Creation of specific flow-based training data sets for usage behaviour classification

Florian Otto, Markus Ring, Dieter Landes, Andreas Hotho
Coburg University of Applied Sciences, Coburg, Germany
University of Würzburg, Würzburg, Germany
florian.otto@hs-coburg.de
markus.ring@hs-coburg.de
dieter.landes@hs-coburg.de
hotho@informatik.uni-wuerzburg.de

Abstract: The majority of security methods for computer networks rely on well-known signatures for detecting malware and attacks. Consequently these methods often fail to detect novel or obfuscated attacks in monitoring data. Usage behaviour classification and anomaly detection seem to be promising approaches to address these problems. In this context, the development of sophisticated classifiers allowing to sift monitored data caused by harmless behaviour patterns can greatly improve the results of downstream anomaly detection methods. The former rely on a valid ground truth, defining harmless usage behaviour sufficiently. Recognising behaviour patterns also may facilitate the detection of insider threats, if single employees behave not as expected. We propose a workflow enabling us to create labeled flow-based data sets containing information about real usage behaviour. Within this workflow, real humans use virtual machines to work on specific tasks and simultaneously log their activities. In addition, the virtual machines log processes inducing network connections. Both logs are then used to attach labels to the monitoring data, to enrich it with information about the corresponding usage behaviour. We describe the process with an example scenario of an adapted computer pool of our university. Finally, the resulting data set is briefly discussed.

Keywords:
behaviour classification, data set generation, intrusion detection, anomaly detection, flow-based data

1. Introduction

Network Intrusion Detection Systems (NIDS) are used to ensure network security and to encounter attacks. NIDS can be distinguished in misuse detection and anomaly detection (Sommer and Vern, 2010). Misuse detection searches for well-known signatures of malware or attacks in network monitoring data (Giacinto et al., 2008). Signatures have to be constantly updated, since malware and attacks are constantly developed, refined and obfuscated (Giacinto et al., 2008). In contrast, anomaly detection tries to distinguish normal from abnormal behaviour. Sufficient recognition of harmless behaviour facilitates detection of malware, attacks and misuse, considering those lead to mutations within the data. Corresponding methods e.g. classifiers rely on representative training data sets containing a valid ground truth. Due to privacy concerns and difficulties to label network monitoring data, few publicly available data sets exist (Sommer and Vern, 2010).

To address this problem, we propose a workflow which allows to easily built networks and gather monitoring data within pre-defined scenarios. The networks are built in virtual environments that allow real humans to work with and to monitor the network in a controlled environment. Since we focus the enrichment of the data with information about corresponding usage behaviour, the users log their activities manually. Additionally, the virtual hosts log processes inducing network traffic. Our environment allows to adapt networks, which can than be used for specific scenarios without interfering with productive systems. Pre-defined scenarios may be defined as the normal usage of the real counterpart network to gather harmless behaviour, or special situations which may appear more seldom. Penetrations tests or attacks can also be performed or malware could be installed to see effects within the monitored data.

Our main contribution is a way to create network traffic of adapted productive networks in a controlled environment and to enrich it with information about real usage behaviour.

2. Related Work

Intrusion detection data sets can be distinguished in packet-based, flow-based and application-based. Since the proposed data set is flow-based, the following review considers only public available flow-based data sets.

One of the first labeled flow-based data sets for intrusion detection was contributed by Sperotto et al. (2009). Flows were collected from a real honeypot with HTTP, SSH and FTP services. The log files of the services were used to label the corresponding flows. The resulting data set contains over 14 million flows, but most of them are suspicious due to missing further background traffic.

Shiravi et al. (2012) describe a systematic approach to generate adapted data sets for intrusion detection. Their

basic idea is to describe profiles which contain detailed descriptions of normal user activities and attacks. Based on these profiles, they generated and published the ISCX data set.
Another publicly available flow-based data set is CTU 13 Malware (García et al., 2014). In this data set, 13 different botnet scenarios are mixed with background traffic. The data is labeled based on the IP-addresses used by the botnets.
In contrast to the above, we enrich our data set with information about real user behaviour based on manually provided activity protocols and with information about processes inducing network traffic.

3. Data Set Generation

3.1 Toolset

Our approach is based on OpenStack which allows the creation and management of virtual networks and virtual machines. Within this environment we are able to adapt existing networks or subnetworks of organisations and to record their corresponding network traffic. Real users which work on the virtual hosts are connected via remote connections that are routed over a single special host, enabling us to easily sift traffic caused by the remote connections.

3.2 Flow-based Monitoring

Widespread standards of flow-based monitoring data are Netflow and Internet Protocol Flow Information Export (IPFIX). Flows represent connections between two network components and are mostly identified by the default five-tuple (Protocol, Source- and Destination Address, Source- and Destination Port) (Kim et al., 2004). Flows encapsulate different metrics, e.g. number of sent packets and bytes, the duration and used TCP-Flags. The encapsulation of flow-based monitoring significantly reduces the amount of data to be stored in comparison with full-packet-monitoring. Additionally, no contents of communication are stored leading to less privacy concerns.

3.3 Labelling

Our goal is to create monitoring data sets enriched with information about usage behaviour. Therefore, we use a two-fold labelling strategy: Firstly, we monitor processes and applications using sockets on the host machines. This allows us to label the flows with applications which induced them. Secondly, human users are instructed to log their activities. Informal and detailed logs would result in complicated extraction information. Although concise terms contain less information they are easier to evaluate. Further, participants should not be distracted too much by writing protocols, since this would distort their behaviour. In due we provide a small extensible taxonomy of terms with short descriptions, which the participants could select. This labelling procedure is quick and flexible enough to adapt specific behaviours. Activities may overlap since it is likely to do things in parallel with computers e.g. searching for information while communicating. We attempt to create realistic data sets, so we don't restrict the usage to one activity at a time. However, this may lead to multiple activity labels per flow.

4. Scenario

As an example scenario, we adapted student workplaces of a computer pool at the university. We prepared six virtual machines with Ubuntu 14.04 LTS. Within a two hour timeframe, the students were asked to do research, work on current projects or work on online tutorials about programming languages. The experimentees were also asked to perform tasks they do in leisure time e.g. streaming videos, using social networks, etc.

5. Resulting Data Set

In the experiment five of the six hosts were used by human participants. One of the hosts was active but not used (*User B* in Figure 1). An overview of the activities during the experiment is given in Figure 1.

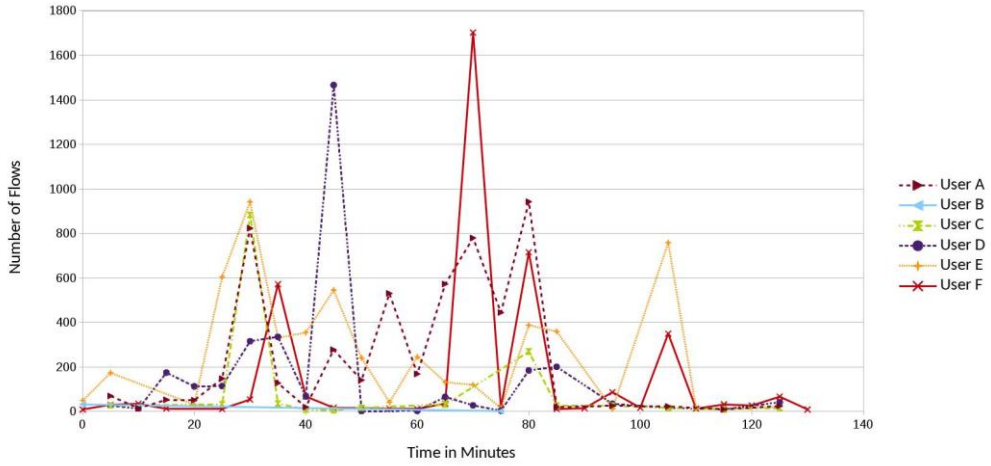

Figure 1: User activity represented as number of flows within 5 minute intervals

The resulting data set contains *19,222* bidirectional flows. The majority, *19,180* (*99.78 %*) of the flows were induced by the hosts used by the participants.

Table 1: Process labels within the dataset

Process Name	Number of Flows
firefox	7,001
plugin-container	98
thunderbird	7
ssh	5
unknown	12,111

As shown in Table 1, *7,111* (*36.99%*) flows are labeled unambiguous with host processes. For the larger proportion of flows no process could be identified (*unknown)*. However, *11,306* (*93.35%*) of these can be lead back to the local domain name service of the environment. Apart from that almost all user activity which can be seen in the flow data is induced by the web-browser, since Firefox and its plugin-container dominate the whole data set.

Table 2: The top ten used terms describing activities during the experiment

Activity	Number of flows	Percentage
surf the web	7,750	40.32
research	6,015	31.29
IRC (webclient)	3,731	19.41
online gaming	2,447	12.73
listening music	2,038	10.60
reading news	1,805	9.39
file transfer	1,199	6.24
online tutorials	884	4.60
videostreaming	538	2.80
mailclient	274	1.43

Table 2 shows the top ten different labels for activities which were chosen by the participants during the experiment. Note that the labels overlap and exceed the total number of flows. The two most frequent labels are *surf the web*, which was described as using the web for private interests and *research* meaning the aimed search for information about specific topics. One of the participants used the internet relay chat (IRC) for

communication within a browser application. The participant used that in the usual way having the application active for almost the whole time, even if it was only rarely used. Due to that, almost every flow of that participant carries that label, which explains the high proportion.
The resulting data set shows that the label for processes introduces little information, since almost all user activity results in flows induced by the web-browser. Since many network services next to the classical presentation of internet pages, e.g. video-streaming, music-streaming or communication can be used within browsers, we expect similar outcomes in other scenarios. Nevertheless, if sophisticated networks are adapted where special software is used or if participants can use their own preferred software, this label will carry more information.

6. Future Work

An open question is if the usage of virtual environments lead to significant differences in the monitoring data compared to data which is created by physical components. The usage of specific applications or tasks should not vary, since there are few differences for human users. However variations in temporal behaviour of the components or other effects are not ruled out.
Further, we plan to adapt more modular networks to create a wide range of data sets describing different scenarios. This includes performing penetration tests, infecting hosts with malware and simulating insider threats.
Finally, we want to use these data sets to train sophisticated classifiers for distinguishing between normal and abnormal usage behaviour in productive networks.

7. Conclusion

We proposed a workflow and toolset enabling us to create labeled flow-based training data sets for usage behaviour classification. Therefore an OpenStack environment is used to adapt networks and to perform scenarios in which networks can be analysed without interfering with their real counterparts.
We focus the enrichment of the data with information about corresponding usage behaviour, which is why real users participating in scenarios log their activities based on a simple taxonomy.
We briefly discussed an example data set which we generated in a small scenario.

References

García, S., Grill, M., Stiborek, J., & Zunino, A. (2014) "An Empirical Comparison of Botnet Detection Methods", *Computers & Security*, Vol. 45, pp 100-123.
Giacinto, G., Perdisci, R., Del Rio, M., and Roli, F. (2008) "Intrusion Detection in Computer Networks by a Modular Ensemble of One-Class Classifiers", *Information Fusion*, Vol. 9, No. 1, pp 69-82.
Kim, A. S., Kong, H. J., Hong, S. C., Chung, S. H., and Hong, J. W. (2004) "A Flow-based Method for Abnormal Network Traffic Detection", *Network operations and management symposium (NOMS)*, IEEE/IFIP, Vol. 1, pp 599-612.
Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. (2012) "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", *Computers & Security*, Vol. 31, No. 3, pp 357-374.
Sommer, R. and Vern, P. (2010) "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection", *Security and Privacy (SP)*, 2010 IEEE Symposium on, IEEE, pp 305-316.
Sperotto, A., Sadre, R., Van Vliet, F., and Pras, A. (2009) "A Labeled Data Set For Flow-based Intrusion Detection", *Proc. of the 9th IEEE Int. Workshop on IP Operations and Management (IPOM)*, Springer, pp 39-50.