Müller, Neumair, Reiser, Dreo Rodosek (Hrsg).: 10. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 11

A new Attack Composition for Network Security

Frank Beer[1], Tim Hofer[1], David Karimi[1], and Ulrich Bühler[1]

Abstract: As the current cyber threat landscape is becoming more depressing, sophisticated intrusion detection systems must evolve to protect network infrastructures efficiently. Building such a detector is highly data-driven and requires quality datasets to evaluate different phases in both the development and deployment process. However, finding publicly available captures with a ground truth is challenging. Most existing datasets focus on very specific subjects such as botnet, flooding, or brute-force traffic rather than providing a broad arsenal of different attack vectors threatening today's networks. This work addresses this gap by introducing a new attack composition comprising a multitude of classic as well as state-of-the-art attacks. The dataset embrace rich and untreated packet traces including payload, collected log events, and a detailed ground truth. Intitial results reveal the proposed captures complement existing traces and provide a sound base for various mining applications in the field of network security research.

Keywords: Attack dataset, data sharing, ground truth, intrusion detection, network security

1 Introduction

The advances of today's cyber attacks against network infrastructures are versatile and alarming. Hence, there is a huge demand for trustworthy remedies. Research in this direction frequently utilize supervised machine learning (see [BG15]) to build sophisticated network intrusion detection systems (IDSs). The downside of these approaches is the inherent necessity of quality datasets for training and validation purposes. Ideally, such a dataset should represent realistic traffic and cover a multitude of classic as well as state-of-the-art attack types providing legitimate traffic from the underlying network of interest. It further should exhibit a ground truth (GT) annotating traffic with class labels to indicate malicious and benign instances. Moreover, the captures should embrace a fine-grained data format, which offers additional opportunities such as mining meaningful features to increase detection capabilities or to benchmark competing solutions that rely on different inputs including packet or flow data.

Finding a publicly available dataset fulfilling these requirments is a challenging task and a true concern in the network security community with respect to reproducibility and comparability [AB14]. However, several attempts have been made over the last decades: The most prominent datasets are the DARPA 98/99 traces [Li00a, Li00b, Ha01] and derived versions such as KDD-Cup 99[2], GureKDDcup [Pe08], and NSL-KDD [Ta09]. Despite their age, these are yet frequently utilized in the community although widely criticized due to design flaws and their inability to meet contemporary requirements with respect

[1] Network and Data Security Group (NDSec), University of Applied Sciences Fulda, Leipziger Strasse 123, D-36037 Fulda, {frank.beer, tim.hofer, david.karimi, u.buehler}@informatik.hs-fulda.de

[2] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

to traffic and state-of-the-art attack variants (e.g. [Mc00, MC03, Ta09]). Other more recent traces provide quality data, but pursue specific goals. L-Flows [Sp09] and SSH-DS [Ho14] mainly focus on brute-force attacks supplying highly aggregated flow data. Other novel captures such as CTU-13 [Ga14] and Booters-15 [Sa15] either concentrate on malware or denial-of-service (DoS) attacks. Considering multi-purpose intrusion detection, the only suitable dataset comprising a broader attack arsenal is the ISCX-2012 dataset [Sh12]. However, ISCX-2012 merely covers few malicious traffic compared to its benign counterpart, which makes its application for both training and testing a detector unfavorable particularly when working on flow level. Assembled datasets are another source of interest by fusing several independent traces. Two of these collections are ISOT [Sh11] and ISCX-Botnet [Be14] concentrating on botnet detection. ISOT incorporates real traces from LBNL/ICSI[3] and the Traffic Lab at Ericsson Reasearch [Sz08] with malware captures from the Honeynet project[4]. On the other side, ISCX-Botnet merges partial traffic from ISOT, ISCX-2012, and CTU-13 to compile a representative dataset. Despite the huge effort made over the years, this indicates that no general network intrusion dataset exists to date, which is in line with the opinion of other authors (e.g. [Ce11, Ga14, MBM15]).

Motivated by these findings and discussions among the community to share resulting traces, this paper introduces a new dataset, which already substantiated promising results in [BB17]. It is based on the following perceptions: According to the given requirements, both benign and illicit network traffic are essential developing and validating IDSs. However the legitimate part highly depends on the underlying infrastructure including influencing factors like software configurations and human interaction, which should be collected at the target domain particularly when considering anomaly-based intrusion detection. In this respect, we argue that malicious traffic is of high interest, because most existing captures focus on isolated attack types such as botnet, brute-force, or flooding, which is very specific for a general assessment of an IDS. Therefore, we build a dataset primarily concentrating on attack data rather than legitimate traffic using state-of-the-art penetration testing suites, malware instances collected "in the wild", recently reported exploits, and classic tools. As this arsenal is very basic equipment for cyber criminals, we carefully incorporated it into well-defined scenarios reflecting realistic attack situations, which are applicable to most conventional network infrastructures. Furthermore, our dataset embrace untreated packet traces including payload and captured log events documented by a rich GT. Thus, it can be reused to salt legitimate traffic based on common strategies such as the overlay methodology (see [AH11]) supporting both the development and deployment process of IDSs.

The remainder is structured as follows: First, we introduce the proposed dataset in Section 2 by illustrating the underlying network infrastructure (Section 2.1), attack scenarios (Section 2.2) and a summary of evolved attack types (Section 2.3). Section 3 outlines obtained results comparing the dataset against other related captures (Section 3.1). Moreover, a qualitative analysis is provided applying our traces to a well-known IDS (Section 3.2). In Section 4, we conclude and frame future work.

[3] `http://www.icir.org/enterprise-tracing/`
[4] `http://www.honeynet.org/chapters/france/`

2 NDSec-1 Dataset

Based on the discussed absence of appropriate traces providing a broad range of different attacks, this section proposes a new dataset. In contrast to most other solutions, it contains very few background traffic, and thus serves as attack repository. Additionally, we attached importance to other practical aspects. The following principles were key to the design of the dataset, which we refer to as NDSec-1:

- Support of various attack types and variants
- Attacks wrapped around realistic scenarios
- Simple infrastructure to incorporate other traces
- Detailed GT based on bidirectional flow semantics
- Provide raw packet captures including log events

In what follows, we highlight network infrastructure (Section 2.1) and describe involved attack scenarios (Section 2.2). Section 2.3 summarizes the resulting attack distribution.

2.1 Network Infrastructure

To build up an appropriate infrastructure, we followed a conventional topology placed on a testbed[5] located at our campus network. It operated as hypervisor mimicking two subnets, i.e. a private network (company or organization) and the simulated Internet. Both subnets were seperated by an OpenWRT[6] router acting as NAT gateway with firewall capabilities for the private network. Only port 80 was open for ingress traffic to the internal web server. Additionally, we configured the router to forward real Internet requests to the campus network, while the simulated Internet contained prepared virtualized machines to serve as controlled infrastructure to most performed scenarios (e.g. email system, exploit kit, or bot master). The private side relied on a heterogenous set of workstations and machines based on both different Windows and Linux versions. Traffic was captured by a tcpdump[7] sensor inside the private network. Thus, only incoming and outgoing connections of that subnet were observed. The log event information (i.e. syslog and Windows event log) were collected locally at each host and extracted after each scenario completed. An overview of this simplified network infrastructure is depicted in Figure 1.

2.2 Attack Scenarios

Bring Your Own Device: The bring your own device (BYOD) phenomenon is a pragmatic mindset established by enterprises and organizations enabling employees and partners to

[5] Hypervisor: VMWare ESXi 6.0; Memory: 100 GByte; CPU: 2x 2.30 GHz Xeon (E5-2630); HDD: 4x 1 TByte (RAID 5); NIC: 2x Intel I350 (1 GBit)

[6] `https://openwrt.org/`

[7] `http://www.tcpdump.org/`

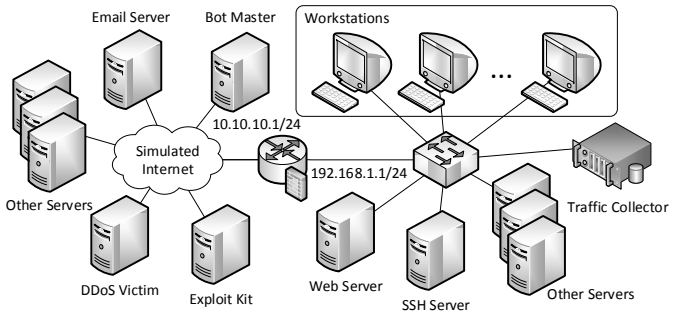

Fig. 1: Simplified topology of our virtualized network landscape: the simulated Internet (left) and the protected private network (right).

utilize personal hardware in-house. Despite all of its advantages, new security risks arises which may permit an attacker to act from the inside. To build such a scenario, a machine was placed to the protected network environment serving as a compromised BYOD. We had full access to the machine via an installed backdoor provided by Metasploit[8] using a binary Linux trojan. In order to study the unknown network, several reconnaissance activities were performed against the infrastructure and potential victims could be identified, i.e. an internal SSH server and a client system connected to both email and web server. We attacked the former by a dictionary brute-force uncovering login credentials. Client and email server were targeted combining ARP and DNS spoofing techniques to get between these machines. Thus, we could pretend to be the legitimate server forcing the client to disclose private information, which was exploited to steal valuable content from the victim's account. Another variant was pursued to get between host and web server utilizing ARP poisoning and an SSL proxy (see SSLsplit[9]). Hence, we could successfully hijack login credentials despite encrypted communication. To exfiltrate all obtained assets, we uploaded related data to an external FTP server using the compromised BYOD.

Watering Hole: It is common for enterprises or organizations to self-host services from within their infrastructure. In this scenario, an external intruder tried to compromise an internally hosted web server with the intention to infect a related group of hosts, i.e. a watering hole attack. First, a brute-force attack was performed against the front end of the web server followed by an SQL injection retrieving several logins and password hashes from the back-end database. Based on that gathered knowledge, we injected cross site scripts (XSS) to private pages of users found in the database. Hence, a small group of users could be targeted. The effect of XSS was a malicious redirect to an external exploit kit (i.e. Crimepack 3.1.3) that seeked for unpatched web browsers and vulnerable plugins on visiting hosts to inject specific malware. In our case, we used an unreported Internet Explorer exploit and ToxiCola as ransomware. Note, each instance of this specific malware contained customized binaries generated by the malware author right before its deployment complicating detection. Using this scenario, two hosts inside the protected network were

[8] `http://www.metasploit.com/`
[9] `https://www.roe.ch/SSLsplit`

successfully infected. ToxiCola encrypted several important local documents and reported back to a known server in the Internet.

Botnet: The rental of botnets operated by cyber crews is a lucrative business in the underground economy. Hence, these illicit infrastructures increasingly gain popularity. This trend is crucial for enterprises and organizations, because essentially any host of a legitimate network may serve as a bot, and thus has potentials to be part of a criminal act once infected. Citadel 1.3.5.1 as revised version of the well-known Zeus botnet was employed in this scenario. Based on a normal operating network, we infected three legitimate hosts with Citadel binaries. This task could be performed through conventional email spam using the recent vulnerabilities CVE-2015-2509 (Windows Media Center), CVE-2015-5122 (Flash Player), and a rogue download caused by XSS placed on a website in the simulated Internet. After the infection, all three bots communicated via HTTP to a prepared bot master. Among several traffic footprints between master and bots, we instructed all bots to download new commands. These contained hostile payload to perform a distributed DoS (DDoS) via SYN flooding to a single destination outside the network. Beside this successful attack, two of the bots stole local configuration files and transfered them to an external FTP server.

Attacks Without Specific Context: In this experiment, all attacks from the previous three scenarios were repeated without specific context. Additionally, we performed a number of other attacks. For instance, we used the tool Yersinia[10] to run DHCP starvation attacks, which exhausted the number of available IP addresses from a known DHCP server utilizing spoofed MAC addresses. HTTP floods were carried out using the Apache HTTP server benchmarking tool[11]. Additionally, we seeked for vulnerabilities with Nikto[12] and attacked an FTP service by the well-known THC Hydra tool[13]. Tsunami[14] was employed to perform DNS amplification attacks resulting in a DNS flooding attempt. Finally, we made use of the classic hping3[15] to send a high amount of UDP packets to specific target hosts in the network.

2.3 Dataset Summary

As a result of the processed scenarios inside the simplified testbed, numerous attack types and variants could be covered within NDSec-1, which can be aggregated to 12 categories. A summary of all instances along with the captured packets and byte distribution is illustrated in Table 1. Note, most of the rogue actions were performed manually. Particularly, the execution of the defined scenarios was carefully crafted to evade detection as much as possible. Thus, we believe the resulting dataset reflects realistic footprints. Each involved activity was labeled according to attack category using network flows. YAF[16] as

[10] `http://www.yersinia.net/`
[11] `http://httpd.apache.org/docs/2.4/programs/ab.html`
[12] `https://cirt.net/Nikto2/`
[13] `http://sectools.org/tool/hydra/`
[14] `https://www.infosec-ninjas.com/tsunami/`
[15] `http://www.hping.org/hping3.html`
[16] `https://tools.netsa.cert.org/yaf/`, version 2.8.4

sophisticated flow exporter was chosen for this reason using default timeout settings and bidirectional flow semantics. In order to share the outcome of this work, all raw packet traces, log files, and GT were published on our website[17]. Again, we would like to emphasize that NDSec-1 was designed to provide pure attack sequences and can be reused to season other legitimate network traces as suggested in [Ce11]. Since the captures are fine-grained, they may support the evaluation of existing or new detection approaches based on packet, flow, or log data.

Attacks	Packets	Bytes per packet
Botnet (Citadel)	5198	707.2697
HTTP brute-force	26093	495.1155
FTP brute-force	1530	63.0137
SSH brute-force	20873	179.0432
HTTP flooding	167238	115.6783
SYN flooding	890895	100.5449
UDP flooding	2275614	137.4647
Malware/exploits	8802	866.7635
Probe	21707	329.1476
Spoofing	1199	60.0083
SSL proxy	11602	776.8269
XSS/SQL injection	334	278.5419

Tab. 1: NDSec-1 attack and packet distribution

3 Results and Discussion

3.1 Comparative Study

This section evaluates six predominant and most related traces found in recent network security literature, i.e. Booters-15, CTU-13, ISCX-2012, ISCX-Botnet, L-Flows, and SSH-DS. We discuss the main characteristics and provide a comparison to NDSec-1.

The available format of a dataset is a key aspect for its applicability. While the majority of captures (i.e. ISCX-2012, ISCX-Botnet, Booters-15, CTU-13, and NDSec-1) provide rich packet traces (PCAP format) allowing detailed analysis on packet header or payload level, L-Flows and SSH-DS supply highly aggregated flow data. Based on this distinction, the latter sources can only be used for the emerging field of flow-based intrusion detection (e.g. [Sp10, Ho14, BB17]). In this context, the applicability is also determined by the involved attack categories, which either permit to assess the performance of IDSs towards a broad arsenal or to very specific attacks. ISCX-2012 and NDSec-1 include the largest attack repertory, while others target on certain instances (e.g. rogue botnet activities, brute-force attempts, or network stresser services). Moreover, the underlying environment is an essential property. Several of the examined datasets were captured “in the

[17] `http://www2.hs-fulda.de/NDSec/NDSec-1/`

wild" or under equivalent conditions providing most realistic traffic (i.e. L-Flows, SSH-DS, Booters-15, and CTU-13). However, traces with this characteristic usually embrace sensitive information raising privacy concerns. Therefore, some of these evaluated traces have been provided on flow level, were anonymized or sanitized such that certain information in these datasets are missed or become ineligible (e.g. anonymized IP addresses for SSH-DS or chopped off payload for CTU-13 on benign data). Note, we refer to raw data if the observed captures are not postprocessed. An alternative to circumvent privacy issues are synthetic datasets. These are recorded in a controlled environments (physical or virtual infrastructure), which does not necessarily mean they are inappropriate or less qualified to train or validate intrusion detection techniques. In fact, this type of datasets is gaining more attention in literature (e.g. [BWM08, Ce11, Be14]) and can produce realistic traffic footprints once the environment is setup properly. Traces comprising this property are ISCX-2012, ISCX-Botnet, and NDSec-1. The last characteristics deal with the underlying GT, which is another critical point for existing datasets [AB14]. Captures including malicious and legitimate traffic generally require a GT to apply supervised learning tasks, that may exist on different levels (e.g. per IP (ISCX-Botnet) or flow (ISCX-2012, L-Flows, CTU-13, and NDSec-1)). Simple annotations (i.e. the distinction between normal and hostile traffic) suffice binary classification problems, but detailed assessments of traces can only be achieved using rich labels. The latter is covered by L-Flows and NDSec-1 only. Note, SSH-DS and Booters-15 do not provide such a GT at all, because all involved data refer to malicious traffic. Table 2 depicts the discussed characteristics of all considered datasets.

Dataset	**Available format**			**Raw**	**Syn-**	**Involved attacks**								**GT**		
	PCAP	**Flow**	**Log**	**data**	**thetic**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**IP**	**Flow**	**Rich**
Booters-15 [Sa15]	✓	–	–	(✓)	–	–	–	–	✓	–	–	–	–	–	–	–
CTU-13 [Ga14]	✓	–	–	(✓)	–	✓	–	–	✓	✓	–	–	✓	–	✓	–
ISCX-2012 [Sh12]	✓	–	–	✓	✓	(✓)	✓	✓	✓	✓	–	✓	–	–	✓	–
ISCX-Botnet [Be14]	✓	–	–	(✓)	✓	✓	–	–	✓	✓	–	–	✓	✓	–	–
L-Flows [Sp09]	–	✓	✓	–	–	–	✓	–	–	✓	–	✓	–	–	✓	✓
NDSec-1	✓	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓
SSH-DS [Ho14]	–	✓	✓	–	–	–	✓	–	–	–	–	–	–	–	–	–

✓=characteristic included, (✓)=characteristic partially included, −=characteristic not included; 1=botnet (command-and-control, fraud, fast flux, etc.), 2=brute-force, 3=other malware/exploit, 4=flooding, 5=probe, 6=spoofing, 7=web attack (SQL injection, XSS, etc.), 8=others (spam, SSL proxy, etc.)

Tab. 2: Comparison of most related intrusion detection datasets

These observations infer most datasets were captured for specific goals. L-Flows and SSH-DS comprise flow data and log event information, which can be correlated providing further insights to potential attack situations. However, the limiting factor on both captures is their format disallowing detailed analysis below flow level such as payload inspection. On the other hand, Booters-15, ISCX-Botnet, and CTU-13 provide rich traces. They concentrate either on botnet or DDoS-as-a-Service traffic ignoring other sophisticated attack vectors including brute-force or web attacks. These findings are crucial particularly when working towards a general IDS covering various attack types. The only two datasets comprising a wider range are ISCX-2012 and NDSec-1. Yet, ISCX-2012 neither covers log events nor a detailed ground truth, which refuse elaborate examinations per attack. Moreover, it does not cover frequently used spoofing attempts, man-in-the-middle attacks, or a real botnet as opposed to NDSec-1.

3.2 Insights to NDSec-1 Using Snort

As opposed to machine learning techniques which we partly examined in [BB17], this section briefly reports about the qualitative results running NDSec-1 against a signature-based detection engine. Snort[18] as state-of-the-art IDS was chosen for this reason employing default system settings, latest community rules, and emerging threats (ET) extension[19]. Note, all alerts per scenario (see Section 2.2) were mapped to the corresponding flow-based GT utilizing timestamps, IP addresses, and ports to diagnose matches.

Starting with the BYOD scenario, most of the probing activities remained undetected particularly for the interesting port range 0 to 1023. However, Snort discovered some vertical scans for specific ranges, i.e. 5800 to 5820 and 5900 to 5920. Additionally, some signatures caused alarms for database ports providing meaningful messages. The SSH dictionary attack was alerted on a periodical basis including the SSL proxy, while backdoor communication using Metasploit and the ARP spoofing attempts were not exposed by Snort. Considering normal traffic, some minor false alarms took place especially on the alert "PROTOCOL-DNS TMG Firewall Client long host entry exploit attempt", which occured in 7% of all benign cases. Since such a security gateway was not installed in our environment, this message was misleading for all involved scenarios. Within the watering hole scenario, a high amount of traffic was caused by the HTTP brute-force in order to take over the involved web server. No signature triggered on this activity. The same took place for the XSS placement and traffic produced by the injected ransomware. On the other side, the SQL injections as well as traffic induced by Crimepack could be detected. Eight different ET rules applied on the former such that 40% of the injection attempts were uncovered. Traffic caused by the latter was unmasked completely by an ET alarm designed for the Eleonore exploit kit. On the botnet scenario, exploited vulnerabilities (i.e. Windows Media Center and Flash Player) and command-and-control traffic could be identified with explicit ET signatures. However, involved DDoS and data theft remained concealed. Clearly, the exfiltration based on a legitimate FTP upload activity within a rogue context, which is difficult to expose using signature-based engines. Applying the last trace revealed similar results for attacks intersecting with the previous scenarios. Yet, flooding attacks based on HTTP and UDP basically remained undetected (hit rate $< 1\%$), while FTP brute-force and vulnerability scans could be identified in 33% and 63% of the cases.

This confirms several attack instances inside NDSec-1 could be safely uncovered by Snort using a default setup. Yet, some basic attacks were missed or hit only partially. Particularly, the latter attacks comprised a high traffic volume compared to other sure detections (see Table 1). Taking this factor into account, the overall classification on flow level revealed poor results in terms of conventional metrics such as hit rate or accuracy. Being aware that a more sophisticated configuration including enterprise ruleset certainly would boost Snort's accuracy, yet the obtained results using NDSec-1 in practice looked very promising. This indicates that incorporating penetration testing suites, recent malware instances and classic attack tools within realistic scenarios provide a sound base to support the development cycles of a new detector or to reveal weaknesses of deployed IDSs.

[18] `https://snort.org/`, version 2.9.9
[19] `https://rules.emergingthreats.net/`, version 8499

4 Conclusion and Future Work

Labeled data are essential for network security research in order to assess existing or new intrusion detection techniques. Most existing datasets are limited to certain attack types such as botnet, brute-force, or flooding. This fact is crucial particularly when examining intrusion detection systems towards a multitude of different attacks. Therefore quality traces are required comprising both classic as well as recent attack vectors. In order to mitigate these findings, this work proposed a new dataset concentrating on realistic attack scenarios containing a broad arsenal of attacks based on penetration testing tools, recent malware, and exploits. A comparative study revealed, it can compete with related work in terms of rich captures and ground truth, but it is superior considering the quantity of attack variants. Furthermore, the evaluation against the latest Snort version demonstrated several attacks remained undetected or were hit only partially, which manifests the merit of our effort in addition. In this respect, the captures can be deemed complementary to existing traces. Based on these results, future work attends to assess further detectors and to evaluate potential limitations using the dataset. Besides these activities, we plan to expand the traces with more sophisticated attacks focusing on current and emerging threats.

References

[AB14] Abt, S.; Baier, H.: Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research. In: 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. pp. 40–55, 2014.

[AH11] Aviv, A.J.; Haeberlen, A.: Challenges in Experimenting with Botnet Detection Systems. In: 4th Workshop on Cyber Security Experimentation and Test. 2011.

[BB17] Beer, F.; Bühler, U.: Feature Selection for Flow-based Intrusion Detection Using Rough Set Theory. In: 14th IEEE International Conference on Networking, Sensing and Control. 2017.

[Be14] Beigi, E.B.; Jazi, H.H.; Stakhanova, N.; Ghorbani, A.A.: Towards Effective Feature Selection in Machine Learning-based Botnet Detection Approaches. In: IEEE Conference on Communications and Network Security. pp. 247–255, 2014.

[BG15] Buczak, A.L.; Guven, E.: A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18(2):1153–1176, 2015.

[BWM08] Brauckhoff, D.; Wagner, A.; May, M.: FLAME: A Flow-Level Anomaly Modeling Engine. In: 2nd Workshop on Cyber Security Experimentation and Test. 2008.

[Ce11] Celik, Z.B.; Raghuram, J.; Kesidis, G.; Miller, D.J.: Salting Public Traces with Attack Traffic to Test Flow Classifiers. In: 4th Workshop on Cyber Security Experimentation and Test. 2011.

[Ga14] García, S.; Grill, M.; Stiborek, H.; Zunino, A.: An Empirical Comparison of Botnet Detection Methods. Computers and Security, 45:100–123, 2014.

[Ha01] Haines, J.; Lippmann, R.; Fried, D.; Zissman, M.; Tran, E.; Boswell, S.: 1999 DARPA Intrusion Detection Evaluation: Design and Procedures. Technical report, MIT Lincoln Laboratory, 2001.

[Ho14] Hofstede, R.; Hendriks, L.; Sperotto, A.; Pras, A.: SSH Compromise Detection Using NetFlow/IPFIX. ACM SIGCOMM Computer Communication Review, 44(5):20–26, 2014.

[Li00a] Lippmann, R.; Fried, D. J.; Graf, I.; Haines, J. W.; Kendall, K. R.; McClung, D.; Weber, D.; Webster, S. E.; Wyschogrod, D.; Cunningham, R. K.; Zissman, M. A.: Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. In: DARPA Information Survivability Conference and Exposition. volume 2, pp. 12–26, 2000.

[Li00b] Lippmann, R.; Haines, J.; Fried, D.; Korba, J.; Das, K.: The 1999 DARPA Off-line Intrusion Detection Evaluation. Computer Networks, 34(4):579–595, 2000.

[MBM15] Małowidzki, M.; Berezinski, P.; Mazur, M.: Network Intrusion Detection: Half a Kingdom for a Good Dataset. In: NATO STO SAS-139 Workshop. 2015.

[Mc00] McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations As Performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4):262–294, 2000.

[MC03] Mahoney, M. V.; Chan, P. K.: An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. In: 6th Symposium on Recent Advances in Intrusion Detection. pp. 220–237, 2003.

[Pe08] Perona, I.; Gurrutxaga, I.; Arbelaitz, O.; Martin, J. I.; Muguerza, J.; Pérez, J. M.: Service-independent payload analysis to improve intrusion detection in network traffic. In: 7th Australasian Data Mining Conference. pp. 171–178, 2008.

[Sa15] Santanna, J.J.; van Rijswijk-Deij, R.; Sperotto, A.; Hofstede, R.; Wierbosch, M.; Granville, L. Zambenedetti; Pras, A.: Booters - An analysis of DDoS-as-a-Service Attacks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management. pp. 243–251, 2015.

[Sh11] Sherif, S.; Issa, T.; Ali, G.; Bassam, S.; David, Z.; Wei, L.; John, F.; Payman, H.: Detecting P2P Botnets through Network Behavior Analysis and Machine Learning. In: Ninth Annual International Conference on Privacy, Security and Trust. pp. 174–180, 2011.

[Sh12] Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A. A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers and Security, 31(3):357–374, 2012.

[Sp09] Sperotto, A.; Sadre, R.; van Vliet, F.; Pras, A.: A Labeled Data Set For Flow-based Intrusion Detection. In: 9th IEEE International Workshop on IP Operations and Management. pp. 39–50, 2009.

[Sp10] Sperotto, A.; Schaffrath, G.; Sadre, R.; Morariu, C.; Pras, A.; Stiller, B.: An overview of IP flow-based intrusion detection. IEEE Communications Surveys & Tutorials, 12(3):343–356, 2010.

[Sz08] Szabó, G.; Orincsay, D.; Malomsoky, S.; Szabó, I.: On the Validation of Traffic Classification Algorithms. In: International Conference on Passive and Active Network Measurement. pp. 72–81, 2008.

[Ta09] Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: IEEE Symposium on Computational Intelligence in Security and Defense Applications. pp. 53–58, 2009.