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 26, 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]

2.2.1 Flow collection

2.2.2 Flow exporting

2.3 Why is machine learning interesting

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 arbitrairy 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:

- DDOS
- Network scans
- Worms
- Botnets

Other attacks either do not use network communication, or they are not visible within the header information of network traffic.

3.5 Distributed Denial of Service

3.6 Network scans

3.7 Worms

3.8 Botnets

Chapter 4

Machine learning

4.1 What is machine learning

Machine learning is ... .

4.2 Machine learning algorithms

4.3 Supervised learning

4.3.1 Classification

Uses discrete categories

4.3.2 Regression

4.4 Unsupervised learning

4.4.1 Clustering

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.

5.3 Advantages of using ML for an IDS

Chapter 6

Flow data

6.1 How to use flow-data

The following attributes are available with flow-data:

- Source IP
- Destination IP
- Protocol name
- Source port
- Destination port
- Starting time of the flow
- Duration of the flow
- Amount of packets in the flow
- Amount of bytes in the flow

However, should an flow exporter be implemented, some additional features can be generated from packet data. 2.2.2

- Amount of TCP SYN within the flow
- Source and Destination Type of Service
- Payload size

These data can be used within the machine learning algorithms. However some variables have undesirable effects on the accuracy of the algorithm. Some care should be taken when training the machine learning algorithms with the additional data. Not all data, both training data as predictive data, will have the additional features.

Most machine learning libraries use numeral data instead of string data. All string data has been hashed in order to be able to use it in machine learning algorithms. The probability on a collision is low enough to be able to ignored.

6.1.1 IP addresses

6.1.2 Ports and protocol name

Both the source and destination port are discrete data. They are usually received in decimal form, however some data-sets might use them in hexadecimal data or refer to ports as "ssh port" instead of "22". Port data, in decimal form, can be directly fed into the machine learning algorithm.

The protocol name can simply be converted to a standard string in lower case, in order to avoid errors by lower and uppercase forms of the same name (for example "tcp" and "TCP"). This string can than be hashed into a discrete value.

6.1.3 Timing

6.1.4 Size

The amount of packets used in the flow and the amount of bytes are both discrete data. They are always received in decimal form. They can immediately be fed into the machine learning algorithm.

Chapter 7

Prevention

This chapter will only be done if this is made in the thesis

7.1 Real-time detection

7.2 Data limiting

7.3 Connection closing

Chapter 8

Implementation

8.1 Structure

8.2 Class diagram

Chapter 9

Datasets

9.1 Cegeka

9.2 CTU Datasets

9.3 Own generation

9.4 Inline placement

Chapter 10

Visualisation

10.1 Logging

10.2 Graphing

Chapter 11

Conclusion

Appendix A

Meetings

A.1 Meeting 1: 9 Feb 2016

aanwezigen: Peter Quax, Bram Bonne, Pieter Robyns, Axel Faes

Dit is de eerste bijeenkomst met de begeleiders en promotor. Er is dus geen rapportering mogelijk van een vorige bijeenkomst. Tijdens de bijeenkomst is beslist om een *intruder detection system* te bestuderen en te implementeren.

De actiepunten die gedaan moeten worden:

- Beslissen voor wie het systeem gemaakt moet worden. Gaat dit voor end users zijn, of voor grote data centers. Hieraan hangt vast welke data (packets of netflow) gebruikt moet worden.
- Bekijken hoe machine learning algoritmes gebruikt kunnen worden in een *intruder detection system*.
- Bekijken wat netflow is.
- Er moet gekeken worden naar de manier waarop anomalies gegenereerd gaan worden om het systeem te testen/trainen.

Volgende afspraken zijn gemaakt:

- Er is gevraagd om te zorgen dat het systeem ook op correcte wijze informatie kan weergeven aan gebruikers. Tijdens het semester moet bekeken worden hoe deze weergave moet gebeuren.
- Libraries gebruiken indien mogelijk, om te vermijden dat het wiel opnieuw uitgevonden word.
- Er is de mogelijkheid geboden om aan de thesis te werken op het EDM.
- Er is afgesproken om *Overleaf* te gebruiken om de thesis in te schrijven.
- Een ruwe planning voor het werk moet gemaakt worden tegen 12 Feb.
- Een wekelijkse meeting is vastgelegd. Dit om 10:00 elke vrijdag.
- Begin mei moet een eerst draft van de thesis klaar zijn en eind mei moet de finale draft af zijn.
- Er moet een vulgariserende tekst gemaakt worden en een postersessie gegeven worden (op 29 juni).

A.2 Meeting 2: 12 Feb 2016

aanwezigen: Bram Bonne, Pieter Robyns, Axel Faes

Dit is de tweede bijeenkomst met mijn begeleider. Netflow bevat op zichzelf niet zoveel informatie, maar het is toch handig om te kijken welke bevindingen gemaakt kunnen worden met deze data. Mogelijks kan er, indien gevonden wordt dat netflow alleen niet genoeg informatie bevat, ook gebruikt gemaakt worden van packet data.

Er is de mogelijkheid besproken om eventueel meerdere machine learning algoritmes te implementeren en te bekijken in welke situaties welke algoritmes beter werken.

De actiepunten die gedaan zijn:

- *Beslissen voor wie het systeem gemaakt moet worden.*: Dit gaat gedaan worden voor data centers
- Er zijn verschillende classificaties van machine learning algorithmes gevonden die gebruikt kunnen worden.
- Verschillende grote data sets van netflow en packets met sporen van anomalies zijn gevonden. Alsook programma's om verkeer te genereren.

Volgende actiepunten zijn besproken:

- Verder uitwerken van welke machine learning algoritmes gebruikt kunnen worden
- Bekijken netflow v9

A.3 Meeting 3: 19 Feb 2016

aanwezigen: Bram Bonne, Pieter Robyns, Axel Faes

Professor Quax is aan het bekijken ofdat ik (gelabelde) netflow data kan verkrijgen van Cegeka. Dit zou heel handig zijn om mijn implementatie te testen op real world data.

Voorlopig moet ik enkel focussen op een passive intrusion detection systeem, geen preventie en niet direct inline in het netwerkverkeer. Ook de visualisatie moet later bekeken worden, de gebruiker is een netwerkadministrator. Er is tevens besproken dat python zelf mogelijks te traag is om packet sniffing op een goede snelheid uit te voeren. Hiervoor zou ik wireshark kunnen gebruiken (of de command line versie). Er is besproken om eventueel zelf datasets te genereren door malware te runnen op een VM of aparte machine.

De datastructuur voor de machine learning algoritmes is bekeken. Ik moet eens bekijken hoe de timestamps van de flowdata gebruikt kunnen worden. Om de effectiviteit (van de machine learning algoritmes) mogelijks te verhogen ga ik eens bekijken of ip-adressen ingedeeld kunnen worden in country-of-origin of iets dergelijks. Dit zou de machine learning algoritmes de mogelijkheid bieden om ook op deze parameter te bekijken of data malicious is of niet.

De actiepunten die gedaan zijn:

- Er is al een basis implementatie uitgewerkt voor het IDS
- De netflow structuur is bekeken en er is een datastructuur opgestelt die gefeed kan worden aan verschillende machine learning algoritmes.
- Progressie in de machine learning cursus: chapter 3 van de 18.

Volgende actiepunten zijn besproken:

- Beginnen aan de thesis: het schrijven van een hoofdstuk over machine learning en over hoe deze algoritmes toegepast kunnen worden op een intrusion detection systeem.
- Verder werken in de machine learning cursus.
- Ik moet eens bekijken ofdat ik een programma vind om pcap files om te zetten naar netflow. Anders moet ik dit zelf schrijven.

Ik heb ook een korte planning gemaakt van hoe de thesis eruit zou zien:

- Inleiding:
  - wat is een IDS
  - Waarom is er gekozen voor dit type IDS (host vs netwerk)
  - Waarom voor data centers
  - Waarom netflow
  - Waarom machine learning
- Wat is machine learning
- Hoe passen we machine learning toe op IDE en wat zijn de voor/nadelen
- Welke machine learning algortimes zijn wel/niet gebruikt
- Wat zijn de voor/nadelen van netflow
- Hoe met combinatie netflow/packets (Als dit gedaan zou worden)
- Welke data sets zijn gebruikt
- Wat zijn de bevindingen
- Hoe kan visualisatie/feedback gebeuren (richting admin en richting automatische preventie)
- Conclusie

A.4 Meeting 4: 26 Feb 2016

aanwezigen: Bram Bonne, Axel Faes

Deze week is voornamelijk besteed aan de implementatie. Er is een netflow exporter geschreven. Er is bekeken ofdat timestamps gebruikt kunnen worden en ofdat ip-adressen opgedeeld kunnen worden per land. Er is besloten dat dit zeer weinig effect heeft op de accuraatheid van de machine learning algoritmes.

Momenteel zijn Support vector machines en K-nearest Neighbor Classifier algoritmes bekeken. Het K-nearest Neighbor Classifier algoritme is zeer efficient (98%).

In een later stadium kan bekeken worden om eventueel verdere analyse te doen op de data die malicious gevonden is, eventueel door pakketten te analyseren, of nogmaals door machine learning technieken. Er kan ook eens bekeken worden om een VM op te zetten, en daarin malware te runnen en dit verkeer te monitoren. Herbij zouden eigen datasets gegenereerd kunnen worden.

De machine learning cursus is gevolgd tot hoofstuk 7. De cursus zou normaal af moeten zijn binnen 2 weken.

De actiepunten die gedaan zijn:

- Er is al een netflow exporter geschreven
- Er zijn experimenten uitgevoerd m.b.t de datastructuur die meegegeven wordt aan de machine learning cursus.
- Progressie in de machine learning cursus: chapter 7 van de 18.
- Er is begonnen aan de thesis.
- Het zou interessant zijn om eens te kijken ofdat ip-addressen opgedeeld kunnen worden in subnets.

Volgende actiepunten zijn besproken:

- Focussen op de thesis
- Verder werken in de machine learning cursus.

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.