UNIVERSITY OF HASSELT

BACHELOR THESIS

Machine learning techniques for flow-based network intrusion detection systems

Author:
Axel FAES

Supervisor:
Prof. Dr. Peter QUAX
Prof. Dr. Wim LAMOTTE
Bram BONNE
Pieter ROBYNS

Bachelorproef voorgedragen tot het behalen van de graad van bachelor in de informatica/ICT/kennistechnologie

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in Computer Science

February 24, 2016

universiteit hasselt

KNOWLEDGE IN ACTION

Declaration of Authorship

I, Axel FAES, declare that this thesis titled, "Machine learning techniques for flow-based network intrusion detection systems" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a bachelor degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UNIVERSITY OF HASSELT

Abstract

Wetenschappen
Computer Science

Bachelor of Science in Computer Science

Machine learning techniques for flow-based network intrusion detection systems

by Axel FAES

Acknowledgements

Contents

List of Figures

List of Tables

List of Abbreviations

IDS	**I**ntrusion **D**etection **S**ystem
IPS	**I**ntrusion **P**revention **S**ystem
IDPS	**I**ntrusion **D**etection (and) **P**revention **S**ystem
NIDS	**N**etwork (based) **I**ntrusion **D**etection **S**ystem
HIDS	**H**ost (based) **I**ntrusion **D**etection **S**ystem
DDOS	**D**tributed **D**enial **o**f **S**ervice
ML	**M**achine **L**earning

List of Symbols

Chapter 1

Nederlandse Samenvatting

Chapter 2

Introduction

The internet is constantly growing and new network sevices arise constantly. This has as effect that security flaws become more and more important. Considering this, it becomes more important to be able to detect and prevent attacks on network systems.

2.1 Intrusion detection systems

An intrusion detection system is a system which tries to determine whether a system is under attack, to detect intrusions within a system. There are different types of intrusion detection systems or IDS. There are network-based intrusion detection systems and host-based intrusion detection systems. This thesis will uses machine learning techniques to detect malicious network behaviour, as such only network-based intrusion detection systems are covered.

2.1.1 Host-based Intrusion Detection Systems

Host-based intrusion detection systems are systems that monitor the device on which they are installed. The way they monitor the system can range from monitoring the state of the main system through log files, to monitoring program execution. In this way they can be quite indistinguishable from Anti-Virus programs.

2.1.2 Network-based Intrusion Detection Systems

Network-based intrusion detection systems are placed at certain points within a network in order to monitor traffic from and to devices within the network. The system can analyse the traffic using multiple techniques to determine whether the data is malicious. There are two different ways to analyse the network data. The analysis can be packet-based or flow-based.

Packet-based analysis uses the entire packet including the headers and payload. An intrusion detection system that uses packet-based analysis is called a packet-based network intrusion detection system. The advantage of this type of analysis is that there is a lot of data to work with. Every single byte of the packet could be used to determine whether the packet is malicious or not. The disadvantage is immediately obvious once we look at networks through which a lot of data passes, such as data centers. Analysing every byte is very work-intensive and near impossible to do in such environments.

Flow-based analysis doesn't use individual packets but uses general data about network flows. An intrusion detection system that uses flow-based analysis is called a flow-based network intrusion detection system. A flow is defined as a single connection between the host and another device. A flow can be defined using a (source_IP, destination_IP, source_port, destination_port) tuple. However flowdata also contains other information such as the duration of the connection, the start time, the amount of bytes and/or packets within the flow. Flow data can even contain data such as the amount of SYN packets within the flow. This could be useful to detect SYN overflow attacks. However not every flow collector collects this data Since flow data is much more compact than all the individual packets, it is much more feasable for data centers to use flow-based intrusion detection systems.

2.1.3 Intrusion Prevention Systems

An intrusion prevention system or IPS/IDPS is an intrusion detection system that also has to ability to prevent attacks. An IDS does not necessarily need to be able to detect attacks at the exact moment they occur, although it is preferred. An IPS needs to be able to detect attacks real-time since it also needs to be able to prevent these attacks. For network attacks these prevention actions could be closing the connection, blocking an IP, limiting the data throughput.

2.2 IP Flows

Flows are aggregated from all packet data that travels through the network. A flow is not the same as a TCP connection. A flow can be any communication between two devices with any protocol. Flows are defined using a (source_IP, destination_IP, source_port, destination_port) tuple. However, the port data is not always required. This is why flows are also called IP Flows.

Since flow data does not contain any payload information, intrusion detection systems that use flow data cannot detect malicious behaviour embedded within payload data. [1]

Chapter 3

Attack Classification

An intrusion detection system can use multiple methods to detect malicious behaviour. Since flow-based intrusion detection systems only have access to the flows and not the payload, they cannot detect every kind of attack. In order to make the IDS as effective as possible, the exact classifications of attacks that can be detected need to be known.

3.1 Classification

There are several types of attacks that can occur. Some of these attacks occur only on the network, other attacks infect computers, called malware. The exact classifications are not mutually exclusive. Some types of malware utilise network attacks. However it is important to make a distinction between these attacks. Every attack is identified by different characteristics. Knowing these characteristics is usefull to be able to tweak the IDS to make identification more effective.

3.2 network attacks

There are **Physical attacks**, these are attacks which attempt to destroy physical equipment and hardware. **Buffer overflows** are attacks that try to execute arbritrairy code or crash a process by overflowing a buffer on the targeted system. **Password attacks** attempts to break into a system by gaining the password that the system uses. The simplest password attacks are brute-force password crackers. **DDOS** attacks are attacks which attempt to make a network resource temporarily or permanently unavailable for the users of that resource. An attack could happen by flooding a system with TCP SYN packets. **Network scans** are information gathering attacks. They do not cause any damage by themselves but usually serve the purpose to gather information about a system that could be used in further attacks. Network traffic sniffing or port scans are examples of network scans.

3.3 Malware

There are several types of malware. We can make four distinct categories of malware. There are **botnets**, **viruses**, **trojan horses** and **worms**. Malware are actual programs that infect a system to execute a specific task. The task of the malware defines which categorie the malware belongs in.

Trojan horses are programs disguised as harmless applications but contain

malicious code. **Worms** are programs that replicate themselves among a network. They can spread extremely fast. **Viruses** are similar to worms. However they only replicate themselves on the infected host computer. Thus they require user interaction in order to be spread around a network. The virus can accomplish this by attaching itself to an email-attachment, embed itself within an executable, etc.

Botnets is malware that causes infected computers to become "slaves" to the master. An infected computer is controlled externally by the bot-master without the knowledge of the owner of the infected computer. The bot-master can use the distributed network of "slave" computer to perform other malicious tasks, such as performing an DDOS attack.

3.4 Detection

An NIDS only monitors the network. As such not every attack can be detected by an NIDS. Only the attacks that actually use the network can be detected. Flow-based IDS have the additional constraint that they can only use flow data. This further limits the attacks that can be detected. The attacks that can be detected using a flow-based network intrusion detection systems are:

- Botnets
- Worms
- DDOS
- Network scans

Chapter 4

Machine learning

Chapter 5

Machine learning for an IDS

5.1 Using ML for an IDS

An intrusion detection system has to detect whether some data it receives is either malicious or regular web traffic. This can be seen as a classification problem which means an machine learning algorithm for classification could be used. It needs to be determined whether data is either normal network traffic or malicious behaviour.

Some parameters have to be chosen that will be feed into the machine learning algorithm.

5.2 Disadvantages of using ML for an IDS

5.2.1 Problems

As said before, machine learning for an intrusion detection system is a classification problem. More precisely, it can be said that intrusion detection systems have to detect abnormal behaviour in a network with mostly normal behaviour. There are several problems that can be encountered when using machine learning techniques.

The first problem is the ability to detect new attacks. A machine learning algorithm compares incoming data with a model that it has created internally. An new type of malicious behaviour might appear to be closer to normal network traffic as compared to the model of known attacks.

Another problem is the diversity of network traffic. The notion of "normal network traffic" is difficult to actually define. The bandwidth, duration of connections, origin of IP addresses, applications used can vary enormously through time. This makes it quite difficult for machine learning algorithms to distinguish between "normal network traffic" and malicious behaviour.[2]

5.2.2 Solutions

There are several solutions that can be used in order to make machine learning algorithms more effective for intrusion detection systems. One option is to chance the way the classification problem is defined. Instead of defining the classes, "normal" and "malicious", there might be different classes for different types of malicious behaviour. In the same way, different classes can be defined for different types normal traffic.

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection.", *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 343–356, 2010. [Online]. Available: http://dblp.uni-trier.de/db/journals/comsur/comsur12.html#SperottoSSMPS10.

[2] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection", in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, ser. SP '10, Washington, DC, USA: IEEE Computer Society, 2010, pp. 305–316, ISBN: 978-0-7695-4035-1. DOI: 10.1109/SP.2010.25. [Online]. Available: http://dx.doi.org/10.1109/SP.2010.25.