Towards Efficient and Privacy-Preserving Network-Based Botnet Detection Using Netflow Data

S.Abt and H.Baier

Center for Advanced Security Research, Faculty of Computer Science,
Hochschule Darmstadt, Germany
e-mail: {sebastian.abt|harald.baier}@cased.de

Abstract

Botnets pose a severe threat to the security of Internet-connected hosts and the availability of the Internet's infrastructure. In recent years, botnets have attracted many researchers. As a result, many achievements in studying different botnets' anatomies have been made and approaches to botnet detection have been developed. However, most of these approaches target at botnet detection using raw packet data. While this data provides the most complete view on botnet induced traffic, it usually cannot efficiently be collected at large network nodes transferring multi-Gigabits per second. Additionally, a deep inspection of network packets endangers the users' privacy. In order to solve these problems different detection methods based on Netflow data have been proposed. To contribute to advances in Netflow-based botnet detection research, we first give an overview of currently known approaches and compare their advantages and disadvantages. We then argue that Netflow-based detection requires the availability of a reference data set based on real data and present a modular data collection environment that is able, amongst others, to generate Netflow data at an ISP node. Finally, we present our vision of a future botnet detection framework based on Netflow data.

Keywords

Botnet detection, Netflow data, reference data set, large network operator, privacy.

1. Introduction

Together with this growth of the Internet a sub-culture constituting the *Internet underground economy* has evolved, aiming at doing business by abusing the Internet's open architecture and structure as well as ingenuous Internet users. The means this underground economy uses for their infamous purposes are manifold. For instance, *phishing* is used to steal user credentials. *Spam e-mails* are used to distribute phishing URLs or *malware* (e.g. worms, trojans, key loggers, scareware) in order to infiltrate a user's computer. Infiltrated computers are used to further distribute phishing URLs or malware or to launch *denial of service (DoS) attacks* (Mirkovic and Reiher, 2004), (Freiling et al., 2005), (Thing et al. 2007), causing severe financial damage (Patterson, 2002). Often, these infiltrated computers are remotely controlled by a miscreant and are usually referred to as *bots*. Many bots grouped together and equally controlled by an attacker are called *botnet*. Botnets have evolved to become one of the biggest annoyances large network operators have to cope with (Arbor Netwokrs, 2005-2011).

Since some years, the question on how to detect botnets has attracted various researchers and different approaches have been published (Freiling et al., 2005), (Gu et al., 2007), (Gu et al., 2008), (Gu et al., 2008), (Racine, 2004), (Karasaridis et al., 2007), (The Honeynet Project, 2005), (Livadas et al., 2006), (Strayer et al., 2006), (Binkley and Singh, 2006), (Ramachandran et al., 2006), (Goebel and Holz, 2007). However, almost all of this work targeted at host-based detection or utilized full packet data for network-based approaches. Approaches utilizing full packet data are commonly referred to as performing deep packet inspection (DPI). Unfortunately, DPI by definition only inadequately takes privacy aspects and efficiency considerations into account. With DPI, recording and analysis of full network traffic is required, which efficiency-wise is not feasible on network links transferring multiple Gigabit/s. Additionally, illegitimate access to DPI-based systems or data recorded by them effectively uncovers possibly private communication (e.g. personal email, HTTP sessions) and thus seriously endagers the communicating parties' privacy. Therefore, the applicability of DPI in high-traffic environments is restricted. We believe that this classical proceeding does not succeed very well in defending against botnets: Host-based detection alone obviously seems to be ineffective as malware and virus detection systems have been released for years, yet the botnet threat is growing, and clearly both efficiency and privacy are essential requirements for network-based detection systems.

In order to comply with these requirements, network-based botnet detection not only has to be based on efficient detection algorithms, but also has to make use of data sets that can efficiently be gathered and protect users' privacy. Data that fulfill both demands are *Netflow data*.

The contributions of the paper at hand are as follows:

1. We aim at reviewing the state of the art in botnet detection with a focus on its applicability to large networks (e.g. an Internet Service Provider). We come up with a comparison of currently proposed approaches.

2. Based on this discussion, we conclude that in order to advance research in this area, an openly available Netflow reference data set satisfying both data efficiency and user privacy has to be created. Up to now such public data set is missing (we speculate about the reasons for this in Section 4). We contribute to this area of research by presenting an extended, network operator centric and easy to manage honeynet environment suitable for collecting botnet induced Netflow data.

3. Furthermore, we develop a roadmap to get a Netflow-based detection appliance, which may easily be integrated in network operators' infrastructures and which may lead to a global botnet early warning system.

The outline of the remainder of this paper is as follows: Section 2 gives an overview on the fundamentals of botnets and introduces Netflow data. Section 3 reviews existing approaches in botnet detection and discusses their issues. After that, Section 4 describes our proposal for a data collection environment of Netflow data and our experiences after its deployment. In Section 5 we propose our roadmap to come to a Netflow-based botnet detection appliance. Finally, Section 6 concludes this paper.

2. Botnets and Netflow Data

This section introduces the basic facts on botnets and on Netflow data that are necessary to present our approach on Netflow-based botnet detection and our collection center of Netflow data.

2.1. Botnets

Botnets are based on a herd of compromised computers that are called bots and are under control of a *botmaster*. To create a botnet, a botmaster has *(1)* to *infect* remote hosts, *(2)* to decide on the *command and control* structure and protocol used within the botnet, and *(3)* to launch further *malicious activities*. These three aspects characterize a botnet and will be described next.

Host infection is the process of compromising a victim's computer. Common attack vectors to infect a device are *remote exploitation*, infection through *mail attachments*, *drive-by downloads* (Li et al., 2009), (Feily et al., 2009) and *direct infection*. Remote exploitation shares commonalities with worm propagation (Adeel et al., 2009). Attackers scan possible targets for known vulnerabilities that can be used to compromise a particular victim's computer by injecting malicious code. Mail attachments are distributed via spam mails. By launching the binary attached to a mail, the bot installs itself on the client device. With drive-by downloads, a user's device is infected by simply browsing websites containing malicious content that targets at the web browser or its plugins (Provos et al., 2007). Direct infection usually happens by exchanging infected storage devices like USB devices, compact flash cards, CD-Roms, etc.

After infection, a host can participate in a botnet. For coordination of a bot's activity, the bot has to receive commands from its botmaster via a *command and control (C&C) channel*. The C&C topology can either be *centralized*, i.e. a hub and spoke topology with one or more fixed rendezvous points bots can connect to, be a *peer-to-peer* architecture where each bot can act equally as client as well as server to forward messages, or be *randomly* structured showing no fixed topology, i.e. no single bot knows about the presence of more than one single infected other machine (Bailey et al., 2009).

Finally, compromised hosts that are connected to a botnet can be abused for various *malicious activities* upon the request of the botmaster. These activities include, but are not limited to, *DoS attacks*, *host infections*, *distributing spam e-mails*, *spying*, *hosting or sharing (illegal) data*, *data stealing*, *bandwidth trading*, *proxying*, and *click fraud* (Govil, 2007), (Liu et al., 2009), (Thing et al., 2007).

2.2. Netflow Data

In a computer network, a *Netflow record* or, in short, *flow* is a statistic derived from an unidirectional data stream between two communication systems that shares common attributes at the network layer (L3) and the transport layer (L4). A flow comprises packets offering the same source and destination IP addresses, source and destination port numbers, and layer 4 protocol type number at a specific period of time. More formally, we write a flow occuring at a specific point in time t as a 5-tuple, which we denote by f_t, i.e. we have f_t = *(srcIP, dstIP, srcPort, dstPort, L4Proto)*. The attributes constituting a flow f_t are called *flow keys* and serve as unique identifiers during flow creation. By filling a *flow cache* with flow keys as well as non-key attributes and statistics, Netflow records are created on intermediate network nodes (e.g. routers or switches) during packet forwarding (Cisco Systems, 2007). After creation, Netflow records are exported to a *flow collector* either after expiration of a pre-defined timer or if a communication channel represented by a flow is closed (e.g. after having seen a TCP FIN or TCP RST packet). Alongside with the flow keys, further non-key attributes are exported. The most interesting ones being the amount of bytes and packets transferred and, in case of a TCP channel, TCP flags set. Netflow records can be collected and exported in different format versions. The currently most commonly used version is Cisco System's proprietary Netflow version 5 (Cisco Systems, 2007). More recently, the Internet Engineering Task Force (IETF) standardized an extended, more flexible and freely avaible version called IPFIX (Claise, 2008).

The definition given above stresses our point about Netflow data fulfilling both, efficiency and privacy requirements predominant in large networks. First, Netflow records can be efficiently created in high-traffic environments as creation is inline with packet forwarding. This effectively transforms an ISPs packet forwarding network into a sensor network without the need for additional capital investment. Second, neither additional header information, nor any payload information is inspected during creation or exported to a flow collector. Hence, Netflow records contain very limited amount of potentially sensitive information, if any[1]. Thus, utilizing Netflow data by definition protects the user's privacy much better than any content-inspecting approach could do as no payload is touched. Additionally, due to not capturing the packets' headers or payloads, using Netflow significantly (see Sect. 4.2) reduces the amount of data that has to be analyzed as well as the processing power required for analysis. Hence, using Netflow data for botnet detection seems to be an interesting and promising approach, both, privacy- and efficiency-wise.

3. Botnet Detection

Detection of botnets has been of interest to various researchers and companies around the world and still is. Common techniques to counter this threat can be classified according to detection methodology, locality, and data source used. *Detection methodology* can either be *signature-based* or *anomaly-based* (Liu et al., 2009), (Feily et al., 2009), (Scarfone and Mell, 2007), (Bailey et al., 2009). The *locality* of botnet detection can either be *host-based* or *network-based* (Scarfone and Mell, 2007). Finally, the *data source* available for botnet detection or detection of malicious activity stemming from botnets can be *raw packet data*, *Netflow data*, *application traces* (e.g. command executions), *system log files*, or *bot binaries* (Bailey et al., 2009). We will next classify current approaches that specifically

[1] IP address data could be regarded as sensitive information.

address origin botnet detection and represent current state of the art according to these criteria. Afterwards we discuss their eligibility for use in high-traffic environments.

Approach	Methodology	Locality	Data source
(Freiling et al., 2005)	sign.-based	host-based	bot binaries
(Racine, 2004)	sign.-based	netw.-based	Netflow data
(Gu et al., 2007), (Gu et al. 2008a), (Livadas et al., 2006), (Strayer et al., 2006), (Binkley and Singh, 2006)	anom.-based	netw.-based	raw packet data
(Ramachandran et al., 2006)	anom.-based	host-based	log files
(Goebel and Holz, 2007)	sign.-based	netw.-based	raw packet data
(Gu et al., 2008b), (Karasaridis et al., 2007)	anom.-based	netw.-based	Netflow data + add. information

Table 1: Summary of classification of current botnet detection approaches

3.1. Classification of Current Approaches

(Freiling et al., 2005) introduced a root-cause methodology to detect DDoS attacks launching botnets by collecting bot binaries using a honeypot (The Honeynet Project, 2005) and infiltrating the botnet by connecting to the botnet's IRC C&C channel using a “silent drone”. We classify this as host- and signature-based approach utilizing bot binaries.

Racine proposed making use of behavioral characteristics of bots (Racine, 2004). He found that IRC-based bots were mostly idle, only responding to commands from their botmaster. To characterize IRC behavior, Racine made use of Netflow data. Thus, this is a network- and signature-based detection approach utilizing Netflow data.

(Strayer et al., 2006), (Livadas et al., 2006) proposed a multi-step network-based approach using machine learning techniques and temporal clustering, i.e. anomaly-based, on raw packet data. This approach does not make use of any packets' payloads, however, due to its use of packet inter-arrival times during temporal clustering, this approach will not work with Netflow data.

(Binkley and Singh, 2006) proposed a network- and anomaly-based algorithm combining IRC message statistics and TCP work weight. By examining IRC messages, this approach relies on packets' payloads and thus utilizes raw packet data. Furthermore, it will not work with encrypted communication.

(Ramachandran et al., 2006) proposed using DNS blacklist counter-intelligence to find spam-generating botnet members. Their approach is based on the insight that botmasters have to determine their bots' blacklist status. This is a host- and anomaly-based approach utilizing system log files (i.e. a DNS server’s log files).

(Goebel and Holz, 2006) proposed a network- and signature-based approach, Rishi, to detecting IRC-based botnets. Rishi uses n-gram analysis to identify patterns of nicknames commonly used by botnets. This approach is limited to IRC-based C&C and cannot detect encrypted communication. As it analyzes IRC messages, this approach makes use of raw packet data.

In (Gu et al., 2007), Gu et al. proposed a system called BotHunter. BotHunter is a botnet detection system that uses IDS-driven dialog correlation. BotHunter is a network- and anomaly-based approach utilizing raw packet data. In (Gu et al., 2008a), Gu et al. proposed a more advanced detection system called BotSniffer. BotSniffer aims at detecting spatial-temporal correlation and similarity patterns, i.e. crowd-like behaviour, in network traffic. As BotHunter, BotSniffer is a network-based anomaly-based detection system utilizing raw packet data. Further, Gu et al. propose a system called BotMiner (Gu et al., 2008b) that extends BotSniffer by clustering similar communication plane (C-plane) traffic and activity plane (A-plane, i.e. malicious) traffic. After that, cross-cluster correlation is performed to identify suspicious hosts. BotMiner is protocol and structure independent and makes use of Netflow data in the C-plane. However, A-plane clustering is conducted using raw packet data. BotMiner is a network- and anomaly-based system.

An approach utilizing Netflow data and external triggers is proposed by (Karasaridis et al., 2007). Karasaridis et al. studied the detectability of IRC botnet controllers on backbone networks by calculating distances between monitored flow data and a pre-defined IRC traffic profile. Their system utilizes external triggers (e.g. IDS alerts, scans, spam e-mails, system logs) to identify malicious hosts. After that, these hosts' Netflow data is analyzed to find candidate control conversations (CCC) with C&C hosts. Due to not depending on packets' payloads during analysis, this system is capable of detecting encrypted communication as well.

3.2. Issues in high-traffic environments

As this classification and discussion of current botnet detection approaches shows, besides (Racine, 2004) all approaches make either (additional) use of raw packet data or are host-based systems. This reflects current state of the art in botnet detection research.

These methods, however, are not feasibly applicable to large networks and high-traffic environments for the following reasons:

1. Host-based systems require the network operator to have access to the end users' devices, which is usually not the case, or force the end user to run specific detection software on all of its devices in order to gain access to the Internet. However, due to anti-discrimination regulations, in most countries the latter is not possible.

2. Capturing and analyzing raw packet data at transfer rates of multi-Gigabits per second in real-time is technically infeasible and additionally puts high financial burden on the network operators. Further, such systems are

restricted to the traffic as seen on a single network node and hence can't benefit from a network operator's spatially large footprint.

3. Categorically inspecting raw packets puts the end user's privacy at risk and contradicts current jurisdiction in many countries.

A promising alternative to raw packet data that has attracted the broader field of network attack detection and high-speed intrusion detection research (cf. (Abt, 2009)) is the use of Netflow data, which satisfies the following requirements:

1. As no information on a packet's payload is revealed, Netflow data do contain only a limited amount of sensitive information, if any. Hence, categorically inspecting Netflow data does not conflict with the end users' privacy.

2. Due to early data reduction at the collection point, flow data can efficiently be processed in near real-time and thus increases detection efficiency.

3. Netflow-based botnet detection comes almost free of additional capital investment as network operators can utilize their existing network devices as distributed sensor nodes.

4. Concurrently using network devices as sensor nodes provides a wide view on current network activity, leveraging the detection of malicious events caused by botnets.

Interestingly, besides (Racine, 2004), a botnet detection system heavily utilizing Netflow data has been proposed by researches at AT&T Labs (Karasaridis et al., 2007), which we believe emphasizes our view on the feasibility of systems based on Netflow data in large networks. Additionally, (Coskun et al., 2010) utilizes Netflow data for identifying “mutual contacts” of bot infected hosts that have originally been detected using different detection strategies.

4. A Reference Data Collection Environment

As discussed before, efficiency and privacy requirements have to be met by botnet detection approaches feasible for use in high-traffic environments. Netflow data honour these requirements. However, only few approaches (Gu et al., 2008b), (Racine, 2004), (Karasaridis, 2007) utilize these data. We believe that the reason for this is twofold: first, no publicly available data set of these data sources exists that can be used to develop, train and test algorithms. Second, such data can best be won in a target environment and therefore cooperation of high-traffic network operators is essential. Unfortunately, however, cooperating with such organizations in this context is usually difficult as collecting Netflow data possibly requires re-configuration of an organization's infrastructure, which binds (human) resources and possibly puts its infrastructure at additional risk. To counter these issues, we've developed a modular data collection environment that aims at reducing a network operator's workload in participating and generates data that can be used for algorithm development and evaluation. We will describe this system in Section 4.1 and present first experiences in Section 4.2.

4.1. Architecture

The system we developed is depicted in **Error! Reference source not found.**. It is currently hosted with an Internet Service Provider supporting this work and consists of several components that will be explained next. It can easily be used to integrate other Internet Service Providers as well as gateways of large institutional networks (e.g. of companies or universities) in the data collection process.

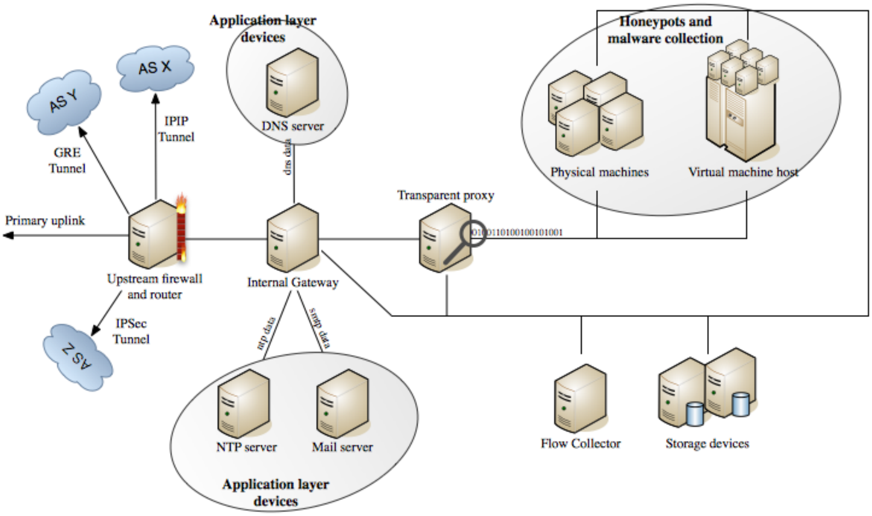

Figure 1: Scheme of the data collection environment. Different autonomous systems (AS) can easily be added by using IP tunneling mechanisms

Upstream firewall and router The upstream firewall and router is used as the network termination point of this setup that connects the data collection system to the rest of the world. The primarily functionality of this component is as follows: first, it protects the Internet from getting attacked by bot-infected hosts from within the system and limits the intense of attacks that are launched from the inner by filtering and rate-limiting all outgoing traffic. Second, this component terminates virtual upstream connections to other network operators by using standardized tunneling protocols like IP security (IPSec), generic routing encapsulation protocol (GRE) or IP-in-IP tunneling, facilitating the virtual integration of this data collection environment into other networks.

Internal Gateway The internal gateway acts as gateway between different subnets the honeypots reside in and can provide dynamic IP address assignment by use of DHCP. Additionally, the internal gateway is used to re-route specific protocols to special application layer devices to capture side-effects caused by botnet activity and to simulate host infections in a controlled, clean-slate environment (cf. (Gorecki, 2007)).

Transparent proxy The transparent proxy device creates and collects netflow data as well as raw packet traces (as used for DPI) for the traffic observed.

Application layer devices Application layer devices are used to handle specific suspicious application layer protocols (e.g. SMTP, NTP, DNS) found in the infiltrated hosts' network traffic and thus can produce system log files typically found in such environments.

Honeypots The honeypots are used to expose itself as potential victims and mostly run on virtual or physical machines as we aim at capturing network traces of bot infection processes and not only at capturing malware samples, as is the focus of mwcollect (Freiling et al., 2005) or nepenthes (Baecher et al., 2006)). In general, basically any operating system can be setup on these nodes either physically or virtualized with varying network configurations simulating both, higher-speed systems with static IP addresses as well as lower-speed dial-up connections with changing IP addresses.

Storage devices The storage devices provide redundant network attached storage for virtual machine images and are used to safely store the collected data.

Our data collection framework, as described above, has the following advantages and novelties in contrast to similar data collection approaches (Freiling et al., 2005), (The Honeynet Project, 2005), (Rajab et al., 2006):

- The system has been carefully designed in cooperation with an Internet Service Provider and hence integrates very well in large networks.

- The system can be easily integrated in other than the hosting ISP's environments. This potentially raises the footprint of the detection system as IP address ranges from various network operators from geographically distant regions, and hence different time zones, can be used to collect data (Dagon et al., 2006).

- The system does not interfere with legitimate users' data and thus does not raise any privacy concerns during data collection.

- The system is capable of capturing Netflow data as well as raw packet data, system log files and application layer data.

4.2. Data Collection and First Experiences

As noted above, development of this system has been supported by an Internet Service Provider and a first installation has been deployed with this ISP for a period of 19 days (without any virtual connection to any other ISP). During this time, we ran a darknet consisting of three subnets spanning three different /8 prefixes (85/8, 95/8, and 62/8). Each subnet consisted of 32 IP addresses, i.e. a /27 subnet. We run 18 honeypot instances at a time during this time frame. The operating systems used varied between Microsoft Windows XP, Windows Vista, Windows 7, and Debian GNU Linux 5.0. Using this configuration, we collected approximately 30 GB of raw packet data and 4 GB of Netflow data. These network traces are completely comprised of suspicious and/or malicious traffic as the devices run within the subnets were not expected to cause any network activity. By comparing the volume of raw packet data with Netflow data collected in this time frame, one can clearly see the massive data reduction capabilities achievable by prefering Netflow data over raw packet data. In our case, the reduction was almost 81%, which clearly emphasizes the gain in both, efficiency and privacy that can be achieved using Netflow data for botnet detection.

The analysis of the collected data is still ongoing work. However, to gain a first insight in the data we used the Snort signature-based intrusion detection system to analyze the captured raw packets. The results of this analysis revealed that the captured traffic mostly consisted of network- and port-scans, worm distribution traffic as well as botnet traces with Microsoft Windows XP and Microsoft Windows Vista being the only infected systems. This is particular interesting as all operating systems were installed without any further modifications. For the Debian GNU Linux 5.0 installation we additionally added 10 credentials commonly found in SSH-scan attempts to facilitate the exploitation of these instances and the establishment of an centralized command and control rendezvous point. In contrast, the fastest exploitation of a Windows XP machine did not last longer than 1:34 minutes.

5. Advancing Netflow-based Botnet Detection

We believe that using Netflow data is both an efficient and privacy-preserving approach to botnet detection feasible for large networks and high-traffic environments. In order to advance research in this direction, the compilation of a labeled reference data set is essential. Using our data collection environment, we will next proceed at compiling such a labeled reference data set by clustering and correlating the traces we collected thus far (and will collect in future). To automate this work, we're looking forward to enhance our data collection environment by means to automatically cluster, correlate and classify network traces. Additionally, we plan to mix the won and classified malicious traces with representative traffic traces belonging to well-known benign network traffic. We will collect this benign traffic at various network demarcation points (e.g. university campus traffic, ISP data center traffic, residential dial-up customer traffic) of different network operators and behind commercial firewall systems (i.e. pre-filtered traffic traces).

Further to that, we believe that for large-scale botnet command and control detection and in order to defeat the threat emerging from botnets, the existence of an open source Netflow-based detection appliance is essential. Such an appliance and its algorithms should

1. be easy to integrate into existing service provider networks,

2. be able to effectively and efficiently detect botnet C&C traffic,

3. be self-adapting to changing conditions, such as the emergence of new bot families or C&C topologies,

4. and should provide interfaces for human-appliance interaction as well as inter-appliance interaction, i.e. the exchange of information belonging to possibly malicious activities between different network operators.

While the latter requirement seems to be especially important to obtain a Netflow-based early warning system, care has to be taken to exchange the necessary information in a privacy-preserving way and according to national jurisdiction.

Given these requirements, additional to the compilation of a state of the art reference data set, we propose studying the following topics in order to advance the area of privacy-preserving Netflow-based botnet detection and to develop a detection appliance suitable as global botnet early warning system. *Flow anonymization*: in order to exchange information between network operators it is essential to anonymize Netflow-traces and related information in a way that *(i)* no sensitive information on the end user or its habits can be won and *(ii)* the correlation of different events is still possible. *Entropy of Netflow data*: we believe it is worth studying the entropy of Netflow data in contrast to raw packet data's entropy in order to gain further insight on what can be detected using network flow data and what cannot. *Netflow-based detection metrics*: in order to detect botnet activity from Netflow data, reliable detection metrics are essential. Systematically studying this field of research should ultimately advance botnet detection research. *Detection algorithms*: for Netflow-based detection of botnet command and control traffic, the development and assessment of specially tailored detection algorithms is necessary to satisfy detection accuracy, efficiency and effectivity. *Combined Netflow-based botnet detection*: combining network flow data with system log files that can equally easy be collected by network operators as flow-traces (e.g. DNS queries, mail traffic, NTP queries) should increase detection accuracy while not affecting detection

6. Conclusion

In this paper, we discussed several approaches to botnet detection that currently exist and reflect current state of the art. Amongst these, we found only one purely Netflow-based approach. We believe, that Netflow-based approaches are promising in high-traffic environments as they contribute to efficiency and protect a user's privacy. We further believe, however, that in order to advance research in Netflow-based botnet detection, the compilation of a labeled reference data set is essential.

In order to compile such a reference data set, we propose a modular data collection environment that can easily be integrated in different network infrastructures and that has been used to collect a total of 34 GB suspicious Netflow and raw packet traces. Further, we highlighted possible future work and open questions that should be studied next and sketched a roadmap to the development of a Netflow-based detection appliance suitable for application in large network provider environments.

The data set collected using the above sketched platform as well as specific configuration details and management scripts will be made availably to researchers by mail to the authors.

7. Acknowledgments

This work has been partially supported by the German Federal Ministry of Education and Research under grant number 01BY1201F and by CASED.

8. References

Abt, S., (2009), "A statistical approach to flow-based network attack detection", Bachelor's thesis, Darmstadt University of Applied Sciences, Faculty of Computer Science, Darmstadt.

Adeel, M., Tokarchuk, L., Cuthbert, L., Feng, C.S., and Qin, Z.G., (2009), "A distributed framework for passive worm detection and throttling in p2p networks", in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference,* pp. 1-5.

Arbor Networks, (2005-2011), *Worldwide Infrastructure Security Report,* Arbor Networks, Inc., http://www.arbornetworks.com/report (last accessed: January 2012).

Baecher, P., Koetter, M., Holz, T., and Dornseif, M., (2006), "The nepenthes platform: An efficient approach to collect malware", in *Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006).*

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., and Karir, M., (2009), "A survey of botnet technology and defenses", in *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security,* pp. 299-304.

Binkley, J., and Singh, S., (2006), "An algorithm for anomaly-based botnet detection", in *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet*.

Cisco Systems, (2007), "Netflow Services Solutions Guide", http://www.cisco.com/en/US/docs/ios/solutions docs/netflow/nfwhite.html (last accessed: January 2012).

Claise, B. (Editor), (2008), "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, http://tools.ietf.org/html/rfc5101.

Coskun B., Dietrich, S., Memon, N., (2010), "Friends of an enemy: Identifying local members of peer-to-peer botnets using mutual contacts", in *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 131-140.

Dagon, D., Zou, C., and Lee, W., (2006), "Modeling botnet propagation using time zones", in *Proceedings of the 13 th Network and Distributed System Security Symposium*.

Feily, M., Shahrestani, A., and Ramadass, S., (2009), "A survey of botnet and botnet detection", in *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pp. 268–273.

Freiling, F., Holz, T. and Wicherski, G., (2005), "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks" in *Proceedings of 10 th European Symposium on Research in Computer Security,* pp. 319-335.

Goebel, J., and Holz, T., (2007), "Rishi: Identify bot contaminated hosts by irc nickname evaluation", in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*.

Gorecki, C., (2007), "Trumanbox – improving malware analysis by simulating the internet", Diploma thesis, RWTH Aachen University, Department of Computer Science, Aachen.

Govil, J., (2007), "Examining the criminology of bot zoo", in *Proceedings of the 6th International Conference on Information, Communications Signal Processing*, pp. 1-6.

Gu, G., Porras, P., Yegneswaran, V., Fong, M., and Lee, W., (2007), "Bothunter: Detecting malware infection through ids-driven dialog correlation", in *Proceedings of the 16th USENIX Security Symposium,* pp. 167-182.

Gu, G., Zhang, J., and Lee, W., (2008a), "Botsniffer: Detecting botnet command and control channels in network traffic", in *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*.

Gu, G., Perdisci, R., Zhang, J., and Lee, W., (2008b), "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection", *Usenix Security Symposium*.

Karasaridis, A., Rexroad, B., and Hoeflin, D., (2007), "Wide-scale botnet detection and characterization", in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*.

Li, C., Jiang, W., and Zou, X., (2009), "Botnet: Survey and case study", in *Proceedings of the International Conference on Innovative Computing, Information and Control,* pp. 1184-1187.

Liu, J., Xiao, Y., Ghaboosi, K., Deng, H., and Zhang, J., (2009), "Botnet: Classification, attacks, detection, tracing, and preventive measures", in *EURASIP Journal on Wireless Communications and Networking,* Vol. 2009.

Livadas, C., Walsh, R., Lapsley, D., and Strayer, W., (2006), "Using machine learning techniques to identify botnet traffic", in *Proceedings of the 2nd IEEE LCN Workshop on Network Security (WoNS'2006)*.

Mirkovic, J. and Reiher, P., (2004), "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Comunications Review*.

Patterson, D., (2002), "A simple way to estimate the cost of downtime," in *Proceedings of the 16th USENIX conference on System administration,* pp. 185-188.

Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N., (2007), "The ghost in the browser analysis of web-based malware", in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets,* pp. 4-4.

Racine, S., (2004), "Analysis of internet relay chat usage by ddos zombies", Master's thesis, Swiss Federal Institute of Technology, Zurich.

Rajab, M., Zarfoss, J., Monrose, F., and Terzis, A., (2006), "A multifaceted approach to understanding the botnet phenomenon" in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*.

Ramachandran, A., Feamster, N., and Dagon, D., (2006), "Revealing botnet membership using dnsbl counter-intelligence", in *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*.

Scarfone, K., and Mell, P., (2007), "Guide *to Intrusion Detection and Prevention Systems (IDPS)", Recommendation of the National Institute of Standards and Technology,* Special Publication 800-94.

Strayer, W., Walsh, R., Livadas, C., and Lapsley, D., (2006), "Detecting botnets with tight command and control", in *Proceedings of the 31st IEEE Conference on Local Computer Networks*.

The Honeynet Project, (2005), “Know your enemy: Tracking botnets”, http: //www.honeynet.org/ (last accessed: January 2012).

Thing, V., Sloman, M., Dulay, N., Venter, H., Eloff, M., Labuschagne, L., Eloff, J., and VonSolms, R., (2007), “A survey of bots used for distributed denial of service attacks,” *IFIP International Federation for Information Processing,* vol. 232, pp. 229.