ISCX
Information Security
Centre *of* Excellence

Botnet Analysis Framework

Saeed Nari, Ali A. Ghorbani

Faculty of Computer Science, University of New Brunswick

UNB

Introduction

Botnets have become a major threat to the security of today's Internet. Botnets consist of compromised hosts connected to the Internet which are controlled by an attacker known as bot-master to perform malicious activities. An attacker uses a malware program known as botware to initially infect the victim hosts and populate the botnet. The infected hosts which are known as bots communicate with botmaster through a communication protocol . In order to create an effective botnet analysis system, it is crucial to expressively and efficiently model the behavior of different botnet entities such as bots, botwares and botmasters.

In this research we propose an automated behavior-based framework for botnet analysis operating at both host and network level.

Objectives

The following objectives are considered in designing the proposed framework:

- The framework should be able to deal with new bots as well as the modified versions of existing bots. Therefore, a signature-based approach is not suitable and high-level behaviors of bots and botnets should be captured to be able to characterize new threats.
- The framework should be general. It should not be limited to a specific protocol, structure or application. To achieve this goal we will profile the general behavior of bots and botnets instead of defining specific features which represent specific types of bots or botnets.
- It should deal with botnets which use encrypted communication. For this purpose we take two approaches. In the first approach, only non-encrypted data are considered. Second approach would be considering metrics that are resilient to encryption such as entropy.
- Our framework should characterize individual bots as well as small botnets. It should not require communication between a large number of bots to identify a botnet.
- The proposed framework should be feasible for real-world deployment. For this purpose, we intend to use lightweight methods for tracking user input instead of tainting analysis which is very resource intensive.
- Our framework should be able to accurately characterize bot and botnet behavior. To this end, we will investigate various features and metrics and introduce profiling schemes such as graph models which can represent the behavior of bots and botnets better than existing models.

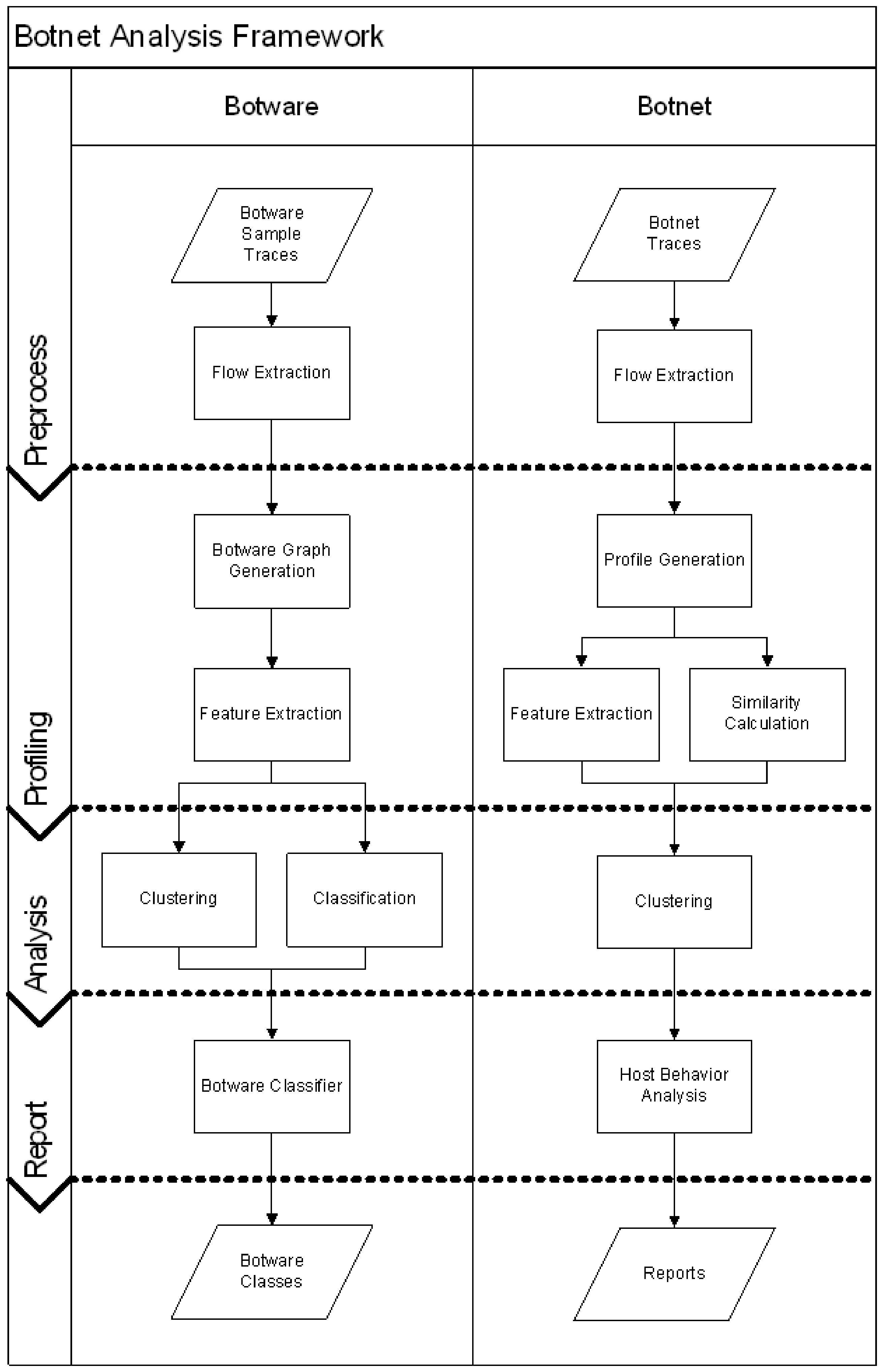
Botnet Analysis Framework

Preprocess: Raw data needs to be preprocessed to be used in the subsequent steps. Different data types such as network traces or IDS alerts can be fed into the analysis framework. An example of preprocessing is extracting flow information from network traces.

Profiling: Extracted features and data processed in the previous step are used to create behavioral profiles which abstract the host and network level behavior of different entities in a botnet such as bots, botwares and botnets themselves.

Analysis: This part consists of different analysis engines for different profiles. For example known botwares are used for training a classifier which is able to identify botware family from behavior profiles. A clustering engine is also used to characterize new botwares.

Report: This component is responsible for generating reports that are easily interpreted by a human analyst. Analysis reports include botware and bot clusters which represent the similarity between different bots or botwares and extend our knowledge of their behavior.


Botnet Analysis Framework

Botware
Botnet
Preprocess
Profiling
Analysis
Report
Botware Sample Traces
Flow Extraction
Botware Graph Generation
Feature Extraction
Clustering
Classification
Botware Classifier
Botware Classes
Botnet Traces
Flow Extraction
Profile Generation
Feature Extraction
Similarity Calculation
Clustering
Host Behavior Analysis
Reports

Implementation and Results

As a proof of concept implementation of our framework we implemented a malware analysis system. In the implemented system we took the following steps:

1. First we extracted network flow information from traffic traces captured during the execution of malware samples. Flow information include IP address, port number, and protocol.
2. Flow information were processed to generate behavioral models by abstracting network flows and their dependencies to a graph representation.
3. Features reflecting the network behavior of malware samples were extracted from malware graphs.
4. Using a dataset of labeled malware samples, a classifier was trained in order to identify malware family from network traces.

To evaluate the accuracy of our classifier we created a labeled dataset by voting between the labels reported by different anti-virus programs. Each malware sample was labeled only if 7 out of 11 anti-virus programs agreed on the malware family of that sample. With this method we created a dataset of 3347 malware samples including 12 malware families. We used 66% of the samples for the training set and the rest for the validation set. The overall accuracy over the validation set was 91.57.

To measure the detailed accuracy of each class we measured the ROC area of our classifier for each malware family.

ROC Area	Family
1	FUJACK
1	MYDOOM
1	PCCLIENT
0.999	PORNDIALER
0.99	P2PWORM
0.985	LDPINCH
0.98	HUPIGON
0.945	SWIZZOR
0.928	BIFROSE
0.905	ARDAMAX
0.857	BANKER
0.839	BANLOAD

Conclusions

In this research we proposed a framework for automated botnet analysis based on host and network level behavior with the following features:

- The proposed framework is independent of botnet structure and communication protocol
- It is capable of analyzing botnets with encrypted communication
- It is able to deal with new botnets types and new variants of botwares.
- It can characterize different entities in a botnet such as bots and botwares.

A proof of concept implementation of the framework shows that our approach is accurate and effective for real-world deployment.

References

[1] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos, "Profiling the End Host," in *Passive and Active Network Measurement*, vol. 4427, S. Uhlig, K. Papagiannaki, and O. Bonaventure, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 186-196.

[2] P. McDaniel, S. Sen, O. Spatscheck, J. Van der Merwe, W. Aiello, and C. Kalmanek, "Enterprise security: A community of interest based approach," in *NDSS*, 2006.

[3] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using Machine Learning Techniques to Identify Botnet Traffic," in *Proceedings 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 967-974.

[4] E. Stinson and J. Mitchell, "Characterizing bots' remote control behavior," *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 89–108, 2007.

[5] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: detecting malware infection through IDS-driven dialog correlation," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, Boston, MA, 2007, pp. 1-16.

[6] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th conference on Security symposium*, San Jose, CA, 2008, pp. 139-154.

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

[8] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, Cambridge, MA, 2007, pp. 7-7.

[9] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *Proceedings of the 10th international conference on Recent advances in intrusion detection*, Gold Goast, Australia, 2007, pp. 178-197.

Contact

s.nari@unb.ca