Generating Benchmark Datasets for Intrusion Detection Systems

João Santos[1]
2140141@my.ipleiria.pt

Catarina Silva[1,2]
catarina@ipleiria.pt

Mário Antunes[1,3]
mario.antunes@ipleiria.pt

[1]School of Technology and Management, Polytechnic Institute of Leiria, Portugal

[2]Center for Informatics and Systems of the University of Coimbra, Portugal

[3]Center for Research in Advanced Computing Systems, INESC-TEC, University of Porto, Portugal

Abstract

Intrusion Detection Systems (IDS) are applications used to detect anomalous activities in computer networks and their computers. An emergent challenge faced by the research community is to automate the construction of real-world based datasets with which new detection algorithms can be exploited and benchmarked. The existing solutions to generate datasets are usually based on synthetic network attacks and thus are not representative of real network activity. In this paper, we propose a generic architecture to generate network traffic datasets, including attacks, based on a representation protocol. We also propose a deployment strategy to automate the network traffic datasets construction based on a statistical model of normal behaviour, which allows the generation of diverse and analogous to real world computer networks behaviours.

1 Introduction

In network security, an *intrusion* is a set of actions carried on by an *intruder* (internal or external) that attempts to compromise the network infrastructure, through a violation of its security foundations, namely confidentiality, integrity and availability. In general, *intrusion detection* is the process devoted to monitor the network (or systems) events and to trigger an alert to those that may be the target of an ongoing attack.

Intrusion Detection Systems (IDS) are applications usually focused on incidents where events usually occur at very high rates and in an obscured way. Signs of network intrusion can be found by analysing network packets and their associated information flows. Therefore, the main goal of an IDS is to analyse, in real time, network packets in transit, to positively identify all occurrences of actual attacks, and, at the same time, not be mistaken by regular events or be distracted by the signalling of falsely identified attacks.

To accomplish this goal, IDS monitor network traffic in real time and analyse user and system activities, further auditing the faults and vulnerabilities to which the system is exposed. These systems also recognise an activity model and statistically model abnormal behaviour.

Traditionally, IDS deployment has been classified into two distinct detection methods [1]: (i) *signature-based*, as these systems are based on the description of known attacks by the means of a "signature" or a pattern of known and previously seen attacks; and (ii) *behaviour-based*, in which the system builds a model of normal behaviour for the network and then looks for anomalous activities, that is those that do not match the previously established profile.

Snort (`www.snort.org`) is the most widely used non-commercial open source signature based IDS, being an anomaly based IDS confined mainly to the research community with little expression in production systems [2].

New IDS deployment research needs to be tested against real world network traffic. However, as intruders usually leave slim or no traces of their activity, and real network packets can carry highly sensitive information, it becomes very difficult to have representative data of real attacks, collected from real data networks. To overcome this limitation, artificially created datasets with "safe" and "sanitised" data have been made available to the research community, as DARPA 1999 KDD Cup Challenge dataset [3] and Massicote *et al.* work [4] are two popular and widely used examples.

In this paper, we propose a generic architecture to generate real-world based datasets of network traffic, with both normal and abnormal network traffic flows, that could be used to test, deploy and benchmark intrusion detection systems. The datasets are built upon a statistical model that shapes the distribution of real raw packets flows through time. The dataset can be in raw format or *text based*, depending on the user needs.

The adopted strategy tries to overcome some limitations of the existing artificial datasets generators and to expedite the tests of new detection algorithms, using real world based network traffic.

2 Generating Datasets

In this section we describe the proposed architecture to generate datasets. We detail the overall architecture, the statistical analysis module, the dataset generation process and the representation protocol used for the resulting datasets.

2.1 Proposed architecture

Figure 1 depicts the main blocks of the general architecture we have designed to generate benchmark datasets for IDS testing.

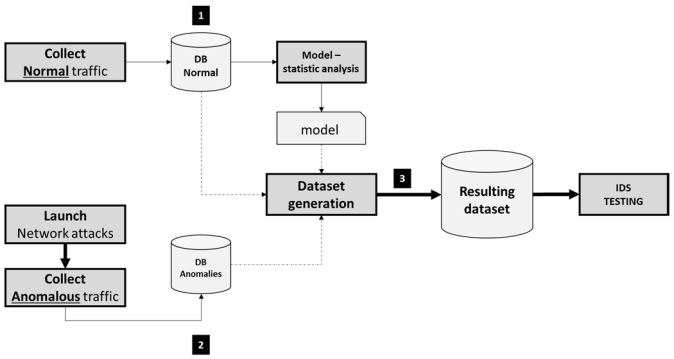


Figure 1: General architecture for data generation

The procedure starts with a normal network traffic collection in a real production network where the IDS solution is going to be implemented. The network should have diversity of networking protocols traffic and heterogeneity of network devices. Privacy issues must be framed into the company's information security policy.

Regarding the collection of packets related with network attacks, a pre-defined set of attacks is then launched against a test network composed by, at least, the computers/applications corresponding to the attacker and the victim. In our tests we have launched the attacks with Metasploit Framework (`www.metasploit.com`), which has a wide range of procedures to exploit already known vulnerabilities.

The collection of packets related with normal and anomalous activities is raw, in the *packet capture* (PCAP) format, and was obtained through the use of specific applications tools, like wireshark (`www.wireshark.org`) and tcpdump (`www.tcpdump.org`).

The statistics analysis block (detailed in Section 2.2) produces a profile of the network regarding the amount of network flows and packets processed. The dataset generation block is then fed with the datasets obtained during the network traffic captures, namely normal and anomalous datasets, together with the normal traffic model obtained. According to the input parameters used to model the dataset creation process, namely the frequency of packets flows and its duration, the system will generate the resulting dataset that interleaves the attacks in the normal traffic in a timely order. The adopted strategy allows the user to freely generate distinct datasets according to the input parameters that shapes the behaviour of the artificial network flows.

2.2 Deployment strategy

The first step after capturing the traffic is the analysis and processing of the packets, that will be the base of construction of the profiles related to normal traffic. Such analysis has two phases: (i) aggregation of the flows and (ii) constructing the profiles (see Figure 2). In the first phase flow information is stored in a hash array with source IP; destination IP; protocol; packets.

The statistics analysis block (see Figure 1) analyses the normal network traffic and generates the corresponding profile, namely the amount of packets collected and network flows characterization. Such profile consists of fitting a normal curve.

Figure 2 depicts the statistics module processing. It starts by opening the input file with the network traffic and then processing the corresponding packets. Network traffic profiles are generated according to the network flows, protocols observed and the corresponding amount of packets collected.

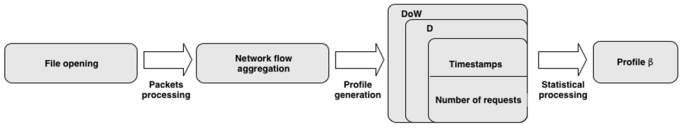
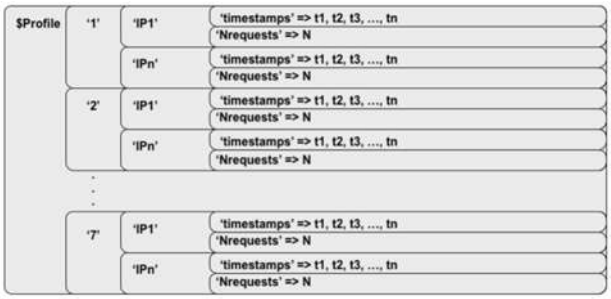


Figure 2 – statistics module processing

a)



b)

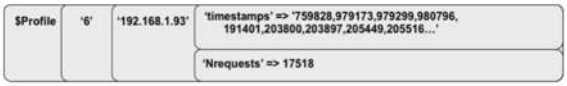


Figure 3: Format used to store network activity profiles

Finally, all users (D) with the same day (DoW) are aggregated in a single cumulative distribution function, and with the average of requests the inverse transform of the cumulative distribution function is obtained taking into account the corresponding traffic distribution and using a Weibull distribution as in [5]. The set of these data constitutes the generated profile. The format used to store the profiles is depicted in Figure 3a) and an example is shown in Figure 3b).

Figure 4a) illustrates an example of a *per flow* analysis, identifying the number of flows for each protocol. In Figure 4b) the analysis is made in a *per packet* basis, as it shows the amount of packets processed for each TCP/IP application.

Protocol	No_of_flows	Total
IGMP	2	
ICMP	2	
TCP	963	
UDP	662	
		1629

a)

Port	No_of_packets	Total
53	567	
67	21	
68	32	
80	8771	
123	14	
137	140	
138	62	
139	108	
443	101932	
445	12	
1073	81	
1900	211	
4070	314	
5222	15011	
5223	17	
5353	174	
11793	6	
17500	478	
31445	15	
41800	45	
		233250

b)

Figure 4: Statistics analysis of network traffic (a) per-packet analysis (a) and (b) per-applications analysis

2.3 Dataset generation

The overall dataset generation process, that is network capture, statistics analysis and dataset construction, is carried out through a Perl script. The general algorithm is illustrated in Figure 5, which present the flow of data and the main processing components carried on by the Perl script.

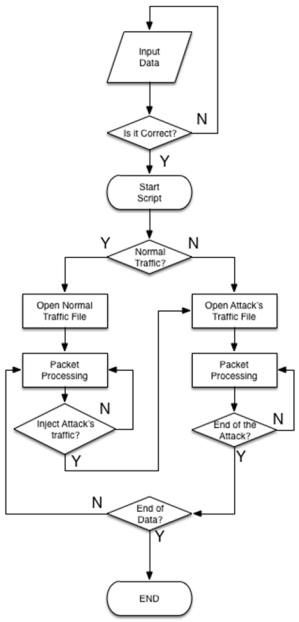


Figure 5: Overall algorithm for datasets construction

The resulting flows can be ready to be tested in real network environments, with the purpose of determining the capabilities of IDS in an organization or can be used as a research tool to construct benchmarks for new algorithm testing. In the later case, a text-based protocol is defined, as the user is able to dump to a text file the contents of each TCP/IP header fields, according to the input data needed to test a specific IDS. The fields chosen to integrate the dataset should be written in a stream of characters, without spaces and separated by a comma. In this preliminary stage, the fields available are the following: source IP address (`sip`), destination IP address (`dip`), source port (`sp`), destination port (`dp`), packet payload and timestamp (`ts`).

3 Conclusions and Future Work

In this work we have focused on an architecture to generate benchmarks for IDS. The proposed approach is able to generate real-world based datasets of network traffic, with both normal and abnormal network traffic flows that statically represent real flows in an organization. Such datasets can be used as benchmark to test, deploy and benchmark intrusion detection systems. The datasets are built upon a statistical model that shapes the distribution of real raw packets flows through time. Moreover, the architecture allows for outputs in raw format to directly test working intrusion detection systems or can be text-based, allowing further research in innovative algorithms for intrusion detection.

Future work is foreseen in further testing the architecture, both in real intrusion scenarios, as well as in developing new detection algorithms.

References

[1] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y., “Intrusion detection system: A comprehensive review”, Journal of Network and Computer Applications, 36(1), 16-24, 2013.

[2] Wu, S. X., & Banzhaf, W., “The use of computational intelligence in intrusion detection systems: A review”, Applied Soft Computing, 10(1), 1-35, 2010.

[3] Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K., “The 1999 DARPA off-line intrusion detection evaluation”, Computer networks, 34(4), 579-595, 2000.

[4] Massicotte, Frederic, et al. “Automatic evaluation of intrusion detection systems”, Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual. IEEE, 2006.

[5] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection”, Comput. Secur., vol. 31, no. 3, pp. 357–374, 2012.