



Kenzo Staelens

Jonathan van Caloen

Jonas Van den Berghe

Michaël Vasseur

Bachelor Elektronica-ICT Mentor: Tom Cordemans

2022-2023





Kenzo Staelens

Jonathan van Caloen

Jonas Van den Berghe

Michaël Vasseur

Bachelor Elektronica-ICT Mentor: Tom Cordemans

2022-2023

# Kwaadaardig internetverkeer onderscheppen met Python

K. Staelens, J. van Caloen, J. Van den Berghe, M. Vasseur

This project investigates a way to intercept malicious internet traffic and how a solution can be provided, with an aim for ease of use. By using a webserver as a gateway and a firewall built on software developed in python. Emphasis is placed on the third and fourth network layers.

The first chapter regarding malware is given to understand how malware works. Types of attacks and what effect they have are discussed here. Understanding how it works is a baseline to prevent malicious software from infecting a system and helps develop a solution. Subsequently the existing solutions are described and compared. The next chapter discusses the criteria for efficient malware interception and blocking.

From all the known data, a firewall will be developed that will efficiently analyze the network traffic and throw up the necessary actions. The coming chapters cover the technical side of the solution, aiming to provide a look at the firewall coded in python with a user-friendly interface. Responding to the end user's demand for ease of use.

In the final chapter the evaluation of the project is described. The pros and cons of combining this solution with artificial intelligence for a possible sequel are also stated.

Keywords: Firewall, infrastructure, packet analyser, webserver, malware, artificial intelligence, user-friendly interface

# inhoudsopgave

Inhoud

[figurentenlijst 3](#_Toc128228165)

[tabellenlijst 4](#_Toc128228166)

[codefragmentenlijst 5](#_Toc128228167)

[begrippenlijst 5](#_Toc128228168)

[afkortingenlijst 6](#_Toc128228169)

[Inleiding 7](#_Toc128228170)

[1. malware 8](#_Toc128228171)

[1.1 types 9](#_Toc128228172)

[1.2 tekenen van infectie 9](#_Toc128228173)

[1.3 gebruik 9](#_Toc128228174)

[1.4 Type aanvallen 9](#_Toc128228175)

[2. firewalls en antivirussoftware 9](#_Toc128228176)

[1.1 Firewalls 9](#_Toc128228177)

[1.2 Antivirussoftware 9](#_Toc128228178)

[3. mogelijke oplossingen 9](#_Toc128228179)

[3.1 criteria voor malware detectie 9](#_Toc128228180)

[3.2 mogelijke oplossingen 9](#_Toc128228181)

[3.2.1 python 9](#_Toc128228182)

[3.2.2 c# 9](#_Toc128228183)

[3.3 vergelijking 9](#_Toc128228184)

[4. uiteindelijke oplossing 9](#_Toc128228185)

[4.1 laag drie: Netwerk 10](#_Toc128228186)

[4.2 laag vier: Transport 10](#_Toc128228187)

[4.3 user interface 10](#_Toc128228188)

[5. evaluatie 10](#_Toc128228189)

[Conclusie 10](#_Toc128228190)

[handleiding 11](#_Toc128228191)

[Literatuurlijst 12](#_Toc128228192)

[Bijlagen 13](#_Toc128228193)

[bijlage 1: kopieën van datasheets 14](#_Toc128228194)

[bijlage 2: vergaderverslagen 15](#_Toc128228195)

[bijlage 3: logboek rapporteren 16](#_Toc128228196)

# figurentenlijst

# tabellenlijst

# codefragmentenlijst

# begrippenlijst

# afkortingenlijst

DoS Denial of Service

# Inleiding

In dit rapport wordt een mogelijke oplossing omschreven om kwaadaardig internetverkeer, beter bekend als malware, uit de pakketten te filteren die bij het netwerk toekomen. De software om deze taak tot een goed einde te brengen is vooral geschreven in Python. Ze loopt op een centrale server en fungeert ook als firewall.

Al het kwaadaardige verkeer dat gedetecteerd kan worden op netwerklaag drie door te kijken naar het IP-adres van de afzender wordt geblokkeerd. Dit is handig omdat er zo op basis van geolocatie bepaalde regio’s uitgesloten kunnen worden. Datzelfde gedeelte van de software laat ook de detective van sommige Denial of Service (DoS) aanvallen toe. Enkel pakketten bekijken op laag drie in echter niet voldoende. De analyse op laag vier zorgt ervoor dat ook de inhoud van het ontvangen pakket kan geanalyseerd worden. De inhoud van de pakketten zijn in het merendeel van de gevallen beschermd door HTTPS-encryptie. Er is dus een sleutel nodig om de vertaling van de pakketten mogelijk te maken. Dit is mogelijk doordat de server opgesteld is als een Man-in-The-Middle apparaat. Zo heeft de software toegang tot alle sleutels om de nodige vertalingen te doen.

De software om al deze zaken te doen bestaat al. Firewalls en antivirusprogramma’s kunnen deze taken zonder problemen volbrengen. Deze kennis wordt dan ook gebruikt doorheen dit project. Het doel van dit project is dan ook niet van iets volledig nieuw te ontwikkelen maar om zoveel mogelijk zaken bij te leren en deze te combineren tot een functioneel programma.

De belangrijkste onderzoeksmethode is literatuurstudie van onlinebronnen. Op deze manier kan veel kennis verworven over de bestaande oplossingen voor het onderscheppen van malware. Op de tweede plaats zijn vooral de eigen testen belangrijk. De interface is geëvalueerd door middel van gebruikerstesten.

In het eerste hoofdstuk wordt een introductie tot malware gegeven. Het is zeer belangrijk om te begrijpen hoe verschillende types werken en hoe de inhoud eruitziet alvorens deze te kunnen detecteren. Hierna volgt een korte omschrijving over hoe firewalls en antivirussoftware malware detecteren, onderscheppen en blokkeren. Een lijst van mogelijke oplossingen is de volgende stap in het rapport. De focus hier ligt op de programmeertaal, de netwerklagen en de nodige infrastructuur voor de applicatie. Op het einde van dit hoofdstuk worden de oplossingen met elkaar vergeleken. Het laatste hoofdstuk staat helemaal in het teken van de uiteindelijk gekozen oplossing. Hier worden technische aspecten besproken en wordt ook de werking geëvalueerd. Ten slotte volgt nog een voorstel tot een mogelijk vervolg. Hier worden de mogelijkheden van de combinatie met artificiële intelligentie besproken.

# 1. malware

## 1.1 types

## 1.2 tekenen van infectie

## 1.3 gebruik

## 1.4 Type aanvallen

# 2. firewalls en antivirussoftware

## 1.1 Firewalls

## 1.2 Antivirussoftware

# 3. mogelijke oplossingen

## 3.1 criteria voor malware detectie

## 3.2 mogelijke oplossingen

### 3.2.1 python

### 3.2.2 c#

## 3.3 vergelijking

# 4. uiteindelijke oplossing

## 4.1 laag drie: Netwerk

## 4.2 laag vier: Transport

## 4.3 user interface

# 5. evaluatie

# Conclusie

# handleiding

# Literatuurlijst

# Bijlagen

# bijlage 1: kopieën van datasheets

# bijlage 2: vergaderverslagen

# bijlage 3: logboek rapporteren

|  |  |  |
| --- | --- | --- |
| Jonas Van den Berghe | Schrijven eerste file | Schrijven inleiding, abstract, toevoegen titels, maken inhoudstafel. |
|  |  |  |
|  |  |  |
|  |  |  |