Association (Jun 2007)
13. Lu, W., Rammidi, G., Ghorbani, A.A.: Clustering botnet communication traffic based on n-gram feature selection. Computer Communications 34, 502–514 (2011)
14. Marsh, S.: Formalizing Trust as a Computational Concept, PhD thesis. Ph.D. thesis, University of Stirling, Dept. of Computer Science and Mathematics (1994)
15. Masud, M., Al-Khateeb, T., Khan, L., Turaisingham, B., Hamlen, K.: Flow-based identification of botnet traffic by mining multiple log file. In: Proceedings of the International Conference on Distributed Frameworks and Applications (DFMA), Penang, Malaysia (2008)
16. Ning, P., Cui, Y., Reeves, D.S.: Constructing attack scenarios through correlation of intrusion alerts. In: Proceedings of CCS'02. pp. 245–254. ACM (Nov 2002)
17. Oliner, A.J., Kulkarni, A.V., Aiken, A.: Community epidemic detection using time-correlated anomalies. In: Jha, S., Sommer, R., Kreibach, C. (eds.) RAID 2010. pp. 360–381. No. 6307 in Lecture Notes in Computer Science, Springer-Verlag (2010)
18. Porras, P., Saidi, H., Yegneswaran, V.: A multi-perspective analysis of the Storm (peacomm) worm. Tech. rep., SRI International (2007), available at: `http://www.cyber-ta.org/pubs/StormWorm/report`
19. Porras, P., Saidi, H., Yegneswaran, V.: Conficker C analysis. Tech. rep., SRI International (2009), available online: `http://mtc.sri.com/Conficker/addendumC/index.html`
20. Ramachandran, A., Feamster, N., Dagon, D.: Revealing botnet membership using DNSBL counter-intelligence. In: SRUTI'06: Proceedings of the 2nd Workshop on Steps to Reducing

Unwanted Traffic on the Internet, San Jose, California. pp. 49–54. USENIX Association (Jun 2006)
21. Roesch, M.: Snort – lightweight intrusion detection for networks. In: Proceedings of Usenix LISA'99. USENIX Association (1999)
22. Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., Hakimian, P.: Detecting P2P botnets through network behavior analysis and machine learning. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust, Montreal. IEEE (Jul 2011)
23. Setia, S., Roy, S., Jajodia, S.: Secure data aggregation in wireless sensor networks. In: Lopez, Zhou (eds.) Wireless Sensor Networks Security (2008)
24. Shin, S., Xu, Z., Gu, G.: EFFORT: Efficient and effective bot malware detection. In: Proceedings of 31st Annual IEEE Conference on Computer Communications (INFOCOM'12), Orlando, Florida. IEEE (Mar 2012)
25. Sinclair, G., Nunnery, C., Kang, B.B.: The Waledac protocol: The how and why. In: Proceedings of International Conference on Malicious and Unwanted Software (MALWARE) (2009)
26. Stinson, E., Mitchell, J.C.: Characterizing bots' remote control behavior. In: Lee, W., Wang, C., Dagon, D. (eds.) Botnet Detection, Advances in Information Security, vol. 36, pp. 45–64. Springer (2008)
27. Strayer, W.T., Lapsely, D., Walsh, R., Livadas, C.: Botnet detection based on network behaviour. In: Lee, W., Wang, C., Dagon, D. (eds.) Botnet Detection, Advances in Information Security, vol. 36, pp. 1–24. Springer (2008)
28. Symantec Inc.: Symantec global internet security threat report, trends for 2010. Security Report XVI, Symantec Inc. (Apr 2011)
29. Symantec Inc.: Counterclank bot. Tech. rep., Symantec Inc. (2012), available online: `http://www.symantec.com/security_response/writeup.jsp?docid=2012-012709-4046-99`
30. Villamarin-Salomon, R., Brustoloni, J.: Identifying botnets using anomaly detection techniques applied to DNS traffic. In: Proceedings of 5th IEEE Consumer Communications and Networking Conference (CCNC 2008). pp. 476–481 (2008)
31. Wang, H., Gong, Z.: Collaboration-based botnet detection architecture. In: Proceedings of 2nd International Conference on Intelligent Computational Technology and Automation, Zhangjiajie, China (2009)
32. Wang, H., Hou, J., Gong, Z.: Botnet detection architecture based on heterogeneous multi-sensor information fusion. Journal of Networks 6(12), 1655–1661 (Dec 2011)
33. Wang, P., Sparks, S., Zou, C.C.: An advanced hybrid peer-to-peer botnet. In: HotBots'07: Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, Mass. USENIX Association (Jun 2007)
34. Weng, J., Miao, C., Goh, A.: Improving collaborative filtering with trust-based metrics. In: Proceedings of ACM Symposium on Applied Computing (SAC). pp. 1860–1864. ACM, New York, NY, USA (2006)
35. Zeng, Y., Hu, X., Shin, K.G.: Detection of botnets using combined host- and network-level information. In: Proceedings of 40th International Conference on Dependable Systems and Networks (DSN) (2010)
36. Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., Luo, X.: Detecting stealthy P2P botnets using statistical traffic fingerprints. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks (DSN), Hong Kong. pp. 121–132. IEEE/IFIP (Jun 2011)
37. Zhang, Y., Meratnia, N., Havinga, P.: Outlier detection techniques for wireless sensor networks: A survey. In: IEEE Communications Surveys and Tutorials (2010)