BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

NETWORK FORENSICS TOOLS SURVEY AND TAXONOMY

PRIESKUM A TAXONÓMIA SIEŤOVÝCH FORENZNÝCH NÁSTROJOV

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. MARTINA ZEMBJAKOVÁ**
AUTOR PRÁCE

SUPERVISOR **Ing. JÁN PLUSKAL**
VEDOUCÍ PRÁCE

BRNO 2021

Brno University of Technology
Faculty of Information Technology

Department of Information Systems (DIFS) Academic year 2020/2021

Master's Thesis Specification

Student: **Zembjaková Martina, Bc.**
Programme: Information Technology Field of study: Computer Networks and Communication
Title: **Network Forensics Tools Survey and Taxonomy**
Category: Networking
Assignment:

1. Research the literature to find existing taxonomies of network forensic tools and discuss their comparison.
2. Research existing network forensic tools and learn how to use them. Try to find new tools that are not mentioned in the literature survey from point 1.
3. Find available datasets that can be analyzed using the forensic tools in step 2. Describe the datasets in detail and compare them.
4. According to the information obtained from points 1 and 2, design frequent use cases for forensic tools and demonstrate them using the datasets found from point 2. Compare the results on the use cases mentioned in the literature from point 1.
5. Due to possible outdated datasets and published taxonomies, create datasets that will contain missing/updated network protocols and use cases.
6. Given the new tools discovered, design a new taxonomy or update the appropriate existing ones. Discuss and justify all decisions properly.

Recommended literature:

1. Khan, S., Gani, A., Wahab, A.W.A., Shiraz, M. and Ahmad, I., 2016. Network forensics: Review, taxonomy, and open challenges. *Journal of Network and Computer Applications*, *66*, pp.214-235.Vancouver
2. Pilli, E.S., Joshi, R.C. and Niyogi, R., 2010. Network forensic frameworks: Survey and research challenges. *digital investigation*, *7*(1-2), pp.14-27.

Requirements for the semestral defence:

- Items 1, 2 and 3

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Pluskal Jan, Ing.**
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: November 1, 2020
Submission deadline: May 19, 2021
Approval date: October 26, 2020

Abstract

This master's thesis addresses network forensic tools survey and taxonomy. It describes network forensics fundamentals, including network forensic process models, techniques, and evidence sources. Furthermore, the project contains a survey of existing network forensic tools taxonomies, including their comparison, followed by the network forensic tools survey. In addition to the tools mentioned in the taxonomy survey, the survey is extended to other network tools. Subsequently, the detailed description and comparison of available datasets that can be analyzed using the forensic tools are provided in this project. According to the information obtained from surveys, frequent use cases for forensic tools are designed, and the tools are demonstrated within the description of individual use cases. In addition to publicly available datasets, the demonstration also uses newly created datasets described in detail in its chapter. Based on the gained information, new taxonomy is designed. This taxonomy is based on the use cases of the tools in contrast to other taxonomies based on NFATs and NSM tools, user interface, capturing the data, analysis, or type of forensics.

Abstrakt

Táto diplomová práca sa zaoberá prieskumom a taxonómiou sieťových forenzných nástrojov. Popisuje základné informácie o sieťovej forenznej analýze, vrátane procesných modelov, techník a zdrojov dát používaných pri forenznej analýze. Ďalej práca obsahuje prieskum existujúcich taxonómií sieťových forenzných nástrojov vrátane ich porovnania, na ktorý naväzuje prieskum sieťových forenzných nástrojov. Diskutované sieťové nástroje obsahujú okrem nástrojov spomenutých v prieskume taxonómií aj niektoré ďalšie sieťové nástroje. Následne sú v práci detailne popísané a porovnané datasety, ktoré sú podkladom pre analýzu jednotlivými sieťovými nástrojmi. Podľa získaných informácií z vykonaných prieskumov sú navrhnuté časté prípady použitia a nástroje sú demonštrované v rámci popisu jednotlivých prípadov použitia. Na demonštrovanie nástrojov sú okrem verejne dostupných datasetov použité aj novo vytvorené datasety, ktoré sú detailne popísane vo vlastnej kapitole. Na základe získaných informácií je navrhnutá nová taxonómia, ktorá je založená na prípadoch použitia nástrojov na rozdiel od ostatných taxonómií založených na NFAT a NSM nástrojoch, uživateľskom rozhraní, zachytávaní dát, analýze, či type forenznej analýzy.

Keywords

taxonomy, survey, overview, network forensic tools, datasets, network forensics, GitHub Pages

Kľúčové slová

taxonómia, prieskum, prehľad, sietové forenzné nástroje, datasety, sieťová forenzná analýza, GitHub Pages

Reference

ZEMBJAKOVÁ, Martina. *Network Forensics Tools Survey and Taxonomy.* Brno, 2021. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Ján Pluskal

Rozšírený abstrakt

Táto diplomová práca sa zaoberá prieskumom a taxonómiou sieťových forenzných nástrojov. Cieľom práce je vykonať prieskum existujúcich taxonómií sieťových forenzných nástrojov a následne na základe získaných informácií vhodne aktualizovať existujúce taxonómie alebo vytvoriť novú taxonómiu. Práca okrem samotnej klasifikácie obsahuje aj prieskum dostupných sieťových nástrojov pre forenznú analýzu, prieskum dostupných datasetov, navrhnutie častých prípadov použitia nástrojov a ich demonštrovanie s použitím získaných datasetov a prípadne vytvorením nových aktualizovaných datasetov. Tieto dodatočné informácie rozširujú obzor sieťovej forenznej analýzy a tým pomáhajú k navrhnutiu taxonómie, ktorá sa dotýka aj praktickej časti forenznej analýzy a nie je založená len na teórií existujúcich taxonómií.

Začiatok práce sa venuje úvodu do sieťovej forenznej analýzy. Tento úvod obsahuje základné informácie zo sieťovej analýzy. Po definovaní sieťovej forenznej analýzy, s použitím niekoľkých definícií, sú popísané dva prístupy k procesu sieťovej forenznej analýzy, a to všeobecný procesný model a OSCAR metodológia vyšetrovania, ktorá na rozdiel od všeobecného procesného modelu obsahuje len päť fáz, ktoré sú však obsahovo podobné. Ďalej sú v tomto úvode obsiahnuté aj niektoré forenzné techniky. Záver tejto úvodnej kapitoly je venovaný zdrojom sieťových dát, v ktorých je možné nájsť dôkazy potrebné pre forenzné vyšetrovanie. Ako zdroj môžu slúžiť nefiltrované zachytené pakety, dáta relácií, alerty generované IDS systémami, či rôzne štatistické dáta.

Následná časť práce obsahuje vykonané prieskumy dostupných taxonómií, nástrojov a dát. Najskôr je detailne popísaných sedem taxonómií, ktoré boli získané z prieskumu dostupnej literatúry (odborných článkov a kníh venujúcich sa danej problematike). Tieto taxonómie sú zoradené chronologicky podľa času publikovania a najstaršia klasifikácia je taxonómia autora Simson Garfinkel publikovaná v roku 2002. Najnovšia získaná taxonómia je z roku 2019 a je definovaná agentúrou ENISA. Kapitolu venujúcu sa prieskumu dostupných taxonómií ukončuje porovnanie jednotlivých získaných taxonómií. Druhý prieskum naväzuje na ten predošlý. Ide o prieskum dostupných nástrojov, ktoré môžu byť použité pri sieťovej forenznej analýze. Obsahom sú nástroje, ktoré sa objavujú ako príklady ku kategóriám v existujúcich taxonómiách. Tento prieskum je rozšírený o nástroje, ktoré neboli spomenuté. Celkovo je popísaných 119 nástrojov. Každý nástroj okrem stručného popisu nástroja a odkazov na príslušné webstránky, obsahuje aj zaradenie do fáze OSCAR modelu, kedy môže byť užitočný v rámci sieťovej forenznej analýzy. Posledným vykonaným prieskumom je prieskum verejne dostupných datasetov. Tento prieskum obsahuje datasety z rôznych oblastí, rôznej veľkostí a dĺžke zachytenej komunikácie. Zahrnuté sú datasety obsahujúce DoS útoky, packet injection útoky, IDS dáta, ale aj šifrovaná komunikácia s dostupnými kľúčmi na dešifrovanie. Medzi datasetmi sa nachádza aj známy dataset zachytenej komunikácie šikanovania na univerzite Nitroba. Celkovo je popísaných šesť datasetov, z ktorých sa viaceré skladajú z niekoľkých PCAP súborov a sú ďalej rozdelené podľa konkrétnejších kritérií. Rovnako ako výskum venujúci sa dostupných nástrojom, aj tento výskum datasetov je zakončený porovnaním.

Druhá polovica práce začína definovaním častých prípadov použitia sieťových forenzných nástrojov a ich demonštrovaním na dostupných datasetoch. Navrhnuté a popísané prípady použitia sú vyvodené z informácií získaných zo všetkých predchádzajúcich prieskumov. Na začiatku je popísaný komplexný prípad sieťovej forenznej analýzy, ktorý je demonštrovaný na Nitroba datasete, kde je dôraz kladený na použité nástroje v jednotlivých častiach vyšetrovania. Následne sú popísané a demonštrované prípady použia ako nástroje na skenovanie, nástroje zachytávajúce komunikáciu, nástroje zamerané na vizualizáciu dát,

analyzačné nástroje, diagnostické sieťové nástroje, IDS/IPS systémy a SIEM systémy. Záver kapitoly zhrňuje popísané prípady použitia a dotýka sa taktiež porovnania s informáciami uvádzanými v prieskumoch.

Keďže datasety získané z prieskumu neobsahujú všetky potrebné dáta na demonštrovanie niektorých nástrojov, boli vytvorené nové datasety. Nové datasety boli vytvárané aj z dôvodu aktualizácie datasetov o dáta dnešnej sieťovej komunikácie. Takto bol vytvorený dataset obsahujúci historické dáta z webového prehliadača, či dataset venujúci sa zachytávaniu rovnakej komunikácie viacerými nástrojmi. Ako ďalší novo vytvorený dataset je dataset obsahujúci komunikáciu dnešnej doby, ktorá zahŕňa služby ako OpenVPN, TeamViewer, ESET prehliadač, Microsoft Mail či WinSCP. Časť tohto datasetu obsahuje taktiež komunikáciu, kde boli demonštrované diagnostické sieťové nástroje. Súčasťou datasetu reflektujúceho súčasnú sieťovú komunikáciu je aj dataset zachytávajúci komunikáciu vybraných mobilných aplikácií — "Ideme vlakom (ZSSK)", "Mapy.cz", "Sygic", "AliExpress", "Gmail", "Alza", a "Strava".

Následne je práca zavŕšená kapitolou popisujúcou novo navrhnutú taxonómiu sieťových forenzných nástrojov. Navrhnutá taxonómia sa opiera o všetky získané informácie z vykonaných prieskumov. Hlavný dôraz je kladený na prípady použitia. Všetky rozhodnutia sú riadne diskutované a odôvodnené zo získaných informácií popísaných v predchádzajúcich kapitolách. Štruktúra taxonómie pozostáva z hlavnej kategórie určujúcej prípad použitia, ktorej je priradených niekoľko ďalších kategórií určujúcich ďalšie vlastností nástroja. Takto je definovaných sedem hlavných kategórií – nástroje na skenovanie, nástroje zachytávajúce komunikáciu, nástroje zamerané na vizualizáciu dát, analyzačné nástroje, diagnostické sieťové nástroje, IDS/IPS systémy a SIEM systémy. Definované podkategórie sa líšia u jednotlivých hlavných kategórií, avšak niektoré sú spoločné. Medzi spoločné kategórie patrí napríklad užívateľské rozhranie, podporovaná platforma, či určenie či sa jedná o bezplatný alebo komerčný nástroj. Súčasťou navrhnutej taxonómie je aj priradenie nástrojov k jednotlivým kategóriám. Celkovo je v taxonómií priradených 12 nástrojov na skenovanie, 17 nástrojov zachytávajúcich komunikáciu, 9 vizualizačných nástrojov, 30 analyzačných nástrojov, 18 diagnostických nástrojov, 8 IDS/IPS systémov a 4 SIEM systémy.

Záver práce zhrňuje výsledky práce a rovnako popisuje výstup časti práce v podobe webovej stránky, ktorá tvorí pridanú hodnotu tejto práce. Táto webová stránka je verejne dostupná v podobe GitHub Pages na https://martinazembjakova.github.io/Network-forensic-tools-taxonomy/ a jej obsahom je výstup prieskumov, ktoré sa venujú dostupným sieťovým nástrojom a datasetom.

Network Forensics Tools Survey and Taxonomy

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Jan Pluskal. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Martina Zembjaková
May 10, 2021

Acknowledgements

I want to express thanks to the supervisor Ing. Jan Pluskal, for providing valuable knowledge, my family for support, and all who provided me valuable help.

Contents

1 Introduction **4**

2 Network forensics **6**
2.1 Definition 6
2.2 Network forensic process 6
2.2.1 Generic process model 7
2.2.2 OSCAR investigative methodology 9
2.3 Network forensic techniques 11
2.3.1 Traceback NFTs 12
2.3.2 Converge Network NFTs 12
2.3.3 Attack Graph NFTs 13
2.3.4 Distributive NFTs 13
2.3.5 Intrusion Detection System NFTs 13
2.4 Network-based evidence 14

3 State-of-the-art network forensic tools taxonomies **15**
3.1 Taxonomy according to Simson Garfinkel 15
3.1.1 The “catch it as you can” approach 15
3.1.2 The “stop, look, and listen” approach 16
3.2 Taxonomy according to IJNSA 16
3.2.1 Email Forensics 16
3.2.2 Web Forensics 17
3.2.3 Packet Sniffers 17
3.3 Taxonomy according to “Digital Investigation” 17
3.3.1 Built-in commands 17
3.3.2 Network Forensic Analysis Tools 17
3.3.3 Network Security and Monitoring Tools 18
3.4 Taxonomy according to Davidoff and Ham 18
3.5 Taxonomy according to “Fundamentals of Network Forensics” 20
3.5.1 Network Forensic Analysis Tools 21
3.5.2 Network Security and Monitoring Tools 21
3.6 Taxonomy according to IOSR-JCE 21
3.7 Taxonomy according to ENISA 22
3.8 Comparison 23
3.9 Summary 23
3.9.1 Pilli and Joshi taxonomies 24

4 Network forensics tools **25**

4.1 Tools from the previous literature survey . . . 25
4.2 More tools . . . 51
4.3 Summary and comparison . . . 58

5 Survey of datasets **59**
5.1 Canadian Institute for Cybersecurity datasets . . . 59
5.1.1 CIC-DDoS2019 . . . 59
5.1.2 CIC-IDS2017 . . . 63
5.2 Nitroba University Harassment Scenario . . . 66
5.3 NETRESEC Packet Injection Attacks . . . 67
5.3.1 Packet injection attack against *www.02995.com* . . . 67
5.3.2 Packet injection attack against *id1.cn* . . . 68
5.4 ICS Cybersecurity — DoS Attacks against SCADA-based systems . . . 68
5.4.1 Nominal state . . . 69
5.4.2 ARP-based, Man-in-the-Middle attack . . . 69
5.4.3 Modbus query flooding . . . 69
5.4.4 ICMP flooding . . . 70
5.4.5 TCP SYN flooding . . . 70
5.5 WireShark SampleCaptures . . . 71
5.5.1 SSL with decryption keys . . . 71
5.6 Summary and comparison . . . 72

6 Use cases and demonstration **76**
6.1 Nitroba scenario . . . 76
6.1.1 Obtain and strategize . . . 77
6.1.2 Collect . . . 77
6.1.3 Analyze . . . 78
6.2 Scanners . . . 82
6.2.1 Port scanners . . . 82
6.2.2 Web scanners . . . 84
6.2.3 Wifi scanners . . . 85
6.2.4 Vulnerability scanners . . . 85
6.3 Sniffers . . . 86
6.3.1 Windows tools . . . 86
6.3.2 Summary and comparison . . . 88
6.4 Visualizers . . . 89
6.4.1 CapAnalysis . . . 89
6.4.2 Graphical Ping (NetScanTools) . . . 93
6.4.3 EtherApe . . . 95
6.4.4 PcapXray . . . 96
6.4.5 Web Historian . . . 99
6.5 Analyzers . . . 103
6.5.1 Packet injection attacks . . . 103
6.5.2 Encrypted traffic . . . 103
6.5.3 SMTP traffic . . . 104
6.6 Network diagnostic tools . . . 104
6.7 IDS/IPS . . . 105
6.8 SIEMs . . . 106

6.9 Summary and comparison . . . 108

7 New datasets **110**
7.1 Captured traffic with multiple tools . . . 110
7.2 Dataset with today's protocols . . . 112
7.2.1 OpenVPN traffic . . . 113
7.2.2 Diagnostic tools traffic . . . 115
7.2.3 Mobile applications traffic . . . 115
7.3 Web browser history data . . . 116

8 New taxonomy **118**
8.1 Scanners . . . 119
8.2 Sniffers . . . 119
8.3 Visualizers . . . 120
8.4 Analyzers . . . 121
8.5 Network diagnostic tools . . . 121
8.6 IDS/IPS and SIEMs . . . 121

9 Conclusion **129**
9.1 Website representing the results . . . 129

Bibliography **131**
Network forensic tool references . . . 131
Datasets references . . . 139
Other references . . . 141

Acronyms and Abbreviations **142**

Appendices **146**

A Details of the PCAP files from "DoS Attacks against SCADA-based systems" dataset **147**
A.1 Nominal state . . . 147
A.2 ARP-based, Man-in-the-Middle attack . . . 147
A.3 Modbus query flooding . . . 149
A.4 ICMP flooding . . . 153
A.5 TCP SYN flooding . . . 156

B Results from Nikto tool **160**

C GitHub Pages design **163**

D Contents of the included storage media **167**

Chapter 1

Introduction

As Maria Dailey says: "Today's forensic tools are characterized by their variety of functions, so determining what you need accomplished is key to making the right choice [185]."

Many network tools can assist in the investigation, whether they are tools designed specifically for forensic analysis or common network security tools. There are also attempts to include these tools in thematic taxonomies to make it easier to get an overview of the existing tools that can be used [188, 194, 198, 186, 189, 192, 187]. This project aims at the network forensics tools survey and taxonomy.

The beginning of this project focuses on the fundamentals of network forensics. Firstly, the network forensics is defined. The following section of chapter 2 describes the network forensic process models. The generic process model and OSCAR investigative methodology are discussed. Furthermore, the network forensic techniques are discussed, including traceback, converge network techniques, attack graphs, distributive techniques, and IDS techniques. The end of this chapter reviews network-based evidence and the sources of this evidence.

There are several articles about the network forensic tools taxonomy. Chapter 3 reviews the found taxonomies and network tool categories. One of the first attempts to classify the tools used for network forensic analysis was published on O'Reilly Network by Simson Garfinkel [188]. The other tool classifications were published in the following journals: International Journal of Network Security & Its Applications (IJNSA) [194], Digital Investigation [198], and IOSR Journal of Computer Engineering (IOSR-JCE) [192]. Moreover, the books that contain network forensic tools taxonomy are "Network Forensics: Tracking hackers through cyberspace" [186] and "Fundamentals of Network Forensics" [189]. ENISA also describes some well-known tools for network forensic analysis in the document handbook for "Trainings for Cybersecurity Specialists" called "Introduction to Network Forensics" [187]. All mentioned taxonomies are compared at the end of this chapter.

The individual network forensic tools gathered from the previous literature survey are described in detail in chapter 4. In addition to the already mentioned tools, other network tools are also described, including open-source and commercial tools.

To demonstrate use cases for network forensic tools, there is a need to have datasets that can be analyzed using some network tools. Some public datasets are described in detail in chapter 5. The end of the chapter deals with the comparison of the described datasets.

Chapter 6 starts with the detailed description of the concrete investigation scenario focusing on the used tools. Description of other use cases according to the information obtained from previous surveys from chapter 3 and 4 follows. Designed frequent use cases

include scanners, sniffers, visualizers, analyzers, network diagnostic tools, IDS/IPS, and SIEMs. Most use cases also include the demonstration of tools.

After network forensic tools, datasets, and taxonomies surveys, and after designing frequent use cases, new taxonomy is designed. This taxonomy is discussed in chapter 8 and is based on the existing taxonomies focusing on use cases.

Chapter 7 contains newly created datasets with the data that extends the existing datasets from chapter 5. A new type of data is included, such as web browser history. In addition, protocols that are part of today's network traffic are included. Furthermore, captured network traffic using several sniffers also appears between these new datasets that are part of the thesis outputs.

Final Chapter 9 summarizes this project's results. The website based on the chapters 4 and 5, which is an output of performed surveys, is also briefly described. This website is created using the GitHub Pages environment.

Chapter 2

Network forensics

The fundamentals of network forensics are presented in this chapter. After the definition of network forensics, the network forensic process is described, where the generic process model and OSCAR investigative methodology are discussed. The following section is focused on network forensic techniques. The last section of this chapter describes network-based evidence and the sources of this evidence.

2.1 Definition

Network forensics can be defined as a part of digital forensics with the focus on networks. It deals with the network traffic data acquired by active or passive network devices. The sources of network evidence are described in section 2.4. The main goal of network forensics is to investigate the network evidence to determine if there was a security incident or other anomaly, provide evidence about the incident and document the whole investigation [190].

Network forensics can be classified as a cross-discipline of digital forensics and communication networks [201]. It is the science that deals with the capture, recording, and analysis of network traffic for detecting intrusions and investigating them. Its concept deals with the network data (data found across a network connection, both ingress and egress traffic) and analyzes them [198].

Palmer defines network forensics as the use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities [197].

The forensic network can be extended to include mobile networks, cloud computing, the Internet of Things (IoT), industrial control systems, and software-defined networks (SDNs). In addition to tools for common network forensics, this modern network forensics also uses tools developed specifically for this specific environment [201].

2.2 Network forensic process

Network forensics is a science that investigates network evidence, as described in the definitions. This whole investigation is a process that consists of many steps. Individual steps

can differ in different types of models. It took time for the process model for network forensics was defined.

At the beginning of network forensics, there were used process models defined for general digital forensics. The first attempt at using the digital forensic process model in a network environment was taken up in the first Digital Forensic Research Workshop [196]. The investigative process in digital forensic science contained the following steps: identification, preservation, collection, examination, analysis, presentation, and decision. There were also other attempts to improve this process model or create new ones, but all were applied only in general digital forensics [198].

The first general process model for network forensics was proposed in 2005 and contained the following steps: capture, copy, transfer, analysis, investigation, and presentation. In 2010, another generic process model for network forensics was introduced based on several existing forensics models. This new model was proposed as a part of the journal article "Network forensic frameworks: Survey and research challenges" [198].

Another process model that can be used for network forensics is OSCAR investigative methodology. This model was defined in 2012 with the following steps: obtain information, strategize, collect evidence, analyze, and report [186].

These last two models, the generic process model from 2010 and the OSCAR model, are described in more detail in the following sections.

2.2.1 Generic process model

Generic process model that was introduced in the "Network forensic frameworks: Survey and research challenges" expands the previous models and adds other process steps. This generic process model can be seen in figure 2.1. The first five phases including Incident Response work with the real-time network traffic data. The other phases are part of the post-attack investigation. The following sections describing the generic process model are based on the original Pilli's research [198].

Preparation

The preparation phase ensures that all network monitoring tools are prepared for use. It checks that the network security tools are placed in strategic points on the network.

Detection

The detection phase helps in attack discovery. It observes the alerts generated by the security network tools and analyzes noticed anomalies and unauthorized events. This phase also includes quick validation of the attack to avoid false alarms. It can provide the incident response.

Collection

The collection phase collects the network traffic data acquired from the sensors. There is a huge expectation for the system memory, gathering maximum evidence with the minimum impact on the victim, and handling different log data formats.

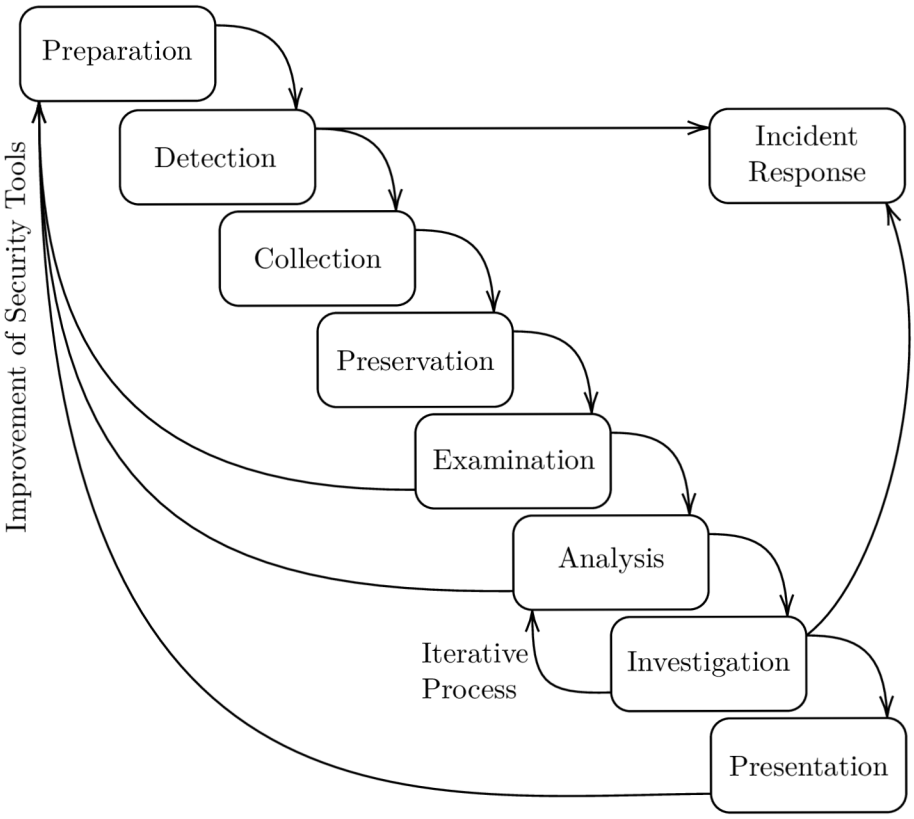


Figure 2.1: Generic process model

Preservation

The preservation phase ensures data integrity. The originally captured data are stored on a backup device with reading access only. There is also a preserved hash of all trace data. The investigation and analysis are done only on the copied evidence.

Examination

The examination phase provides a copy of the packet capture file for investigation. It integrates and fuses evidence data from all sources into one large data set. The evidence is also searched for specific attack indicators.

Analysis

The analysis phase classifies and correlates the attack indicators. There are used the following approaches for searching the data and matching attack patterns: statistical, soft computing, and data mining approaches.

Investigation

The investigation phase determines the traceback. The obtained packet captures and statistics are used for the attribution of the attack. The investigation phase works together with

the previous analysis phase when some additional features are required. This phase provides data for incident response.

Presentation

The presentation phase results in the attacker's prosecution based on the data provided in the investigation phase. The observations are presented with explanations of the procedures used to arrive at a conclusion. There is also included systematic documentation and visualized conclusions.

Incident Response

After the detection phase, a suitable incident response is generated based on the nature of the attack. There is also an action plan on how to defend against future attacks, recover from existing damage, and decide to continue investigating and gathering more information. After the investigation phase, there may be needed to take some actions to control and mitigate the attack.

2.2.2 OSCAR investigative methodology

OSCAR investigative methodology is an overall step-by-step process model for network forensics defined in the book "Network Forensics: Tracking hackers through cyberspace". The following figure 2.2 visualizes the OSCAR investigative methodology. The acronym "OSCAR" reflects the first letters of the individual steps of this process model [186].

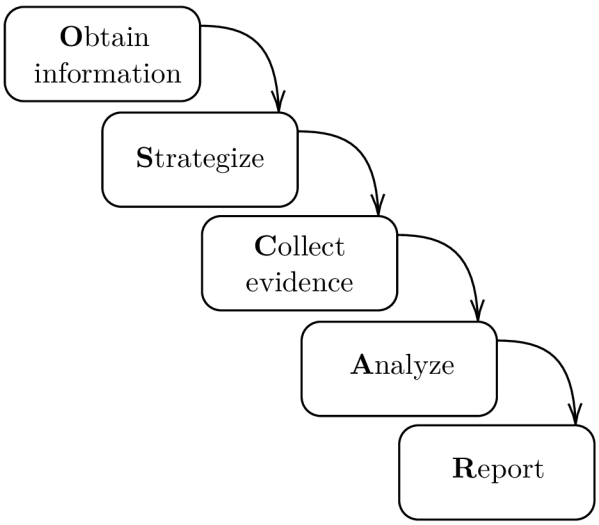


Figure 2.2: OSCAR investigative methodology model

Obtain information

Obtaining information before the investigation is the first thing the investigator needs to do. It is necessary to obtain information both about the incident itself and about the environment [186].

The information obtained about the incident can differs for individual incidents and may contain [186]:

- description of what happened,
- date, time, and method of incident discovery,
- persons, systems, and data involved,
- actions taken since discovery,
- summary of internal discussions,
- legal issues,
- time frame for investigation/recovery/resolution,
- goals.

Knowing about the environment is also essential because every environment is constantly changing. The information about the environment may contain [186]:

- business model,
- legal issues,
- network topology,
- sources of network evidence,
- organizational structure,
- incident response management process/procedures,
- communications systems,
- resources available.

Strategize

It is essential to plan the investigation, especially in network forensics, because of the evidence of many sources and the evidence's volatility. The strategizing the investigation makes the work more effective [186].

Authors of the book "Network Forensics: Tracking hackers through cyberspace" provide some tips for developing an investigative strategy [186]:

- understand the goals and time frame of the investigation,
- list resources (including personnel, time, and equipment),
- identify likely sources of evidence,
- for each source of evidence, estimate the value and cost of obtaining it,
- prioritize evidence acquisition,
- plan the initial acquisition/analysis,
- decide upon the method and times of regular communication/updates,
- after initial analysis there may need to be acquired more evidence (iterative process).

Collect evidence

After prioritizing the sources of evidence and creating an acquisition plan, the evidence collection can be started. When acquiring the evidence, the following three components should be addressed [186]:

- *Document* — keeping a careful log of actions taken during the evidence collection and addressed systems (recording date, time, source, method of acquisition, name of the investigator, and chain of custody),
- *Capture* — capturing the evidence (packets, logs, hard drive images),
- *Store/Transport* — the collected data should be stored securely and maintain the chain of custody.

Analyze

Although the analysis process can differ for each case study, in the analysis phase, there should be considered at least the following elements [186]:

- *Correlation* — what data should can be compiled, from which sources, and how it can be correlated,
- *Timeline* — building the timeline of activities (who did what, when, and how),
- *Event of Interest* — isolating more relevant events and seeking to understand how they transpired,
- *Corroboration* — identifying "false positives",
- *Recovery of additional evidence* — widening the net of evidence acquisition and analysis,
- *Interpretation* — developing working theories of the case.

Report

The last step of this methodology is the report. Providing a report of the whole investigation (starting the obtain step until the whole analysis of the collected evidence) is an essential part of network forensics. The report should be written in nontechnical language to be understandable by managers, judges, juries, and people with a nontechnical background. It should also be defensible in detail and factual [186].

2.3 Network forensic techniques

In network forensics, several techniques can be used to investigate cybercrimes. The network forensic techniques can be studied on the basis of forensic process models, forensic tools, and forensic frameworks. The forensic process models were discussed in the previous section 2.2, and the network forensic tools are mentioned in chapter 4. The following section deals with the network forensic frameworks.

The authors of the article "Network forensics: Review, taxonomy, and open challenges" discuss network forensics techniques (NFTs), including defining a thematic taxonomy of

network forensic techniques. Their review describes the fundamental mechanism of network forensics techniques to determine how network attacks are identified in the network. They focus on network forensic frameworks. Reviewed network forensics techniques include traceback based NFT, converge network-based NFT, attack graphs based NFT, distributive based NFT, and NFT using IDS [191].

The individual NFTs are described in detail in the following sections.

2.3.1 Traceback NFTs

Traceback, also called IP traceback, is the identification of the origin of packets in a network by investigating the attack path. This technique can be especially useful for DDoS and IP spoofing attacks. The following traceback techniques were discussed [191]:

- *Topology assisted deterministic packet marking technique* — works against DoS and DDoS attacks; uses hash correction code forensic approach and packet marking; can detect known attacks,
- *Collaborative forensics scheme* — VoIP traffic investigation; determines fake values in SIP request; uses network operator records methodology, can detect unknown attacks,
- *Network forensic evidence acquisition scheme* — provides integrity to the collected evidence; works with authenticated evidence and flow-based selection marking scheme; packet marking approach; can detect known attacks,
- *Lightweight IP traceback scheme* — uses TTL field of IP header; traces DDoS attacks; packet marking approach; can detect known attacks,
- *Scalable network forensic scheme* — self-propagating attacks identification; scalable based network forensic approach; logging approach; can detect known attacks,
- *Hopping based spread spectrum technique* — identifies attacks in anonymous communication; can detect unknown attacks,
- *IP traceback protocol* — real-time attack investigation; uses real-time and periodic analysis; logging and packet marking approach; can detect unknown attacks.

The authors of the article "Tools and Techniques for network forensics" describe two varieties of link testing techniques: input debugging and controlled flooding. The ICMP traceback and packet marking techniques are also discussed as a part of the IP traceback techniques. Moreover, they described the source path isolation engine architecture, that is the architecture implementing the log-based IP traceback technique [194].

2.3.2 Converge Network NFTs

The converge network NFTs aims the identification of digital evidence in converge networks, especially in VoIP communication. The individual techniques include [191]:

- *Pattern based network forensics technique* — identification of attack patterns; log correlation forensic approach; logging approach, can detect known attacks,
- *VoIP network forensics analysis with digital evidence procedure* — identification of malicious packet in network traffic; logging approach; can detect known attacks,
- *VoIP Evidence Model* — reconstructs attack path; logging approach; can detect known and unknown attacks.

2.3.3 Attack Graph NFTs

Attack graphs are used to identify and visualize all possible attack paths in the network. They consists of vertices and edges that represent attack nodes and state transition between different nodes. Moreover, the worst attack paths can be visualized in this graph. The individual attack graph NFTs include [191]:

- *Scalable analysis approach* — identify attack and their impact on enterprises; dependency graph; can detect known and unknown attacks,
- *Attack graph for forensic examination* — monitor attacker actions; anti-forensics approach; can detect known attacks,
- *Multi-level and layer attack tree* — investigation of multi-level attacks; network modeling approach; can detect known attacks,
- *Fuzzy cognitive map* — identification of worst attack; uses finite cognitive map; generic algorithm approach; can detect known attacks,
- *Cost-benefit security hardening* — root cause of the attack identification; design model forensic approach; probabilistic approach; can detect known attacks.

2.3.4 Distributive NFTs

Distributive NFTs perform the investigation, act on emerging responses, identify origin of the attack, and perform evidence collection. Network forensics server collects data from distributed data agents at different locations on network. The individual techniques include [191]:

- *ForNet framework* — distributive analysis; bloom filter tracking; logging approach; can detect known attacks,
- *Distributive agent based real-time network intrusion forensics system* — real-time network intrusion analysis; log and network traffic analysis; logging approach; can detect known attacks,
- *Distributive cooperative network forensics model* — integrity and validity of evidence; mapping topology and network attack statistics; logging approach; can detect known attacks,
- *Dynamical network forensics framework* — integrity and authenticity of evidence; multi-immune theory; logging approach; can detect known attacks.

2.3.5 Intrusion Detection System NFTs

Intrusion Detection System NFTs are network forensic techniques using IDS systems. The IDS triggers an alert about the security incident. The individual IDS techniques include [191]:

- *Analytical intrusion detection framework* — identification of unidentified signature rules; probabilistic model and interference; can detect unknown attacks,
- *Dynamic forensics intrusion tolerance modeling system* — forensic server tolerance; formal methods and analysis; can detect known attacks,

- *Intrusion investigation framework with data hiding* — monitoring log files; stenography; logging approach; can detect known attacks,
- *Network forensics based on intrusion detection analysis* — credibility and reliability for evidence; multi-dimensional analysis; logging approach; can detect known attacks.

2.4 Network-based evidence

There are many types of network-based evidence that can be used for network forensic analysis. The ENISA distinguishes four types of network-based evidence [187]:

- *full content data* — non-filtered packet captures (PCAPs), the data are analyzed using tools like tcpdump or Wireshark (using filters),
- *session data* — aggregated traffic metadata into flow records, provides information about conversations between two network entities without looking at the content (answers questions like *who talked to whom, when, for how long*),
- *alert data* — data generated by Intrusion Detection Systems (IDS), they are triggered when the traffic matches predefined patterns,
- *statistical data* — provides network-related aspects (such as the number of bytes contained in a packet trace, start and end times of network conversations, number of services and protocols being used, most active network nodes, least active network nodes, outliers in network usage, average packet size, average packet rate, and so on); they can be used for anomaly detection, this data are generated by tools such as Wireshark.

Other network-based evidence may include CAM tables, routing tables, NAT tables, port mirroring information from switches, SNMP data, firewall logs, DHCP mapping, DNS queries, authentication logs, web surfing habits, cached web pages, web server's logs, database server's logs, email server's logs, chat server's logs, VoIP server's logs, and logs from central log servers [186].

Collecting network-based evidence can be active or passive. Active acquisition, gathering evidence by interacting with systems on the network, includes for example sending queries, systems logging to a log host, SIEM or management station, and scanning the network. The passive acquisition, gathering evidence without emitting data at OSI Layer 2 or above, includes traffic capturing and sniffing [187].

The network-based evidence can be acquired from many sources. There are several devices or software-based tools that can act as sources, such as network taps and hubs, switches, routers, DHCP servers, naming servers (DNS), authentication servers, IDS/IPS, firewalls, web proxies, application servers, and central log servers [186].

Chapter 3

State-of-the-art network forensic tools taxonomies

After a brief introduction to the network forensics described in the previous chapter 2, it can be seen that this part of digital forensics deals with the huge amount of data evidence that should be analyzed. Many network forensic techniques were described in chapter 2, but for effective forensic analysis, there are also needed tools that can provide the investigator with a detailed analysis of input network-related data.

It is the use of tools that speed up the network forensic analysis, and in many cases, the key evidence can be found by using the right network tool.

This chapter provides an overview of existing network forensic tools taxonomies – classifications of the network tools used in network forensics. The taxonomies are described in the order they were published, starting with the one published in 2002, and the latest described classification is from 2019.

3.1 Taxonomy according to Simson Garfinkel

One of the first attempts to classify the tools used for network forensic analysis was published on *O'Reilly Network*[1] in 2002, written by Simson Garfinkel.

Simson Garfinkel describes two approaches to monitoring and recording network traffic data. He also discussed tools that use this approach, and therefore this classification can be considered a basic taxonomy of network forensic tools based on the approach of monitoring and recording the network data. This taxonomy is shown in table 3.1 [188].

3.1.1 The "catch it as you can" approach

The first approach he describes is called "catch it as you can". This method is based on the collection of the real-time network traffic data and recording them if possible. Based on the fact that this approach tries to capture all the data that passes through the network in real-time, it is clear that it requires a huge amount of storage place. For this purpose, there are usually used RAID systems [200, 188].

This captured network traffic data are then analyzed in the batch mode, which means that all data are analyzed at once. Also because of this batch mode, this method consumes a lot of storage space [193].

[1] http://www.oreillynet.com/

He divides the tools that use this "catch it as you can" approach into two categories – open-source and tools used in commercial systems.

3.1.2 The "stop, look, and listen" approach

Another similar approach is to examine all the traffic data but record only those that seem to be relevant to further analysis. The advantages of this method include monitoring more information (examining more information that is archived, and therefore it is effective also for busy networks), and privacy (the network traffic can also contain sensitive information, and since these data could not be stored in the disk, the possibility of misuse of this information is reduced) [188].

Consequently, the "stop, look, and listen" approach also includes the situation when it is not legal to record information unless for some specific reason, like a court order [188].

"catch it as you can"		**"stop, look, and listen"**
Commercial	NetVCR	Network Flight Recorder (NFR)
	NetIntercept	SillentRunner
Open Source	tcpdump	snort intusion detection system
	windump	NetWitness
		„Carnivore" Internet wiretapping system

Table 3.1: Network forensic tools taxonomy according to Simson Garfinkel

3.2 Taxonomy according to IJNSA

In the journal article "Tool And Techniques For Network Forensics" published by the *International Journal of Network Security & Its Applications (IJNSA)* in 2009, in addition to network forensic techniques, the authors also surveyed network forensic tools and discussed some categories. They distinguish three classes into which network forensic tools can be classified: Email forensics, Web forensics, and Packet sniffers. This classification is visualized in table 3.2, and individual categories are described in the following sections [194].

Email forensics	**Packet sniffers**	**Web forensics**
emailTrackerPro	AirPcap	Index.dat analyzer
SmartWhoIs	Ethereal	Web Historian
	WinPcap	

Table 3.2: Taxonomy according to IJNSA

3.2.1 Email Forensics

This part of the classification includes tools that can be used to investigate email traffic as a part of email forensics. The aim of email forensics is gathering the email relative information like the sender, the recipients, the content of the email, the timestamps relative to the email, and analyzing other email-related information relevant for investigators. For example, tools can identify the sender of an email, trace the path traversed by the message, help with investigating the spam of phishing emails, or provide other details of email communication [194].

3.2.2 Web Forensics

The Web Forensics part of the classification focuses on web traffic. The tools included in this category can provide the investigator with details of the web traffic, such as the browsing history, cookies, downloaded and uploaded files, the number of times the website was visited, and other web information [194].

This survey is based on the most used web browsers when this survey was taken place: the Internet Explorer (IE) and the Mozilla Firefox.

3.2.3 Packet Sniffers

The last category of classification includes packet sniffers. These tools capture the packets from the network traffic and allow collecting the traffic information from the packets. The sniffers are used for data acquisition in the investigation process [194].

3.3 Taxonomy according to "Digital Investigation"

The journal article "Network forensics frameworks: Survey and research challenges" was published in the *Digital Investigation* in 2010. Authors of this article, E. S. Pilli, R. C. Joshi, and R. Niyogi, distinguish tools that can be used for network forensic analysis between Network Forensic Analysis Tools (NFATs) and Network Security and Monitoring tools (NSM Tools). The overview of this taxonomy is visualized in table 3.3 [198].

Moreover, there are many built-in commands in operating systems that can provide the investigator with useful information for forensic analysis, and also these commands are part of the NFATs [198].

3.3.1 Built-in commands

Built-in commands are commands included in operating systems, and there is usually no need to install any other tools. These commands can be found in modern operating systems and can differ for each operating system.

Although these commands are not designed for forensics, they can be helpful in gathering data, analyzing the data flow, or in other phases of the network forensic process. The article describes the following common built-in commands: dig, nbtstat, netstat, nslookup, ping, tcpslice, traceroute/tracert, and whois [198].

3.3.2 Network Forensic Analysis Tools

Network Forensic Analysis Tools are designed for network forensics. These tools are primarily intended for network forensic analysis. NFATs help security administrators to monitor their network environment for unusual traffic and perform forensic analysis. The advantage of NFATs is that they can work with many sources and therefore combine more sources and, as a result, produce a more detailed overview of the analyzed environment [202].

NFATs allows users to analyze the entire captured network traffic according to their needs. The data are categorized for more comfortable work with them. The captured network traffic packets can be viewed as individual transport layer connections between hosts, and therefore the user can analyze protocol layers, packet contents, retransmitted data, and extract traffic patterns between various hosts [198].

The article divides NFATs into Open Source and Proprietary tools. Disadvantages of NFATs tools may include that there are not a lot of open-source tools, and many NFATs are paid [198].

Although paid tools often provide more features and better performance, many open source tools provide an acceptable environment for analyzing the network forensic data. Another benefit of commercial products is their reporting capability [202].

3.3.3 Network Security and Monitoring Tools

The second part of the classification is Network Security and Monitoring Tools. NSM tools are not primarily designed for network forensics. These tools were developed to provide network security, and although they do not have a forensic standing, they can help a lot in the investigation process [198].

According to this article, the NSM Tools can distinguish between the following subcategories based on the primary function of the tools: Fingerprinting, IDS, Manipulation, Packet Capture, Pattern Matching, and Statistics [198].

NFATs	
Open Source	NetworkMiner
	PyFlag
	Xplico
Propiertary	DeepSee
	InfiniStream
	Iris
	NetDetector
	NetIntercept
	NetWitness
	OmniPeek
	SilentRunner

NSM Tools	
Fingerprinting	Nmap
	P0f
IDS	Bro
	Snort
Manipulation	TCPReplay
	SiLK
Packet Capture	Argus
	flow-tools
	NfDump
	Nessus
	PADS
	Sebek
	TCPDump
	TCPFlow
	Wireshark
Pattern Matching	Ngrep
	TCPXtract
Statistic	NetFlow
	Ntop
	TCPDstat
	TCPStat
	TCPTrace

Table 3.3: Network forensic tools classification published in the *"Network forensics frameworks: Survey and research challenges"* article

3.4 Taxonomy according to Davidoff and Ham

Davidoff and Ham are the authors of the book "Network Forensics: Tracking hackers through cyberspace" published in 2012. This book contains the network forensic fundamen-

tals, and traffic analysis. Moreover, the network devices and servers, and some advanced topics like network tunneling or malware forensics are discussed. The tools used during the investigation process are part of the traffic analysis part [186].

According to the book's traffic analysis part and evidence acquisition chapter, the network forensic tools classification can be described as shown in the following table 3.4. There can be noticed that some tools are used in more than one category. The tool's multi-usability can cause this issue and the fact that the tools are classified according to the tool's functionality and phases of the network forensic process [186].

WAP discovery tools	
Open-source	KisMAC
	Kismet
	NetStumbler
Proprietary	Skyhook

IDS/IPS	
Open-source	Bro
	Snort
Proprietary	CheckPoint IPS-1
	Cisco IPS
	Corero Network Security
	Enterasys IPS
	HP TippingPoint IPS
	IBM Security NIPS
	Sourcefire 3D System

Traffic acquisition
dumpcap
libpcap
tcpdump
tshark
winpcap
Wireshark

Packet analysis	
Protocol Analysis Tools	tshark
	Wireshark
Packet Analysis Tools	Bless
	ngrep
	tshark
	Wireshark
Flow Analysis Tools	pcapcat
	tcpflow
	tcpXtract
	tshark
	Wireshark
Higher-layer Traffic Analysis Tools	findsmtpinfo.py
	NetworkMiner
	oftcat
	smtpdump

Statistical flow analysis	
Sensors	Argus
	softflowd
	yaf
Flow Record Export Protocols	IPFIX
	NetFlow
	sFlow
Collection Systems	Argus
	flow-tools
	nfdump
	NfSen
	SiLK (flowcap, rwflowpack)
Flow Record Analysis Tools	Argus Client Tools *(ra, racluster, ragraph, ragrep, rahisto, rasort)*
	EtherApe
	FlowTraq
	flow-tools
	nfdump
	NfSen
	SiLK *(PySiLK, rwcount, rwcut, rwfilter, rwidsquery, rwpmatch, rwstats, rwuniq)*

Table 3.4: Network forensic tools classification according to Davidoff and Ham

One of the categories is "Traffic acquisition." This category contains tools that are used during the acquisition of network evidence. Another category is "Packet analysis," which focuses on the analysis of the packets. This category distinguishes the following

subcategories: Protocol Analysis Tools, Packet Analysis Tools, Flow Analysis Tools, and High-layer Traffic Analysis Tools. Furthermore, the next category is based on the statistical flow analysis and consists of the subcategories: Sensors, Flow Record Export Protocols, Collection Systems, and Flow Record Analysis Tools. There can also be seen the "WAP discovery tools" category, which contains tools that can detect wireless access points. The last category contains IDS and IPS divided into open-source and proprietary tools [186].

3.5 Taxonomy according to "Fundamentals of Network Forensics"

Authors of the book "Fundamentals of Network Forensics," R. C. Joshi and E. S. Pilli, discussed the network forensic tools in the fourth chapter called "Network Forensic Tools". This book was published in 2016. The book section divides forensic tools into two categories, NFATs and NSM Tools, similar to the previously described taxonomy in section 3.3, where the authors are the same [189].

Unlike the previous taxonomy, the subcategories in the NSM category are slightly different. Furthermore, some network tools were added; some were not mentioned (the list of tools given is not exhaustive). The overview of this classification can be seen in the following table 3.5.

NFATs	
Open Source	PyFlag
	Xplico
Propiertary	NetDetector
	NetIntercept
	OmniPeek

NSM Tools	
Intrusion Detection Systems (IDS)	Bro
	Snort
Network Monitoring Tools	IPTraf
	Ntop
	TCPStat
	VisualRoute
Network Scanning Tools	Angry IP Scanner
	Nmap
	Wireless Network Watcher
Network Sniffers and Packet Analyzing Tools	Aircrack-ng
	eMailTrackerPro
	Kismet
	NetworkMiner
	ngrep
	WebScarab
	Wireshark
Vulnerability Assessment Tools	Acunetix WVS
	Metasploit
	Nessus
	Nikto
	Yersinia
	Wikto

Table 3.5: Network forensic tools taxonomy according to "Fundamentals of Network Forensics"

3.5.1 Network Forensic Analysis Tools

NFATs also in this taxonomy represent tools specially designed for network forensic. They are divided into two classic subcategories: open source tools and proprietary tools [189].

The authors bring to the fore the possibility to master and reprogram the tools to become new, more powerful tools as an advantage of the open-source tools. On the other side, the proprietary tools are often connected to the logging appliance that logs the data and can store them for more than a year for future investigation [189].

3.5.2 Network Security and Monitoring Tools

This classification takes into account the functionality of network tools and distinguishes the following five categories:

- Intrusion Detection Systems (IDS),
- Network Monitoring Tools,
- Network Scanning Tools,
- Network Sniffers and Packet Analyzing Tools,
- Vulnerability Assessment Tools.

The IDSs are devices that monitor the network, and the aim is to detect potential unusual traffic. The IDS can be distributed as software or hardware and can be divided into network-based (NIDS), host-based (HIDS), and IDS that are combined with the IPS (IDPS) [189].

Network monitoring tools focus on monitoring the network (performance, QoS, delay, and bandwidth). In general, these tools are a collection of simple network tools including system commands [189].

Network scanning tools provide an automated and efficient way to carry jobs related to network scanning. This includes procedures like ping sweep (find active hosts), port scan (find offered services), and inverse mapping procedure (find IP that belongs to the active host) [189].

The main goal of network sniffers and packet analyzing tools is to intercept and capture the network traffic data. They can also be both software and hardware. The sniffers capture the network data, and the packet analyzing tools analyze these captured data according to the specified standards. In network forensics, these tools are helpful in analyzing network problems, detecting exploitation attempts isolating exploited systems, monitoring system usage, and other similar issues [189].

The last of the five categories is the category for vulnerability assessment tools. These tools aim to scan the system for known vulnerabilities. In some cases, they can also mask a fake attack to find new vulnerabilities [189].

3.6 Taxonomy according to IOSR-JCE

The journal article “NetworkForensic Application in General Cases” was published in the IOSR Journal of Computer Engineering in 2016. This article is focused on the applications for improving the success of network forensic processes. The authors discussed the classification of network forensic tools that are applications used by forensic experts. The

classification divides network forensic tools into two categories: console-based tools and tools with GUI. This basic taxonomy can be seen in table 3.6 [192].

GUI-based tools have a graphical user interface and are usually more user-friendly. The console-based network forensic tools do not have a graphical user interface and are managed through the command line. These console-based tools include common networking tools, such as ifconfig or ping.

Console-based tools
ARP
Gnetcast - GNU
ifconfig
Network Mapper (Nmap)
ping
snoop
TCP dump
Xplico

GUI-based tools
E-detective
Netcat
Wireshark/Ethereal

Table 3.6: Classification of network forensic tools published in IOSR-JCE

3.7 Taxonomy according to ENISA

ENISA describes some well-known tools for network forensic analysis in the document handbook for "Trainings for Cybersecurity Specialists" called "Introduction to Network Forensics" written in 2019. These tools are divided into categories based on the properties of the tools: flow capture & analysis tools, full-state analysis tools, IDS, packet capturing tools, and pattern matching tools. This classification of tools is shown in table 3.7 [187].

The packet capturing tools' main goal is to acquire packets from the network traffic, usually in PCAP format. The pattern matching tools serve mainly for searching for a particular pattern in the captured network traffic data. Since the network flow is useful for creating an overview of the activities on the network, the flow capture and analysis tools are part of one of the categories, and these tools are focused on the network flow. The flow tools obtain the network flow data and analyze them. Another category is network IDS. IDS tools play an essential role in network forensics thanks to monitoring the network and looking for any unusual or malicious traffic. The last but not least category is full-state analysis tools. These tools are more complex and can obtain more information from the network traffic data [187].

flow capture & analysis tools	full-state analysis tools	IDS
Argus	WireShark	Snort

packet capturing tools	pattern matching tools
tcpdump dumpcap	ngrep

Table 3.7: Network forensic tools classification according to ENISA

3.8 Comparison

Each described taxonomy has in common that it classifies network tools with examples of tools. Many taxonomies also have common categories. According to the discussed taxonomies, many emphasize IDS and sniffers as a category of network forensic tools. Classification based on subscription is also prevalent – distinguishing open-source and proprietary tools. Less frequent categories include web forensics, email forensics, console-based, and GUI-based tools.

The taxonomies differ in the number of categories or the depth of the taxonomy. Some of them distinguish one level with a few categories, like in taxonomies according to IJNSA, IOSR-JCE, or ENISA. Simson Garfinkel's taxonomy adds a second level for "catch it as you can" approach with commercial and open-source categories. More complex second levels can be seen in the Pilli and Joshi taxonomies or Davidoff and Ham taxonomy.

3.9 Summary

The previous sections describe the existing network forensic tools taxonomies. It can be seen that it depends on the author's perspective on the network forensic tools. Each classification tries to provide an appropriate description of the tools used for network forensics and classify them into meaningful categories.

The oldest taxonomy for network forensic tools, the classification of Simson Garfinkel, provides only two categories and focuses only on monitoring and recording network data. The "catch it as you can" approach is divided into popular groups: open-source and proprietary tools. The "catch it as you can" approach can be understood as network sniffers.

The taxonomy published by IJNSA in 2009 focuses on three categories: sniffers, email, and web forensics.

Pilli and Joshi discussed network forensic tools in two of their publications. We can find similar features in these classifications. The detailed comparison of these two taxonomies can be seen in the following subsection.

The book "Network Forensics: Tracking hackers through cyberspace", which is a basement of the taxonomy described in section 3.4, provides network forensic tools classification based on the investigation process phases. This is a different approach to taxonomy compared to the other described taxonomies. Although the book is more specialized on the traffic analysis and network forensic process itself, and the tools are described as part of the individual analysis process part, this taxonomy provides deep insight into the use of tools during the network forensic process.

The taxonomy according to the IOSR-JCE journal, describes only two categories based on the approach of working with these tools — console-based and GUI-based. This classification differs a little from the others described in the thesis. Its classification is not based on network and security approaches but on how the investigator interacts with the tools.

The last described taxonomy is the taxonomy according to ENISA, from 2019. There can also be seen the network and security-based classification. Compared to the Pilli and Joshi taxonomies, it can be understood as the classification of NSM Tools. The cons of this taxonomy are that the author provided only a small number of network forensic tools for each category.

3.9.1 Pilli and Joshi taxonomies

Pilli and Joshi published two network forensic tools classifications that are more extensive than the other described taxonomies. Both published in "Digital Investigation" journal and in "Fundamental of Network Forensics" book, distinguish two main categories: NFATs and NSM Tools. The NFATs' subcategories are the same for these two taxonomies, but the NSM Tools categories differ slightly. The same subcategories in the NSM Tools category include only IDS. Some categories in the lately published book can be understood as another named categories that contain tools from previously defined categories or as a merge of previously defined categories in the earlier published journal, following:

- *Network Monitoring Tools* contain *Statistical* tools,
- *Network Scanning Tools* contain *Fingerprinting* tools,
- *Network Sniffers and Packet Analyzing Tools* can be understood as merged *Packet Capture* and *Pattern matching* tools,
- *Vulnerability Assessment Tools* is completely new category containing Nessuss tool that was previously defined between *Packet Capture* tools.

The interesting issue is that the NetworkMiner tool is in the earlier publication classified between NFATs and in the later publication appears between NSM Tools.

Although the authors are the same for these two taxonomies, we can see more different features than in common. The authors also did not provide the same network forensic tools as examples but tried to use the most useful and practical tools that fit the defined categories best. Therefore, these taxonomies have in common only the main structure and in detailed classification differs.

Chapter 4

Network forensics tools

Each previously described taxonomy provides an example of network tools for each defined category. It is useful to have an overview of tools that can be used in network forensics with its basic description. The overview of available tools helps to choose the suitable tool that can assist in obtaining information, collecting and analyzing the evidence, or creating reports.

A simple overview of some network forensic tools can be found on the Forensics Wiki. However, as these web pages were not updated since 2016, there are a lot of outdated information and broken links [29, 31].

This chapter describes individual tools from the previous chapter 3. In addition to the mentioned tools, also other tools are described. The tools are sorted alphabetically.

4.1 Tools from the previous literature survey

This section describes the network tools that were mentioned in the literature survey of network forensic tools in chapter 3.

4.1.1 Acunetix Web Vulnerability Scanner

An Acunetix is a complete web application security testing solution that includes a web vulnerability scanner. It is a commercial tool and the pricing is based on the number of scanned websites. The Acunetix can be used both standalone and as part of complex environments. This product offers built-in vulnerability assessment, vulnerability management, and integration with software development tools [50].

It can be deployed locally on Linux, Mac OS, and Microsoft Windows operating systems. In addition to the locally deployed product, the cloud version is also supported.

This network tool can be used during the Obtain phase of the OSCAR investigative methodology.

The official Acunetix website[1] provides the following advantages of using this product:

- increase the cybersecurity stance and eliminate many security risks at a low resource cost,
- it is one of the best DAST tools,
- efficiency in both physical and virtual environments,

[1] https://www.acunetix.com/

- Acunetix integrations are designed to be easy,
- support of third-party issue trackers including two-way integration,
- it is constantly being improved since 2005,
- it is written in C++, making it one of the fastest web security tools on the market (Acunetix also uses a unique scanning algorithm – SmartScan),
- very high vulnerability discovery effectiveness (very low false-positive rate),
- Acunetix provides proof of exploit for many vulnerabilities,
- you can use multiple scanning engines deployed locally (on-premises or cloud version),
- it can discover security threats listed in OWASP Top 10[2],
- To protect your key assets, you can use the unique AcuSensor IAST technology for PHP, Java, or .NET. This technology helps you remediate by making it easier to pinpoint the cause of the security hole,
- Acunetix is integrated with the OpenVAS open-source tool.

4.1.2 Aircrack-ng

An Aircrack-ng is an open-source that was started in 2006 and is still developing. It is a suite of tools to assess WiFi network security. The Aircrack-ng can run on Windows and Linux machines. It also works on OS X, FreeBSD, OpenBSD, NetBSD, Solaris and eComStation 2 [51].

There are four areas of WiFi security, the aircrack-ng focuses on [51]:

- *Monitoring* — packet capture and data export for further analysis,
- *Attacking* — replay attacks, deauthentication, fake AP and others via packet injection,
- *Testing* — checking WiFi cards and driver capabilities (capture and injection),
- *Cracking* — WEP and WPA PSK.

The Aircrack-ng suite consists of the following tools – airbase-ng, aircrack-ng, airdecap-ng, airdecloak-ng, airdriver-ng, airdrop-ng, aireplay-ng, airgraph-ng, airmon-ng, airodump-ng, airolib-ng, airserv-ng, airtun-ng, besside-ng, dcrack, easside-ng, packetforge-ng, tkiptun-ng, and wesside-ng. The other tools include WZCook, ivstools, Versuck-ng, buddy-ng, makeivs-ng, and kstats [51].

The Aircrack-ng can be used during the Collect and Analyze phases of the network forensic process based on the OSCAR methodology.

[2] https://www.acunetix.com/vulnerability-scanner/owasp-top-10-compliance/

4.1.3 AirPcap/Riverbed AirPcap

An Riverbed AirPcap, formerly AirPcap, is a USB-based adapter that captures 802.11 wireless traffic. The captured data can be analysed by other analysis tools like Wireshark. The only supported platform for AirPcap is Windows [96].

The AirPcap Product Family contains products like AirPcap Classic, AirPcap Tx, and AirPcap Nx. All these products can capture full 802.11 frames, are fully integrated with Wireshark, have an open API, support multi-channel monitoring (with two or more adapters), and have USB dongle form. Packet transmission is available only on AirPcap Tx and AirPcap Nx. Frequency bands for AirPcap Classic and AirPcap Tx are 2.4 GHz (b/g), for AirPcap Nx 2.4 and 5 Ghz (a/b/g/n) [16].

The Airpcap can be used during the Collect phase of the network forensic process based on the OSCAR methodology.

4.1.4 Angry IP Scanner

An Angry IP Scanner, also known as ipscan, is a free and open-source network scanner. The aim of this tool is to scan IP addresses and ports, the results can be saved in many supported formats including CSV, TXT, XML, or IP-Port list. It also supports many plugins that can provide the user with detailed information about scanned nodes like hostname, MAC address, or NetBIOS information. Other features include favorite IP address ranges, web server detection, customizable openers, and so on [53].

This network tool can be run on many platforms including Linux, Windows, and Mac OS X.

The advantages of the Angry IP Scanner include user-friendly interface, it is a very fast IP address and port scanner, it does not require any installation, and it uses a multithreaded approach for increasing the scanning speed [53].

There are also described some usage scenarios in the product documentation [52].

This network tool can be used during the Obtain phase of the OSCAR investigative methodology, for example, when gathering information about the network topology.

4.1.5 Argus

An Argus is a network flow system, developed by Carter Bullard in the early 1980's at Georgia Tech. The Argus Project is an open source project focused on proof of concept demonstrations of all aspects of large scale network awareness derived from network flow data. It is a real-time flow monitor that is designed to perform comprehensive data network traffic auditing [91].

The Argus's main goal is to process captured packets or on the wire into the network flow data. It deals with the following issues of network flow data: scale, performance, applicability, privacy, and utility [91].

The Argus system consists of two parts:

1. *argus* — a packet processing network flow sensor that generates Argus data,
2. *argus-clients* — a collection of argus data processing programs.

The Argus Project efforts include: data generation, transport, collection, storage, analytics, and various metadata enhancements [91].

The Argus is multi-platformed tool, and it supports more than 24 platforms. The argus-clients focuses on data processing including data distribution, collection, filtering, aggregation, binning, minimization, privacy, metadata enhancement, geolocation, net-spatial location, compression, anonymization, graphing, databases, analytics, storage, and error correction [91].

Considering this tool as a tool that works with the network flow data (captures and analyzes them), it can be used during the Collect and Analyze phases of the OSCAR process model.

4.1.6 ARP

An arp is a command-line network tool that is used to display and modify the ARP cache. This command is available for many platforms, including Linux, Windows, and MacOS systems. The main features include displaying ARP cache for a single interface or all interfaces, deleting and adding an address mappings [32, 54].

This command can be used during the Obtain and Analysis phases of the OSCAR process model.

4.1.7 Bless

A Bless is an open-source binary HEX editor [143].

The main target platform for this tool is GNU/Linux. However, since all used libraries are cross-platform, the Bless is able to run also on other platforms like BSD, Solaris, and Win32 [143].

Some of the main features include:

- efficient editing of large data files,
- raw disk editing,
- multilevel undo - redo operations, fast find and replace operations, multi-threaded search and save operations,
- conversion table,
- export to text and html (others with plugins),
- extensibility with plugins.

Since this network forensic tool is a HEX editor, it can be used mainly during the Analyze phase of the OSCAR process model.

4.1.8 Bro/Zeek

Zeek, formerly Bro, is a network security monitoring tool. It is a passive, open-source network traffic analyzer. This tool can be installed in the Unix and MacOS systems [107].

In addition to acting as a security monitor that inspects all traffic data and searches for abnormal activities, this tool also provides a wide range of traffic analysis tasks, including performance measurements. The Zeek also provides a management framework, named ZeekControl [49].

Zeek's architecture consists of the event engine and the policy script interpreter. The event engine receives the packets from the network and produces events read by the policy

script interpreter. The policy script interpreter then generates logs and other notifications. The script interpreter executes a set of event handlers written in Zeek's custom scripting language [49].

The user manual of the tool provides also examples and use cases of common using the Zeek tool [49].

Some of the features Zeek provides include:

- real-time and offline analysis,
- cluster support,
- support for many application-layer protocols and analysis of file content exchanged over application-layer protocols,
- tunnel detection and analysis,
- support for IDS-style pattern matching,
- event-based programming model,
- alternative backends for ElasticSearch and DataSeries [49].

The Zeek can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.9 Carnivore

A Carnivore, also known as DCS1000, is an FBI software-based tool used to examine all IP packets on an Ethernet and record only those packets or packet segments that meet very specific parameters. Therefore, this tool can be classified as a packet sniffer [164, 22].

This tool can be installed by properly authorized FBI agents on a particular Internet Service Provider's (ISP) network. This software system is used together with a tap on the ISP's network. The aim is to intercept, filter, seize, and decipher digital communications on the Internet [195].

Considering this tool as a packet sniffer, it can assist in the Collect phase of the OSCAR process model.

4.1.10 Check Point IPS-1/IPS

A Check Point IPS-1 is an IPS that uses IPS-1 Sensors that can be placed on the network perimeter or at any location of the internal network [9].

The advantages of IPS-1 may include:

- Unified security management,
- Mission-critical protection against known and unknown attacks,
- Granular forensic analysis,
- Flexible deployment,
- Confidence Indexing.

The Check Point IPS is available in two deployment methods: IPS Software Blade, and IPS-1 Sensor [24].

The Check Point IPS is part of the Check Point Next Generation Firewall. It is a commercial IPS that detects or prevents attempts to exploit weaknesses in vulnerable systems or applications [69].

The Check Point IPS can assist during the Collect and Analyze phases of the OSCAR process model.

4.1.11 Cisco FireSIGHT System

A Cisco FireSIGHT System, formerly SourceFire 3D System, is an integrated suite of network security and traffic management products. The appliances can be used in switched, routed, or hybrid environments. The products can be deployed either as software-based appliances or on purpose-built platforms. It is possible also to configure NAT, establish VPN tunnels between endpoints, configure bypass interfaces, aggregated interfaces, fast-path rules, and strict TCP enforcement [23, 39].

FireSIGHT components include Redundancy and Resource Sharing, Network Traffic Management, FireSIGHT, Access Control, SSL Inspection, Intrusion Detection and Prevention, Advanced Malware Protection and File Control, and Application Programming Interfaces [39].

Some of the managed devices are Series 2 and Series 3 Managed Devices (Cisco FirePOWER 7000 Series and 8000 Series devices), 64-Bit Virtual Managed Devices, Cisco NGIPS for Blue Coat X-Series, and Cisco ASA FirePOWER Devices [39].

The Cisco FireSIGHT System can assist in the Obtain, Collect, and Analyze phases of the OSCAR process model.

4.1.12 Corero Network Security

The Corero provides network security products focused on DDoS protection. The company's products include:

- *SmartWall DDoS Protection* — a real-time, automatic, highly acalable DDoS protection solution with multiple deployment options (on-premise, cloud, hybrid),
- *SecureWatch Managed DDoS Protection Services* — a SmartWall deployment managed by Corero experts [58].

This network security tool can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.13 DeepSee

A DeepSee is a proprietary network security tool developed by Solera Networks, acquired by NortonLifeLock.

Solera DS 5150 was an appliance for high-speed data capture, complete indexed record of network traffic, filtering, regeneration, and playback [198].

According to Symantec's *Security Analytics Hardware EOL (End of Life)*, all DeepSee hardware already reached EOL [41].

4.1.14 dig

Dig is a BIND's command line DNS diagnostic tool. BIND 9 is an open-source full-featured DNS system. It is available for Windows and Linux platforms [55].

This tool can be helpful during the Analysis phase of the OSCAR investigative methodology.

4.1.15 dumpcap

A Dumpcap is a part of the Wireshark distribution. It is a network traffic dump tool that captures packet data and stores them according to the specified parameters [60].

This tool can be used during the Collect phase of the OSCAR process model.

4.1.16 E-detective

An E-detective is a real-time LAN Internet monitoring tool developed by the Decision Group. This tool's function is to capture and decode network packets, and it reconstructs them and saves them in the original format. Thanks to the reconstruction of the data and saving it in the original format, the E-detective user can see the data in the same way seen on the network. Despite using this tool during forensic analysis and investigation, it can also be used in auditing, record keeping, legal and lawful interception, and others [59].

It can be deployed as a temporary deployment (a tactical standalone system) or permanently deployment (Private Enterprises) [59].

The range of the protocol the E-detective can decode and reconstruct is more than 140 different protocols, including

- email and webmail protocols (POP3, SNMP, IMAP, Yahoo Mail, Windows Live Hotmail, Gmail),
- Instant Messaging protocols (Yahoo, MSN, ICQ, QQ, Google Talk, IRC, UT Chat Room, Skype),
- File Transfer protocols (FTP, P2P),
- Social media sites (Facebook, Twitter),
- Telnet, Online games, HTTP, VOIP, mobile service protocols, and more.

The advantages include that this tool can also work with the HTTPS traffic when the HTTPS module is enabled. When using the HTTPS decoder, the user's login and password information are captured. Furthermore, the data can be archived using the automated FTP service, and also they can be downloaded as an ISO file. Moreover, this tool can work with other reporting tools, and therefore can provide users with professional reports like reports with Up-Down View, Total Throughput Statistical Report, Network Service Report (Daily, Weekly basis), Top Websites, and others. Additionally, there are also available search functions like Free Text Search, Conditional Search, Similar Search, and Association with Relationship Search. Alert and notification functions are also provided. Some other functions include Bookmark, Capture File List (Comparing the content of two files), Online IP List, Authority Assignment, Syslog Server, hashed export (backup), and file content comparison [59].

The E-detective can be considered a complex forensic network tool, and therefore it can be used during the whole investigation process. This tool can capture the data, analyze them, and provide the report.

4.1.17 EmailTrackerPro

An EmailTrackerPro is an email tracker and spam filter tool. It is a commercial tool available for the Windows platform [19].

The main functions of this tool are

- *tacking the email* — providing the location of the email (usually displayed on the world map), the tracing of the email is done using the email header information,
- *report abuse* — ,a more proactive approach to dealing with spam, EmailTrackerPro provides a platform that auto-generates an abuse report,
- *spam filter* — stopping spam before it reaches the inbox.

This tool analyzes the email traffic, and therefore, can be used during the Analysis phase of the OSCAR process model.

4.1.18 Enterasys IPS

The Enterasys was acquired by "Extreme Networks" on September 2013 [17].

The IPS system of "Extreme Networks" company, Extreme Networks Intrusion Prevention System (IPS), is an IPS system that is able to gather evidence of an attacker's activity, remove the attacker's access to the network, and reconfigure the network to resist the attacker's penetration technique [119].

This IPS system can assist during the Collect and Analyze phases of the OSCAR process model.

4.1.19 EtherApe

An EtherApe is an open-source graphical network monitoring tool. This tool is available for UNIX systems. It can work with the live data or with a tcpdump captured file. The aim is to display the network traffic graphically - node and link color show the most used protocol. There can be displayed traffic within their network, end to end IP, or port to port TCP [62].

Moreover, there is possible to select the level of the protocol stack to concentrate on. The displayed data can be refined using a network filter. Furthermore, there is able to display averaging and node persistence times. This tool also provides TCP statistics and node statistics that can also be exported [62].

The EtherApe can be used during the Collect, Analyze, and Report phase of the OSCAR process model.

4.1.20 Ethereal/Wireshark

A WireShark, formerly Ethereal, is an open-source network protocol analyzer. It provides users with what is happening on the network at a microscopic level. It can work with live captured data or already captured data in many supported captured formats [115].

Other features of Wireshark include filtering packets according to the specified filters, searching for packets on many criteria, saving and exporting captured traffic data, and creating statistics [116].

It is a multiplatform tool that can be run on Windows, MacOS, UNIX, Linux, BSD, Solaris, and many other systems [116].

The Wireshark can be classified as a complex network forensic tool that can help in the Collect and especially Analyze phase of the OSCAR investigation topology. This tool can also be useful for creating statistics during the Report phase of the OSCAR model.

4.1.21 findsmtpinfo.py

A findsmtpinfo.py is a network tool written for the Network Forensic Puzzle #2 Contest by Jeremy Rossi. This tool reads a PCAP file, decodes authentication data (username and password), gathers email information, stores attachments (decompresses them if in compressed format), checks the MD5sum, and creates a report of the SMTP information [162].

This smtp tool can be used during the Analyze phase of the OSCAR process model.

4.1.22 flow-tools

A flow-tools is a set of network tools for working with NetFlow data. The function is to collect, send, process, and generate reports from NetFlow data [63].

It can be deployed as a package for Linux systems. It can be used on a single server or distributed to multiple servers [63].

The flow-tools distribution includes the following tools:

- *flow-capture* — collect, compress, store, and manage disk space for exported flows,
- *flow-cat* — concatenate flow files,
- *flow-dscan* — tool for detecting some types of network scanning and DoS attacks,
- *flow-expire* — expire flows,
- *flow-export* — export data,
- *flow-fanout* — replicate NetFlow datagrams to unicast or multicast destinations,
- *flow-filter* — filter flows and can be used with other programs to generate reports,
- *flow-gen* — generate test data,
- *flow-header* — display meta information in flow file,
- *flow-import* — import data,
- *flow-merge* — merge flow files (in chronological order),
- *flow-receive* — receive exports using the NetFlow protocol (without storing to disk),
- *flow-report* — generate reports for NetFlow data sets,
- *flow-send* — send data using the NetFlow protocol,
- *flow-split* — split flow files,

- *flow-tag* — tag flows for and can be used to group flows and generate reports,
- *flow-xlate* — perform translation on some flow fields.

This set of network tools can be used during the Collect, Analyze and Report phases of OSCAR model.

4.1.23 FlowTraq

A FlowTraq is a commercial flow record analysis tool developed by ProQueSys, lately acquired by Riverbed. It offers two deployment models: cloud and on-premise solution [64].

This tool can recognize DDoS and other attacks in real time and trigger automated scrubbing, protect sensitive information, defend the network from malicious botnets, and improve the network forensics capabilities [64].

The FlowTraq supports all flow formats that can be mixed as sources. It can sniff traffic directly, generate flow records, filter, search, sort, and produce reports [64].

The FlowTraq also provides a tool named the FlowTraq Exporter. The FlowTraq Exporter is a free software flow exporter that exports existing PCAP traffic data files into flow format data. It can be run on Windows, Linux, and BSD systems [65].

As other network forensic tools that work with the flow data, also the FlowTraq can be used during the Collect, Analyze and Report phases of OSCAR model.

4.1.24 Gnetcast (GNU Netcat)

A Gnetcast is a GNU rewrite of netcat tool (the netcat tool is described in section 4.1.38) [192].

It is fully compatible with the netcat tool and portable. The supported platforms include Linux, FreeBSD, NetBSD, Solaris, and MacOS [5].

Like the netcat tool, the GNU Netcat can be used during the Obtain, Collect, and Analyze phases of the OSCAR process model.

4.1.25 HP TrippingPoint IPS

HP TrippingPoint company was acquired by "Trend Micro" on October 2015 [153].

The Trend Micro Intrusion prevention consists of TippingPoint solutions: TippingPoint Threat Protection System, Centralized Management and Response, and Threat Intelligence. Trend Micro TippingPoint Threat Protection System Family provides real-time detection, enforcement, and remediation without compromising security or performance. Key features include Cloud Network Protection, On-Box SSL Inspection, Performance Scalability, Flexible Licensing Mode, Real-Time Machine Learning, Enterprise Vulnerability Remediation (eVR), Advanced Threat Analysis, High Availability, Integrated Advanced Threat Prevention, Asymmetric Traffic Inspection, Agility and Flexibility, Best-in-Class Threat Intelligence, Virtual Patching, Support for a Broad Set of Traffic Types, and Centralized Management [61].

The IPS system can assist during the Collect and Analyze phases of the OSCAR process model.

4.1.26 IBM Security NIPS

An IBM Security Network Intrusion Prevention System (NIPS) is an IPS system developed by the IBM company. The IBM Security NIPS appliances are purpose-built, Layer 2 net-

work security appliances. The aim is to block intrusion attempts, denial of service (DoS) attacks, malicious code, backdoors, spyware, and peer-to-peer applications [121].

This IPS system can assist during the Collect and Analyze phases of the OSCAR process model.

4.1.27 ifconfig/ipconfig

An ifconfig is a Linux command line network tool used for configuring a network interface. This command can be replaced by commands *ip addr* and *ip link*. Without specifying arguments, ifconfig displays the status of the currently active interfaces [68].

The Windows alternative is command named ipconfig [33].

This tool can be useful during the Obtain and Analysis phase of the OSCAR process model.

4.1.28 Index.dat analyzer

An Index.dat is a free network tool used for viewing, examining, and deleting the contents of index.dat files. Index.dat files contain all online activity information, like searching history, visited websites, and accessed URLs, files, and documents [4].

This tool can be used during the Analyze OSCAR phase.

4.1.29 InfiniStream (nGeniusONE)

An InfiniStream is an intelligent deep packet capture and analysis appliance that is the foundation for a nGeniusONE platform. This is a proprietary tool for customized Linux systems, owned by the NetScout Systems company [83, 26].

The features of the InfiniStream appliances include real-time packet flow-based data monitoring, packet storage for forensics (back-in-time analysis), passive and non-intrusive capturing all network traffic and generating metrics, ASI technology (for high performance, deep packet inspection, and analysis), scalable architecture, working also with packet crossing the wire, and flexible range of appliances [26].

The InfiniStream is a powerful tool that can assist in the Collect and Analyze phases of the OSCAR investigative methodology.

4.1.30 IPFIX

An IPFIX (Internet Protocol Flow Information Export) is a protocol that transmits traffic flow information over the network. The architecture contains a collector (Collecting Process) and an exporter (Exporting Process). It defines how IP flow information is to be formatted and transferred from an exporter to a collector. It supports TCP, UDP, and SCTP as transport protocols. The IPFIX provides the following three record formats: the Template Record format, the Options Template Record format, and the Data Record format [138].

The IPFIX can be useful during the Collect phase of the OSCAR process model.

4.1.31 IPTraf

An IPTraf is an open-source network monitoring utility for IP networks for Linux systems. It intercepts packets on the network and analyzes the IP traffic [1].

The provided information about the IP traffic include:

- Total, IP, TCP, UDP, ICMP, and non-IP byte counts,
- TCP/UDP/OSPF source and destination information (addresses and ports),
- TCP packet counts, byte counts, flag statuses,
- ICMP type information,
- TCP/UDP service statistics, LAN station statistics,
- Interface packet counts, IP checksum error counts, and activity indicators

The IPTraf tool can be used during the Collect and Analyze phase of the OSCAR model.

4.1.32 Iris

An Iris, formerly SpyNet CaptureNet, is a network traffic analyzer developed by eEye Digital Security, acquired by Beyond Trust. It is a commercial tool for the Windows platform that analyzes all the traffic of a network according to filters. It can capture, analyze and show the network data [70].

This tool is designed to take the guesswork out of bandwidth monitoring. It can scan the network packets for searching certain words or monitor specific IP addresses or users. When it detects packets that meet the criteria, it can reconstruct the website or notify when the specific program was used. The Iris is also able to read sent emails, including its attachments [199].

The Iris can be applied in the Collect and Analyze phase of the OSCAR process mode.

4.1.33 KisMAC

A KisMAC is an open-source WiFi scanner that identifies WiFi networks around the device, including hidden, cloaked, and closed ones. This tool is available for MacOS systems [67].

The other features of the KisMAC include:

- information about the logged users on the network (MAC Address, IP address, signal strength),
- supports mapping. GPS, 802.11b/g frequency, and Kismet drone captures,
- PCAP import and export,
- Different attacks against encrypted networks, Deauthentication attacks.

The KisMAC is no longer being updated or maintained. The latest version was released in 2011 [67].

The KisMAC is a scanning network tool, and it can be used during the Obtain, Collect and Analyze phases of the OSCAR process model.

4.1.34 Kismet

A Kismet is an open-source wireless network and device detector, sniffer, wardriving tool, and WIDS framework. It contains Python plugins like kismetdb database module, kismetrest module, or kismetexternal module [71].

The Kismet is a multiplatformed tool that can be run on Linux, Windows, and MacOS systems. It works with Wi-Fi interfaces, Bluetooth interfaces, and some hardware interfaces like SDR hardware [71].

This tool can be used during the Obtain, Collect and Analyze phases of the OSCAR process model.

4.1.35 Metasploit

A Metasploit is a penetration testing framework that is available as open-source Metasploit Framework and commercial Metaspolit Pro. It is available for MacOS, Windows and Linux systems [72].

The Metasploit Framework is a Ruby-based, modular penetration testing platform that provides a complete penetration testing environment and exploits development. It is a collection of commonly used tools, and it is used to write, test, and execute exploit code. The Metasploit Framework tools can be used to test security vulnerabilities, enumerate networks, execute attacks, and evade detection. It provides a command line interface named MSFconsole, for working with this framework. This framework allows manual exploitation and credentials brute-forcing [73].

The Metasploit Pro is an exploitation and vulnerability validation tool. This tool can divide the penetration testing workflow into manageable sections. The typical workflow steps are as following: create a project, get target data, view and manage host data, run a vulnerability scan, set up a listener, exploit known vulnerabilities, post-exploitation and collect evidence, clean up sessions, generate a report. Unlike the Metasploit Framework, the Metasploit Pro provides a web interface, automated exploitation and credentials brute-forcing, baseline penetration testing reports, wizards for standard baseline audits, task chains for automated custom workflows, closed-Loop vulnerability validation to prioritize remediation, web app testing for OWASP Top 10 vulnerabilities, network discovery, integrations via Remote API, and more [73].

These network tools can be used during the Obtain phase of the OSCAR process model to identify the system's network environment and vulnerabilities. Since the Metasploit Pro is a more complex tool and provides collecting evidence and reports, it can also be useful during other OSCAR phases like Collect, Analyze, and Report.

4.1.36 nbtstat

A nbtstat is a Windows command-line diagnostic tool that displays NBT (NetBIOS over TCP/IP) statistics. The other features include displaying NetBIOS name tables and the NetBIOS name cache, and refreshing of the NetBIOS name cache and names [34].

This tool can assist during the Analyze phase of the OSCAR process model.

4.1.37 Nessus

A Nessus is a vulnerability assessment tool. It is a multiplatform tool that can be run on Windows, Linux, and MacOC operating systems [74].

It can be deployed as:

- *Nessus Essentials* — free version of the Nessus tool,
- *Nessus Proffesional* — commercial version of the Nessus tool,

- *tenable.io* — commercial cloud tool with unlimited Nessus Scanners.

The commercial version of Nessus includes, except high-speed, in-depth assessments, unlimited and configuration assessment, live results, and configurable reports [74].

The advantages of Nessus include:

- the industry's lowest false positive rate with six-sigma accuracy,
- the deepest and broadest vulnerability coverage in the industry,
- the #1 deployed solution for application vulnerability assessment.

The Nessus can be used during the Obtain phase of the OSCAR process model. The commercial version can be helpful also during the Report phase.

4.1.38 Netcat

A Netcat, also known as nc, is a utility that opens TCP or UDP connections and reads and writes the data. It supports inbound and outbound connections to or from any port. It is designed as a Linux backend tool. This utility can be considered a powerful debugging and exploration tool since it can provide any network connection [75].

The other features of this tool include full DNS forward/reverse checking, port-scanning, loose source-routing, slow-send mode, hex dump of transmitted and received data, and telnet-options responder [75].

There are many tools similar to the Netcat, such as Ncat, Socat, OpenBSD's nc, Cryptcat, Netcat6, pnetcat, SBD, and GNU Netcat [76].

The Netcat can assist in the Obtain and Analyze phases of the OSCAR investigative methodology. It may also be helpful during the Collect phase.

4.1.39 NetDetector

A Niksun NetDetector is a packet capture and network security forensic tool. It is a proprietary network forensic tool, available in 4 iterations [84, 131].

The NetDetector's variations are:

- *NetDetector* — full packet capture, application fingerprinting/reconstruction, IDS and anomaly detection,
- *NetDetectorLive* — NetDetector with real-time reconstruction, indexing and content alarming,
- *Virtual NetDetector/NetDetectorLive* — cloud version of NetDetector/NetDetectorLive,
- *IntelliDefend* — NetDetector in a lightweight, notebook-sized device.

The NetDetector is a full-featured appliance for network security monitoring. It is used to capture and analyze packets. It is also possible to import and export data in many formats. It provides ad-hoc and scheduled reporting on multiple timescales [27].

The most important features of NetDetector include: dynamic application recognition and plug-ins, integrated anomaly and signature-based IDS, application and session reconstruction, and 100Gbps packet capture and analysis [84].

This network tool can be used during the Collect, Analyze and Report phase of the OSCAR process model.

4.1.40 NetFlow

NetFlow is a protocol developed by Cisco. It is a part of the Cisco IOS Software. It provides information about who, what, when, where, and how network traffic is flowing. It exports the data to NetFlow collectors that create reports [15].

It provides information about network users and applications, peak usage times, and traffic routing. The latest version, NetFlow v9, is the basis of a new IETF standard, and it is a flexible and extensible method to record network performance data [57].

The NetFlow can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.41 NetIntercept

A NetIntercept is a network tool with a focus on data flows. It is similar to the open-source tool tcpflow. The NetIntercept is a commercial program created by Sandstorm Enterprises, acquired by NIKSUN company [144].

The NetIntercept is an IDS/IPS with forensics capability. It can detect and block attacks, restrict traffic by IP and port, utilize full packet-based evidence and deep packet inspection with intelligent threat response, and search the records of blocked traffic. This tool provides a web-based interface [35].

This tool can assist in the Collect and Analyze phases of the OSCAR investigative methodology.

4.1.42 netstat

A netstat is a command line network tool available for Windows, Linux, and other Unix systems. It provides network statistics including network connections, routing tables, interface statistics, masquerade connections, and multicast memberships [36, 79].

4.1.43 NetStumbler

A NetStumbler is a network tool for detecting Wireless Local Area Networks (WLANs) using 802.11b, 802.11a, and 802.11g. The NetStumbler is developed for Windows systems. For Windows CE, there is available MiniStumbler [12].

The main features include: detecting networks that may cause interference, detecting unauthorized "rogue" APs, and finding locations with poor coverage [12].

The NetStumbler can be used during the Obtain OSCAR phase.

4.1.44 NetVCR

A NetVCR is a part of the NetVCR Suite. It is a commercial network tool used to capture packets [85].

The NetVCR Suite includes:

- *NetVCR* — full packet capture with stream-to-disk recording, real-time indexing and application analytics,
- *Virtual NetVCR* — cloud version of NetVCR,
- *IntelliNetVCR* — NetVCR in a lightweight, notebook-sized device,
- *NetVoice* — analyzing Voice-over-IP traffic,

- *NetTradeWatch* — analyzing financial transactions and associated market data feeds,
- *NetBlackBox Pro* — NetVCR-like full packet capture and archiving without the extensive metadata warehouse, providing a cost-effective, flexible solution for performance analysis and forensics.

Some benefits of the NetVCR include: proactive alerting, QoS management and reporting, diagnostics and troubleshooting, accounting, performance analysis, and application/services monitoring [80].

Since the NetVCR works as a packet sniffer, it can be used during the Collect phase of the OSCAR investigative methodology.

4.1.45 NetWitness

A NetWitness was acquired by EMC company, and NetWitness products were integrated into EMC's RSA Security unit [141].

The RSA NetWitness Platform combines SIEM and threat defense solutions. It is a commercial network tool that collects and analyzes data across many capture points, computes on physical, virtual, or cloud platforms, and enriches this data with threat intelligence and business context [97].

This tool can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.46 Network Flight Recorder (NFR)

A Network Flight Recorder is a lightweight application for processing network traffic. It uses the AlphaSOC Analytics Engine. It is an open-source tool and can be run on Linux systems, it can also be run as a service in Windows systems using NSSM[3] [81].

The NFR can monitor and actively read log files from disk (log files by other applications like Bro/Zeek IDS, Microsoft DNS, Suricata DNS) or process events directly from the network (capture traffic data as a network sniffer). In addition to capturing packets, it also provides in-depth analysis and alerting of suspicious events, including identifying gaps in security controls, highlighting targeted attacks, and policy violations [81].

The data can be exported in JSON or CEF format or sent via Syslog. NFR provides a command-line interface with a few predefined commands [81].

The NFR is a complex monitoring tool that can be used during the Collect and Analyze phase of the OSCAR process model.

4.1.47 NetworkMiner

A NetworkMiner is an open-source network security tool. It can be run on Linux, Windows, FreeBSD, and MacOS systems. It can be deployed as a free edition NetworkMiner, or commercial edition NetworkMiner Professional [82].

The NetworkMiner can be run as a passive network sniffer, or it can parse and analyze already captured network traffic data in PCAP files. It can identify the involved hosts, including detailed information like IP, MAC, Operating system, Sent/Received bytes, Open ports, Incoming/Outcoming traffic. It can also obtain files, images, and messages from the traffic. Moreover, it can parse credentials, sessions, DNS, parameters, keywords, and anomalies. Moreover, it provides filtering according to the selected criteria [82].

[3] http://nssm.cc/

The NetworkMiner is a complex network forensic analysis tool and therefore can assist in the Collect and Analyze phase of the OSCAR investigative methodology.

4.1.48 NfDump

A NfDump is an open-source toolset used for collecting and processing the network flow data (netflow v1, v5/v7, v9, IPFIX, and SFLOW). This tool provides the command line interface. The NfDump is used as a backend toolset for NfSen [149].

The NfDump contains the following tools:

- *nfcapd* — netflow collector daemon,
- *nfdump* — process collected netflow records,
- *nfanon* — anonymize netflow records,
- *nfexpire* — expire old netflow data,
- *nfreplay* — netflow replay,
- *nfpcapd* — pcap to netflow collector daemon,
- *sfcapd* — sflow collector daemon,
- *nfprofile* — netflow profiler for NfSen (reads data from nfcapd),
- *nftrack* — port tracking decoder for NfSen plugin PortTracker,
- *ft2nfdump* — flow-tools flow converter into nfdump format,
- *nfreader* — framework for reading nfdump files,
- *parse_csv.pl* — Perl reader that reads nfdump csv output [149].

This toolset can be used during the Collect and Analyze phases of the OSCAR investigative methodology.

4.1.49 NfSen

A NfSen is an open-source web-based frontend of NfDump toolset. It provides the graphical interface. This tool is hosted by Sourceforge [13].

The main features of this tool include displaying the netflow data, navigating through the netflow data, processing netflow data within a specified time range, creating history, setting alerts, and writing own plugins [13].

Since the NfSen is based on the NfDump toolset, it can be used during the Collect and Analyze phases of the OSCAR process model.

4.1.50 Ngrep

A Ngrep is an open-source network tool that can be run on multiple platforms, including Linux, Windows, and MacOS. This tool searches and filters the network packets according to the specified patterns. It can work as a sniffer and monitor the network interfaces or read the packet data from the network capture file [160].

The features of ngrep include debugging plaintext protocol interactions, identifying and analyzing anomalous network communications, and storing, reading, and reprocessing pcap dump files while looking for specific data patterns [160].

The Ngrep can assist in the Analyze phase of the OSCAR process model. It can also be useful during the Collect phase since this tool can also work as a packet sniffer.

4.1.51 Nikto

A Nikto is an open-source web server scanner for Linux systems [86, 166].

It performs comprehensive tests against web servers for multiple items, including potentially dangerous files/programs, checks for outdated versions, version specific problems, and checks server configuration items [86].

The major features of Nikto include: SSL support, full HTTP proxy support, identifying installed software, replaying saved positive requests, enhanced false positive reduction, guessing credentials for authorization realms, scanning tuning to include or exclude entire classes of vulnerability checks, username enumeration, template engine to easily customize reports, and more [86].

This tool can be used during the Obtain and Analyze phases of the OSCAR model.

4.1.52 Nmap

A Nmap is an open-source network mapper used for network discovery and security auditing that works on a single host or large networks. This tool can also be useful during network inventory, managing service upgrade schedules, and monitoring host or service uptime. It can provide information about the available hosts and services, operating system information (such as a version of the operating system), type of packet filters/firewalls, and more [87].

In addition to the command-line interface, this tool also provides a graphical user interface called Zenmap. This network tool can be run on multiple platforms, including Windows, Linux, and MacOS [87].

The other tools that are part of the Nmap include:

- *Ncat* — a flexible data transfer, redirection, and debugging tool,
- *Ndiff* — a utility for comparing scan results,
- *Nping* — a packet generation and response analysis tool [87].

Since the Nmap provides network discovery, it can assist in the Obtain phase of the OSCAR process model.

4.1.53 nslookup

A nslookup is command-line network tool that queries Internet domain name servers. This tool is used for diagnosing DNS infrastructure. It can work in interactive or non-interactive mode. The interactive mode is used to look up more than one piece of data. The non-interactive mode is recommended for looking up only a single piece of data. It is available in multiple operating systems, including Windows, Linux, and MacOS [37, 88].

This tool can assist in the Obtain and Analyze phases of the OSCAR process model.

4.1.54 Ntop

A Ntop is an open-source network traffic monitoring software that consists of many tools that can capture packets and record and analyze traffic [89].

The Ntop contains the following tools:

- *ntopng* — web-based traffic analyzer and flow collector,
- *nDPI* — Deep Packet Inspection framework,
- *nProbe* — NetFlow v5/v9/IPFIX probe with plugins support for L7,
- *PF_RING* — packet capture,
- *n2disk* — network traffic recorder,
- *disk2n* — network traffic replayer.

The Ntop can assist in Collect and Analyze phases of the OSCAR process model.

4.1.55 oftcat

An oftcat is a simple Perl script that parses OFT (Oscar File Transfer) packages and saves the gained info into a specified file [146].

The oftcat can be useful during the Analyze phase of the OSCAR process model.

4.1.56 OmniPeek

An OmniPeek is a commercial network protocol analyzer with graphical user interface. It provides real-time deep packet analysis including layer 7 traffic. The other features include analyzing traffic from any remote network segment, monitoring voice and video over IP traffic in real time, capturing and analyzing 802.11n and 802.11ac wireless traffic from already deployed access points, integrated flow and packet-level analysis, expert analysis, and automatic alerts [90].

The OmniPeek can assist in the Analyze phase of the OSCAR process model.

4.1.57 P0f

A P0f is a free network tool that identifies the players behind TCP/IP communications without interfering. It uses passive traffic fingerprinting mechanisms [169].

Other capabilities include highly scalable and fast identification of the operating system and software, automated detection of connection sharing/NAT, load balancing, and application-level proxying setups. Moreover, it can measure the system uptime and network hookup, distance (including NAT or packet filters), and user language preferences [169].

This tool's typical uses include reconnaissance, network monitoring, detection of unauthorized network interconnects, and forensics [169].

The P0f can assist in Obtain and Analyze phases of the OSCAR process model.

4.1.58 PADS

A PADS, Passive Asset Detection System, is a free portable lightweight signature-based detection engine that passively detects network assets. It listens to network traffic, attempts to identify the applications running on the network, and creates reports in the CSV format [2].

The PADS can be used during the Collect and Analyze phases of the OSCAR process model.

4.1.59 pcapcat

A pcapcat is a simple Perl script that reads PCAP files and prints information about the connections. This tool also provides the ability to filter the data using traditional pcap filters and stores the gained information into a file [147].

The pcapcat can be used during the Analyze phase of the OSCAR process model.

4.1.60 ping

A Ping is a command line network tool used for verifying IP-level connectivity to another TCP/IP computer. Is sends ICMP Echo request messages to network hosts. This command is available on multiple operating systems, including Windows, Linux, and MacOS [40, 93].

This command can be used during the Obtain phase of the OSCAR process model. In some cases, it can also assist in the Analyze phase.

4.1.61 PyFlag

A PyFlag, Python implementation of FLAG (Forensic and Log Analysis GUI), is an open-source advanced forensic tool for analyzing log files and forensic investigation. This tool can process large PCAP files, analyze and extract the content of the communication. The PyFlag is able to recursively examine data at multiple levels and discover files encapsulated within other files. It also provides advanced reconstruction of web pages, and specific analysis for popular webmail sites [140, 94, 18].

The architecture consists of the following components: IO Source, File System Loader, VFS, Scanners, Database, and Web GUI [140].

The PyFlag is marked as deprecated and is no longer maintained according to the Forensics Wiki [95].

The PyFlag can be a helpful tool during the Analyze phase of the OSCAR process model.

4.1.62 Sebek

A Sebek is an open-source data capture tool that works on the kernel level. It uses techniques similar to those used by rootkits. The aim is to capture an attacker's activities (keystrokes, file uploads, passwords) on a honeypot, without the attacker knowing it. This tool is available for Linux (2.4 and 2.6 kernels) and Windows systems [48, 6, 139].

This capture tool consists of two components:

- *a client* — runs on the honeypots, captures activities, and sends the data to the server,
- *the server* — runs on the Honeywall gateway (or independently), collects the data from the honeypots [48].

The Sebek can assist in the Collect phase of the OSCAR process model.

4.1.63 sFlow

A sFlow is a technology for monitoring network traffic data. The architecture consists of agents and collectors. It is an industry standard that provides a network-wide view of usage and active routes (measuring network traffic, collecting, storing, and analyzing traffic data) [98].

The sFlow can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.64 SiLK

A SiLK is a set of traffic analysis tools. Its components are open-source. It is designed to analyze traffic. The supported platforms include Linux, Solaris, OpenBSD, Mac OS X, and Cygwin [99].

The SiLK tool suite supports the efficient collection, storage, and analysis of network flow data, enabling network security analysts to rapidly query large historical traffic data sets [99].

The installation consists of the following categories of applications:

- *the packing system* — collecting flow data (IPFIX, NetFlow v9, or NetFlow v5) and converting them into a more space efficient format, recording the packed records into binary flat files,
- *the analysis suite* — tools for reading the created flat files that can perform various query operations, such as per-record filtering or statistical analysis of groups of records.

This set of tools can be used during the Analyze phase of the OSCAR process model. Moreover, in the packing system, it can also be used during the Collect phase of the OSCAR model when collecting flow data.

4.1.65 SilentRunner

A SilentRunner is a part of the AccessData Platform family of products. The SilentRunner Sentinel worked as visibility for network traffic [11].

The features of SilentRunner Sentinel include capturing real-time network data in all OSI layers (including VoIP), visualization of the network activity, audit logs, and alerts, determining the root cause of a security breach, building integrated maps, conducting post-event analysis, and reconstructing events [10, 14].

The SilentRunner Sentinel seems no longer to be an active tool. This tool could be useful during the Collect and Analyze phases of the OSCAR process model.

4.1.66 Skyhook

Skyhook is a company that develops geo-positioning software solutions. They focus on location positioning, context, and intelligence [134].

One of the first products of this company was a Wi-Fi Positioning System (WPS). The WPS is a software location solution for determining the location of devices using land-based Wi-Fi access points [7].

Now, Skyhook's products include

- *Skyhook Precision Location* — The fast, accurate location for any app or device, available for Linux, Windows, MacOS, and Android systems,

- *Skyhook Context SDK* — client-side geofences, comprehensive DB locations, and insight into offline user behavior, available for iOS and Android systems,
- *Skyhook Geospatial Insights* — insight into the localization of mobile consumer behavior, available for iOS, Android, and Web systems [134].

The Skyhook products can assist during the Obtain phase of the OSCAR process model.

4.1.67 SmartWhois

A SmartWhois is a commercial network information utility used to look up information about the IP address, hostname, or domain. The SmartWhois can also provide information about the country, state or province, city, name of the network provider, administrator, and technical support contact information. This tool is available for Windows systems [100].

This SmartWhois can be used during the Analyze phase of the OSCAR process model.

4.1.68 smtpdump

A smtpdump is a free network tool that extracts SMTP information from PCAP files. It was written for the Network Forensic Puzzle #2 Contest [148].

This tool can be used during the Analyze phase of the OSCAR process model.

4.1.69 snoop

A Snoop is an open-source network tool for Windows systems. It is a WPF spying utility used to browse the visual tree of a running application and change properties, view triggers, or set breakpoints for property changes [163].

This tool can be used during the Obtain and Analyze phases of the OSCAR process model.

4.1.70 snort

A snort is an open-source IPS tool that uses rules for defining malicious network activity. It finds packets that match against rules and generates alerts. It is a multiplatform tool available for Windows, Linux, and FreeBSD systems. The snort can be used as a packet sniffer, a packet logger, or a full-blown IPS system [101].

This tool can be used during the Collect and Analyze phases of the OSCAR process model.

4.1.71 softflowd

A Softflowd is a software NetFlow probe. It is a flow-based network traffic analyzer that semi-statefully tracks traffic flows. The Softflowd can read a capture file or listen on the specified interface. The flows can be summarised by softflowd or reported via NetFlow. It is designed for minimal CPU load on busy networks [102].

The Softflowd can assist in the Collect phase of the OSCAR process model.

4.1.72 TCPDstat

A tcpdstat is a Linux open-source tool that analyzes network traffic data files (dump files) and provides statistics [30, 137].

It can be used during the Analyze phase of the OSCAR process model.

4.1.73 tcpdump/libpcap

A tcpdump and a libpcap are open-source network tools for Linux systems. The tcpdump is a command-line packet analyzer, and the libpcap is a portable C/C++ library for network traffic capture [105].

The tcpdump can capture, display, and store network traffic data. It also can work with the already captured or real-time packets. Using flags and parameters when running the tcpdump, many functions can be set, for example, specifying the read/write file, interface, and other functions defined in the manual page [105].

These tools can be used during the Collect phase of the OSCAR process model, and they can also be applied during the Analyze phase.

4.1.74 TCPFlow

A TCPFlow is an open-source network tool similar to the commercial tool NetIntercept. This tool was developed by Jeremy Elson, today it is maintained by Simson Garfinkel. It can be run on Linux, Windows, and MacOS systems [144, 145].

This tool aims to capture the network data, process them as TCP connections, store each flow into a separate file. Therefore, one typical TCP flow has two files, one for each direction. In addition to the live data capturing, the tcpflow can also process already captured data in capture files. Each created file contains source IP and port, and destination IP and port, in the filename. These created files after processing the packets, are used for later analysis. There are also many options that can be used when running the tcpflow, such as interpreting HTTP responses [145].

The tcpflow can be used to obtain HTTP session content, including web page reconstruction or malware extraction. Moreover, the tcpflow can create the output report in the DFXML format that contains detailed information, including the system information and every TCP flow information [145].

The tcpflow can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.75 TCPReplay

A TCPReplay is a suite of tools for Unix systems for editing and replaying network traffic. This toolset works with the already captured network traffic data into capture files [151, 152].

This suite contains the following tools: tcpreplay (replays pcap files), tcpreplay-edit (replays pcap files with option to modify packets), tcpliveplay (replay TCP pcap files directly to servers), tcpprep (pcap file pre-processor), tcprewrite (pcap file editor which rewrites packet headers), tcpcapinfo (raw pcap file decoder and debugger), and tcpbridge (bridge two network segments) [151].

Other features of the TCPReplay include support for netmap, flow statistics and analysis, and support for both single and dual NIC modes for testing both sniffing and in-line devices [151].

The TCPReplay can assist in the Analyze phase of the OSCAR investigative methodology.

4.1.76 tcpslice

A tcpslice is an open-source tool that extracts portions of packet trace files. It can concatenate multiple pcap files together or extract time slices from one or more pcap files. This tool was developed by "Lawrence Berkeley National Laboratory" and is now maintained by "The Tcpdump Group" [158].

The tcpslice can be used during the Analyze phase of the OSCAR process model.

4.1.77 TCPStat

A tcpstat is a Unix command-line tool that reports network interface statistics. The tcpstat can work with already captured traffic data in a dump file or monitor specific interfaces. This tool's statistics include bandwidth, number of packets, packets per second, average packet size, a standard deviation of packet size, and interface load [150].

The tcpstat can assist in the Collect and Analyze phases of the OSCAR process model.

4.1.78 TCPTrace

A tcptrace is a Unix command-line tool that analyzes the TCP connections from dump files. This tool provides detailed information about TCP connections by sifting through dump files. It can work also on Windows and MacOS systems. The provided statistics include packet statistics, RTT statistics and CWND (Congestion Window) statistics [157, 106].

The tcptrace can be used during the Analyze phase of the OSCAR process model.

4.1.79 TCPXtract

A TCPXtract is an open-source network tool used to extract files from network traffic based on file signatures. This tool can be used against a live network (using the libpcap library) or a tcpdump formatted capture file [3].

The TCPXtract can assist in the Analysis phase of the OSCAR process model.

4.1.80 traceroute/tracert

A traceroute and a tracert are built-in monitoring and network diagnostic tools. The traceroute is a command for Unix and MacOS systems; the tracert is a command for Windows systems. Both commands work the same way. They print the route packets trace to network host using the ICMP messages and IP protocol's TTL [108, 38].

These tools can be used during the Obtain and Analysis phase of the OSCAR process model.

4.1.81 tshark

A tshark is a network protocol analyzer. It is a part of the Wireshark tool and can be installed together with the Wireshark. Since Wireshark is a multiplatform tool, also tshark can be run on many operating systems, but Linux is preferred. The tshark provides a command-line interface [109].

It provides capturing packet data from a live network or reading packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. It also supports applying filters on captured data using the parameters [109].

As Wireshark, also tshark can be used during the Collect and Analyze OSCAR process model phases.

4.1.82 VisualRoute

A VisualRoute a network tool that offers a wide variety of network features. It is available for Windows and MacOS systems in full or lite version [110].

The features include continuous trace routing, reverse tracing, response time graphing, port probing, network scanning, trace route history, side -by-side trace route comparison, route analysis (NetVu), custom maps, remote access server, and save traceroutes as text, image or HTML. Extra features include Whois lookups, IP Locations, traceroute tests from Visualware servers, and IPv6 compatibility [110].

The VisualRoute can be helpful during the Obtain and Analyze phases of the OSCAR investigative methodology.

4.1.83 Web Historian

A Web Historian is an extension for the Google Chrome browser. It is used to visualize the web browsing history. It can visualize visited websites, searched items, network, time heatmaps, and data tables. It also provides a comparison with the last week's statistics. This tool is available in the Educational and Community edition [111].

This network tool can be useful during the Obtain and Analyze phase of the OSCAR process model.

4.1.84 WebScarab

A WebScarab is a framework that is used to analyze applications using HTTP/HTTPS protocols. This tool is available for Windows and Linux systems [92].

The WebScarab provides several modes and plugins. It is mainly used as a proxy intercepting HTTP and HTTPS communications, allowing an investigator to review and edit requests and responses. Other features include: reviewing the conversations, bandwidth simulator, parameter fuzzer, searching, BeanShell, sessionID analysis, and others [92].

It is recommended to have a good knowledge of HTTP protocol and code writing basics when using this tool [92].

This tool can assist in the Analyze phase of the OSCAR process model.

4.1.85 whois

A whois is a command for performing the registration record for a specified domain name or IP address. It is available in many operating systems, including Linux and Windows. It searches for an object in a RFC 3912[4] database [112, 8].

The whois can be used during the Analyze phase of the OSCAR model.

4.1.86 Wikto

A Wikto is an Nikto version for Windows platform with some additional features. It is a web server scanner. Some extra features include: fuzzy logic error code checking, a

[4]https://tools.ietf.org/html/rfc3912

back-end miner, Google assisted directory mining, and real time HTTP request/response monitoring [167].

The Wikto provides graphical user interface. For full instalation, there is needed also WinHTTrack[5] and HTTprint[6] [113].

This tool can be used during the Obtain and Analysis phases of the OSCAR process model.

4.1.87 windump/WinPcap

A Windump is a free command-line network analyzer for Windows systems. This tool can be understood as an Windows version of the tcpdump. The Windump is used to capture, analyze, and export the network traffic data [42].

A WinPcap is an industry-standard Windows packet capture library. It allows applications to capture and transmit network packets. It can be understood as an Windows version of the libpcap. This library is used in many network tools, including Windump, Wireshark, Nmap, Snort, and ntop [43].

These tools assist in the Collect and Analyze phase of the OSCAR model.

4.1.88 Wireless Network Watcher

A Wireless Network Watcher is a freeware utility that scans a currently connected wireless network and displays the currently connected devices. This tool can be run on Windows systems. It provides the basic information of the connected devices, such as IP address, MAC address, computer name, and network card manufacturer [114].

The Wireless Network Watcher can assist in the Obtain phase of the OSCAR process model when obtaining information about the environment.

4.1.89 Xplico

An Xplico is an open-source network forensic analysis tool that supports many protocols, including HTTP, SIP, IMAP, POP, SMTP, TCP, UDP, IPv6, Facebook, MSN, RTP, IRC, and Paltalk. This tool aims to gather application information from captured network traffic data, such as emails, HTTP content, VOIP calls, FTP, and others. It is not a network protocol analyzer [44].

It allows concurrent access by multiple users where one user can manage one or more cases. It provides a web user interface and can also be used as a cloud network forensic tool. Other features include multithreading, real-time elaboration, reverse DNS lookup, IPv4 and IPv6 support, and modularity [44].

The Xplico can assists in the Analyze phase of the OSCAR process model.

4.1.90 YAF

A YAF, Yet Another Flowmeter, is an open-source flow sensor developed for Linux systems. It processes packet data, aggregates packets into flows, and export the information in IPFIX format. The YAF can work with dump files and also with the live captures from the interface [117].

[5] http://www.httrack.com/
[6] http://www.net-square.com/

In addition to yaf itself, the YAF toolchain also includes other tools, such us yafscii (printing in ASCII format), yafMeta2Pcap (PCAP metadata file parser and PCAP file creator), getFlowKeyHash (flow key calculator), airdaemon (run as a daemon process), filedaemon (poll a directory and move files), and yafzcbalance (load balance from zc interfaces) [117].

This tool can be used during the Collect and Analyze phases of the OSCAR process model.

4.1.91 Yersinia

A Yersinia is an open-source framework for performing attacks on the data link layer including attacks on STP and CDP network protocols. It can be run on Linux systems [156].

This tool can assist in the Obtain phase of the OSCAR process model when discovering the system's environment and vulnerabilities.

4.2 More tools

After describing the network tools from the previous literature survey, some more tools are described that were not mentioned. This section describes more tools that can assist in network forensics. There are too many tools to describe them all, and therefore only some of them are described. Both open-source and commercial tools are involved.

4.2.1 Bricata

A Bricate is a commercial end-to-end network detection and response platform. It fuses signature inspection, stateful anomaly detection, and machine learning-powered malware conviction. It provides the real-time detection, response, hunting and defending against threats [56].

This tool can be useful during the Collect and Analyze phase of the OCSAR process model.

4.2.2 CapAnalysis

A CapAnalysis is an open-source web-based capture file viewer that can work with more than one PCAP file. It performs indexing of data set of PCAP files and visualizes their contents in many forms – flows, statistics, source IPs, destination IPs, per hour statistics, Geo map, protocols, and timeline. The data or flows can also be filtered according to the IP, port, protocol, country, data volume, or date. This tool is available for Linux systems, like Debian or Ubuntu [45].

The CapAnalysis can be used in the Analyze phase of the OSCAR process model.

4.2.3 CapLoader

A CapLoader is a Window-based commercial tool that can handle large amounts of captured network traffic. It performs indexing of capture files and visualizes their contents as a list of TCP and UDP flows. It provides filtering of the packets and exporting packets/flows into the packet analyzer tool [118].

In addition to the professional edition, the 30 days trial is also available. This trial version can handle 500 GB of captured data, supports pcapng and IPv6, can filter key-

words and provides keywords string search. Other features of the trial version include flow transcript view, DNS parser, initial RTT calculation, network packet carving, input filter (BPF), display filter (BPF), hide flows in GUI, and service regularity/period detection. The professional version does not have a limit for PCAP files, and in addition to the features of the trial edition, it provides Alexa and Cisco Umbrella top 1M lookup, port independent protocol identification (PIPI), OS fingerprinting, Geo-IP localization, ASN lookup, regular expression (regex) search, select flows from the log file or PCAP file, and Wireshark style Coloring [118].

The CapLoader can assist during the Analyze phase of the OSCAR process model.

4.2.4 chkrootkit

A chkrootkit is a free tool that locally checks for signs of a rootkit. This is a multiplatform tool that can be run on Linux, Windows, MacOS, Solaris, or BSD systems [155].

The chkrootkit can be used in the Obtain and Analyze phases of the OSCAR process model.

4.2.5 DoHlyzer

A DoHlyzer is a network flow tool that detects and characterizes DoH (DNS over HTTPS) traffic. It can be run using Python, and therefore it is a multiplatform tool [154].

The DoHlyzer is a set of tools that consists of the following modules:

- *Meter* — captures packets or reads PCAP file, groups packets into flows, and extracts statistical and time-series features for traffic analysis,
- *Analyzer* — creates the DNN models and benchmark them against the aggregated clumps file (can be created by the Meter module),
- *Visualizer* — visualizes the clumps files.

The DoHlyzer can be used during the Collect and Analyze phases of the OSCAR process model.

4.2.6 Dshell

A Dshell is an open-source network forensic analysis framework written in Python. This tool works with plugins that can be run on the capture file or live on an interface. Key features include deep packet analysis, robust stream reassembly, IPv4 and IPv6 support, and custom output handlers. It also supports elasticsearch to store the output [168].

The plugins can be chained. The available plugins include dhcp, dns, filter, flows, ftp, http, malware, misc, nbns, portscan, protocol, ssh, ssl, tftp, visual, voip, and wifi [168].

The Dshell can be used during the Collect and Analyze phases of the OSCAR process model.

4.2.7 findject.py

A findject.py is an open-source python script that can detect TCP packet injection attacks in HTTP sessions. Unlike the IDS solutions, this script can also properly detect Man-on-the-Side (MOTS) attacks. This script analyzes PCAP files and prints the output of gained injections [120].

The findject.py can assist in the Analyze phase of the OSCAR process model.

4.2.8 Forensics Investigation Toolkit (FIT)

The Forensics Investigation Toolkit (FIT) is a licensed tool developed by the "Decision Group Inc." This toolkit is available for Windows systems and can analyze network packet data. These data can be read from a PCAP file or real-time captured. The FIT provides a graphical user interface. The other features include full-text searching, bookmarking, immediate parsing and reconstruction of the raw data into categories, WhoIS and Google Map integration, association analysis, and export of the analyzed data [66].

The FIT can be used during the Collect, Analyze, and Report phases of the OSCAR process model.

4.2.9 Haka

A Haka is an open-source security-oriented framework that allows to describe protocols and apply security policies on captured traffic or live on interfaces. The two main features of the Haka are writing security rules and specifying network protocols and their underlying state machine [20, 21].

The Haka project provides modules for packet capturing, alerting, and logging. It also provides a tool suite that consists of the following programs:

- *haka* — the main program of the collection, can capture packets (pcap or nfqueue) and filter or alter them according to the specified Haka policy file, usually launched as a daemon,
- *hakactl* — allows to control a running haka daemon (displays haka status, start or stop the haka daemon, show logs, debug haka rules),
- *hakapcap* — a tool to quickly apply a Haka policy file to a pcap file [21].

A Hakabana is a monitoring tool to visualize network traffic going through Haka in real-time using Kibana and Elasticsearch. Both Haka and Hakabana can be installed through Debian package, Tarball install, or Live ISO [20].

The Haka can assist in the Collect and Analyze phases of the OSCAR process model.

4.2.10 HoneyBadger

A HoneyBadger is an open-source TCP stream analysis tool for detecting and recording TCP injection attacks (Quantum Insert detector). It performs passive analysis of TCP traffic and detects evidence of MOTS attacks. It can work with PCAP files or analyze an interface [25].

The HoneyBadger provides a *honeybadgerReportTool* that is a report deserialization tool. It displays a dump output (ASCII and hex) [25, 165].

This tool can be used during the Analyze phase of the OSCAR process model.

4.2.11 IP Address Tracker and IP Address Manager

An IP Address Tracker is a free network tool that can scan, track, and manage IP addresses and obtain detailed IP histories and event logs. It is a reduced feature set version of commercial tool IP Address Manager (IPAM) developed by SolarWinds [123].

Key features of IP Address Tracker include managing up to 254 IP addresses, detecting IP conflicts, getting detailed IP histories and event logs, getting detailed reporting for IP addresses, and monitoring subnets [123].

The licensed tool IPAM can, in addition to the features of IP Address Tracker, manage up to 2 million IP addresses, monitor DNS and DHCP, assess DHCP, DNS, and IP address role-based task permissions, and administer integrated DNS and DHCP. The Solarwinds provides the free 30 days trial of IPAM [122, 123].

These tools can assist in the Obtain, Collect, and Analyze phases of the OSCAR process model.

4.2.12 Log Analyzer

Log Analyzer is a commercial network tool for log and event collection and analysis. It is integrated with the Solarwinds Orion platform. The other key features include powerful search and filter, real-time log stream, event log tagging, and flat log file ingestion. There is also available free 30 days trial [124].

This tool can be used during the Collect and Analyze phases of the OSCAR process model.

4.2.13 LogRhythm NetMon and LogRhythm NetMon Freemium

A LogRhythm NetMon is a commercial SIEM network monitoring tool that helps detect, stop, and recover from attacks. It provides real-time visibility, security analytics, and network-based incident response. The features include unstructured search across all network data, deep packet analytics, full packet capture and SmartCapture, automatic recognition of applications, continuous search-based alerting, data forwarding via Syslog, data processing up to 10 Gbps, unlimited packet capture storage, and metadata indexing up to 30 days [125].

A LogRhythm NetMon Freemium is a free version of the LogRhythm NetMon tool. The main functions are the same. There are only limits on processing, packet storage, and data forwarding — it is not able to forward data via Syslog, data processing rate is 1 Gbps, packet capture storage is 1 GB, and metadata indexing retention is 3 days [126].

These tools can be used during the Collect and Analyze phases of the OSCAR process model.

4.2.14 NetFlow Configurator

A NetFlow Configuration is a free version of the Solarwinds NetFlow Traffic Analyzer but with fewer functions. It can remotely activate NetFlow on network devices via SNMP. The key features include analyzing network performance, activating NetFlow, finding bandwidth hogs, bypassing the CLI with an intuitive GUI, setting up collectors for NetFlow data, specifying collector listening ports, and monitoring traffic data per interface [127].

The NetFlow Configurator can assist in the Collect and Analyze phases of the OSCAR process model.

4.2.15 NetFlow Traffic Analyzer (NTA)

A NetFlow Traffic Analyzer is a commercial network analyzer and bandwidth monitor. The other key features include application traffic alerting, VMware vSphere distributed switch support, performance analysis dashboard, and advanced application recognition. This tool is available for Windows systems (on-premise) or may be installed on VMware Virtual Machines and Microsoft Virtual Servers. The free 30 days trial version is also available [128].

The NTA can be used during the Analysis phase of the OSCAR process model.

4.2.16 Netfox Detective

A Netfox Detective is an open-source Windows network forensic analysis tool that extracts the application content from the communication. This tool supports the following application protocols — BTC – Ṡtratum, DNS, Facebook, FTP, Hangouts, HTTP, OSCAR – ICQ, IMAP, Lide.cz, Messenger, Minecrat, MQTT, POP3, RTP, SIP, SMTP, SPDY, Twitter, Webmails – various services, Xchat.cz, XMPP, YMSG. It can work with capture files; the live capture is not supported. The key features include multiple PCAPs support, large PCAPs support, advanced visualization, filters, and full-text search [46, 159, 28].

The Netfox Detective can be used in the Analyze phase of the OSCAR process model.

4.2.17 NetScanTools Basic and NetScanTools Pro

A NetScanTools Basic Edition is a freeware set of essential network tools that includes DNS Tools (simple IP/hostname resolution, computer name, IP and DNSs), Ping, Graphical Ping, Ping Scanner, Traceroute, and Whois [77].

A NetScanTools Pro is a powerful commercial set of network tools that include many network tools and utilities. It covers the following category of tools:

- *Active Discovery and Diagnostic Tools* — to locate and test devices connected to the network (ARP Ping, DHCP Server Discovery, Email Validate, Finger, Network Routing Visualizer, OS Fingerprinting, Ping, Port Scanner, SMB Scanner, SSL Certificate Scanner, and others),
- *Passive Discovery Tools* — to find information from third parties or to monitor the activities of devices connected to the network (Connection Monitor, Network Connection Endpoints, Packet Capture, Passive Discovery, Real-Time Blacklist Check, Whois),
- *DNS Tools* — to help to find problems with DNS (Simple Query - IPv4/IPv6, Who Am I?, Flush Default DNS Cache, Edit DNS HOSTS File, Auth Serial Check, DNS Verify, IP Drilldown, SPF/Domain Keys, DNS List Speed Test, IP or Hostname to ASN, Get VOIP/Misc SRV Records, and others),
- *Local Computer and General Information Tools* — provide information about the local computer's network and also include general information tools (ARP Cache, Cache Forensics, IP to Country, IP/MAC Address Database, Launcher, Network Interfaces and Statistics, IPv6 Network Neighbors, TCP/UDP Service Lookup, and others) [78].

These sets of network tools can be used in Obtain and Analyze phases of the OSCAR process model.

4.2.18 Network Topology Mapper

A Network Topology Mapper is a commercial network tool used to automatically plot the network. There is available 14 days free trial version. The key features include automatic device discovery and mapping, building multiple maps from a single scan, exporting network diagrams to Visio, auto-detecting changes to network topology, performing multi-level network discovery, and addressing regulatory PCI compliance. This tool is available for Windows systems [130].

This tool can assist in the Obtain phase of the OSCAR process model.

4.2.19 Network Performance Monitor

A Network Performace Monitor is a commercial tool that provides a lot of features. The goal is to monitor whole network infrastructure. The user can see network traffic, configuration and performance details. It supports on-prem, hybrid and cloud services. It uses hop-by-hop analysis with NetPath [129].

This tool can be used during the Obtain and Analyze phases of the OSCAR process model.

4.2.20 PassiveDNS

A PassiveDNS is an open-source network sniffer that collects DNS record passively. It can be used in Incident handling, Network Security Monitoring (NSM) and general digital forensics. This tool can read PCAP file or capture the traffic data from an interface. It also can export the DNS-server answers to a log file. The other features are IPv4 and IPv6 support, and it can parse DNS traffic over TCP or UDP. One of the typical use case is searching for domain or IP history when working on an incident [142].

The PassiveDNS can be used during the Collect and Analyze phases of the OSCAR process model.

4.2.21 PcapXray

A PcapXray is an open-source network tool that reads a PCAP file and visualizes the network in a network diagram. It displays hosts in the network, network traffic, highlight significant traffic and Tor traffic as well as potentially malicious traffic, including data involved in the communication [47].

The PcapXray can assist in the Obtain and Analyze phases of the OSCAR process model.

4.2.22 Port Scanner

A Port Scanner is a free network tool that scans available IP addresses and their corresponding TCP and UDP ports to identify network vulnerabilities. It provides lists of open, closed, and filtered ports for each scanned IP address. The other features include defining a DNS server, saving scan configurations, tracking user and endpoint device connection activity, and viewing and editing IANA port name definitions [132].

The Port Scanner can assist in the Obtain phase of the OSCAR process model.

4.2.23 Security Event Manager (SEM)

A Security Event Manager (SEM) is a commercial SIEM tool that provides security information and event management solution. The main features include centralized log collection and normalization, automated threat detection and response, integrated compliance reporting tools, an intuitive dashboard and user interface, and built-in file integrity monitoring. The free 30 days trial version is also available. The SEM is based on a manager and agent system, where the manager is distributed as a virtual machine, and agents can be installed on multiple platforms including Linux, Windows, and MacOS [133].

The SEM can assist in the Collect and Analyze phases of the OSCAR process model.

4.2.24 SplitCap

A SplitCap is a free tool used for splitting capture files. There can be specified criteria for splitting the PCAPs – BSSID (WLAN BSSID), Flow (5-tuple), Host (IP address), Host Pair (IP pairs), MAC address, Session (bi-directional flow), Time, and Packets Count. This tool can be run Linux and Windows systems [103].

The SplitCap can be used during the Strategize and Analyze phases of the OSCAR process model.

4.2.25 Splunk

Splunk is a SIEM tool. It is a commercial tool that is available as "Data-to-Everything Platform", "Splunk Cloud", "Splunk Enterprise", "Splunk Machine Learning Toolkit", and "Splunk Data Stream Processor". It covers cybersecurity solutions, observability, IT monitoring tools and other security products [135].

As a SIEM tool, the Splunk can be used during the whole investigation process, including collecting and analyzing the data.

4.2.26 SSLsplit

An SSLsplit is an open-source tool used for man-in-the-middle attacks against SSL/TLS encrypted network connections. This tool is available for multiple platforms, including Linux, FreeBSD, OpenBSD, Debian, and MacOS systems. It supports plain TCP, plain SSL, HTTP and HTTPS connections over both IPv4 and IPv6. It also fully supports Server Name Indication (SNI) and can work with RSA, DSA and ECDSA keys and DHE and ECDHE cipher suites. There are also logging options that include traditional SSLsplit connect and content log files as well as PCAP files and mirroring decrypted traffic to a network interface. Certificates, master secrets, and local process information can also be logged [161].

The SSLsplit can be used during the Analyze phase of the OSCAR process model.

4.2.27 Stenographer

A Stenographer is an open-source packet capture tool. It is designed to write packets to disk, store as much history as it can, and read a very small percentage of packets from disk based on analyst needs [104].

The Stenographer can be used in the Collect phase of the OSCAR investigative methodology.

4.2.28 Suricata

A Suricata is an open-source network threat detection engine that is capable of real time intrusion detection (IDS), inline intrusion prevention (IPS), network security monitoring (NSM) and offline pcap processing. The Suricata uses rules and signature language with Lua scripting. It provides YAML and JSON output, and it can integrate with tools like existing SIEMs, Splunk, Logstash/Elasticsearch, Kibana. This tool is available for multiple platforms including Linux, Windows, and MacOS [136].

The Suricata can assist in the Collect and Analyze phases of the OSCAR process model.

4.3 Summary and comparison

This section listed and described many network tools that can be used in network forensics. As one of this master's thesis outputs, these network tools overview is also publicly available at https://martinazembjakova.github.io/Network-forensic-tools-taxonomy/ using GitHub pages environment. The design of this website can be seen in appendix C.

Different types of tools were described, such as scanners, sniffers, analyzers, visualizers, IDS/IPS, SIEM tools, and others. Extended properties of tools are mentioned within the new designed taxonomy described in chapter 8 — scanners are described in table 8.6, sniffers in table 8.6, analyzers in tables 8.6 and 8.6, visualizers in table 8.6, IDS/IPS in table 8.7, SIEMs in table 8.8 and network diagnostic tools in table 8.6.

Chapter 5

Survey of datasets

A network trace dataset is a set of packet capture files that can be analyzed using the network packet analyzers. This chapter aims for some datasets that can be analyzed using the network analysis tools described in chapter 4.

In 2019, the authors of the article "A survey of network-based intrusion detection data sets" published in the journal "Computers & Security," researched the network-based datasets. They described available packet-based and flow-based datasets for IDS in the mentioned article. The discussed datasets include AWID (2016), Booters (2015), Botnet (2014), CIC DoS (2017), CICIDS 2017 (2018), CIDDS-001 (2017), CIDDS-002 (2017), CDX (2009), CTU-13 (2014), DARPA (2000), DDoS 2016 (2016), IRSC (2015), ISCX 2012 (2012), ISOT (2011), KDD CUP 99 (2018), Kent 2016 (2015), Kyoto 2006+ (2011), LBNL (2005), NDSec-1 (2017), NGIDS-DS (2017), NSL-KDD (2009), PU-IDS (2015), PUF (2018), SANTA (2014), SSENET-2011 (2011), SSENET-2014 (2014), SSHCure (2014), TRAbID (2017), TUIDS (2012), Twente (2009), UGR'16 (2018), UNIBS (2009), Unified Host and Network (2017), UNSW-NB15 (2015) [178].

The NETRESEC also provides a list of several publicly available datasets separated into categories: Cyber Defence Exercises (CDX), Malware Traffic, Network Forensics, SCADA/ICS Network Captures, Capture the Flag Competitions (CTF), Packet Injection Attacks/Man-on-the-Side Attacks, Uncategorized PCAP Repositories, and Single PCAP files [173].

A detailed description of several datasets is provided in the following sections. The described datasets were selected based on diversity to cover different use cases and protocols. These datasets serve as a basis for demonstrating the tools in the use cases described in chapter 6.

5.1 Canadian Institute for Cybersecurity datasets

The "Canadian Institute for Cybersecurity" created datasets that are focused of several aspects of cyersecurity. The currently available datasets include Android malware, DoS, VPN, Tor, IPS/IDS, and DNS over HTTP traffic. Some datasets are described in the following sections.

5.1.1 CIC-DDoS2019

The dataset "DDoS2019" is a dataset of "Canadian Institute for Cyersecurity" that contains benign and most up-to-data DDoS attacks. The dataset contains realistic background

traffic. There was built the abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols [171].

The dataset contains captured data from 2 days. The first day, the training day, took place on 3.11.2018, started at 09:40 and ended at 17:35 local time (converted into UTC time format: from 12:40 UTC to 20:35 UTC). The second day, the testing day, took place on 1.12.2018, started at 10:30 and ended at 17:15 local time (converted into UTC time format: from 13:30 UTC to 20:15 UTC). The original dataset description uses wrong dates in the research paper (followed by switched naming of the first and the second day) – they use the first day as the January 12th and the second day as the March 11th. The information used in this document is based on the PCAP files and CSV files of this dataset, not the research paper. Therefore they differ from the original dataset's descriptions. This dataset includes PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, SNMP, SSDP, WebDDoS, and TFTP attacks [180].

The table 5.1 contains the victim network information. The attacker network consists of the third party company.

The original dataset PCAPs are split into multiple PCAP files. The first day contains 145 PCAPs, the second day contains 818 PCAPs. The individual capture days of the dataset are discussed in the following sections. Firstly, the essential time frames of some individual PCAPs of that day are described (timestamps of the start and end of attacks). Secondly, the annotation of the whole day is provided. The time is in the UTC format. The description of the attacks are based on the research paper "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy" [180].

Network Information	
Firewall	205.174.165.81 (Fortinet)
Victim	192.168.50.4 (First day), 192.168.50.1 (Second day) (Web server Ubuntu 16.04) 192.168.50.9 (First day), 192.168.50.8 (Second day) (Win 7 Pro) 192.168.50.6 (First day), 192.168.50.5 (Second day) (Win Vista) 192.168.50.7 (First day), 192.168.50.6 (Second day) (Win 8.1) 192.168.50.8 (First day), 192.168.50.7 (Second day) (Win 10 Pro 32)

Table 5.1: Network information of CIC-DDoS2019 dataset [180]

First day – training day

PCAP filename	Time range (UTC)	
SAT-03-11-2018_000	12:18:16.583626	13:01:48.920573
SAT-03-11-2018_011	13:09:00.565557	13:21:56.124692
SAT-03-11-2018_068	13:29:52.072724	13:34:11.268896
SAT-03-11-2018_106	13:42:57.176611	13:54:11.631481
SAT-03-11-2018_136	14:01:43.652741	14:14:54.297925
SAT-03-11-2018_137	14:14:54.298079	14:30:25.830426
SAT-03-11-2018_145	17:51:18.675623	20:36:56.349321

Table 5.2: Important time frames of individual PCAP files of the first day 3.11.2018

- **Number of packets:** 61 407 883

- **Timeline:** 2018-11-03 12:18:16.583626 UTC – 2018-11-03 20:36:56.349321 UTC
- **Involved hosts:** 172.16.0.5, 192.168.50.4, 192.168.50.6, 192.168.50.7, 192.168.50.8, 192.168.50.9
- **Protocols:** 3Com XNS, 3GPP2 A11, 802.11, A21, ADP, AH, ALC, ALLJOYN-ARDP, ALLJOYN-NS, AMS, AMT, ANSI C12.22, AODV, ARP, ASAP, ASTERIX, ATH, AX4000, AYIYA, Auto-RP, BACnet-APDU, BAT_BATMAN, BAT_GW, BAT_VIS, BFD Control, BJNP, BOOTP, BROWSER, BVLC, Bundle, CAPWAP-Control, CAPWAP-Data, CDP, CLDAP, CLNP, CN/IP, CUPS, CoAP, DAYTIME, DB-LSP-DISC, DCC, DCERPC, DCP-AF, DCP-PFT, DHCP, DHCPv6, DIS, DMP, DNP, DNS, DPNET, DPP, DSR, DTLS, DTP, EAP, EAPOL, ECAT, ECATF, ECHO, ECMP, EGD, EIGRP, ENIP, ENRP, ESP, Elasticsearch, GPRS-NS, GSM SIM, GSM-TAP, GTP, Geneve, H.225.0, H.248, HART_IP, HCrt, HPEXT, HTTP, HTTP/XML, HiQnet, IAPP, IAX2, ICMP, ICMPv6, ICP, IEEE 802.15.4, IGMPv3, IO-RAW, IP, IPVS, IPX, IPv4, IPv6, ISAKMP, ISO, KDP, KINK, KNET, KPASSWD, KRB4, KRB5, L2TP, L2TPv3, LBT-RU, LDP, LISP, LLC, LLDP, LLMNR, LMP, LTP Segment, LWAPP, MANOLITO, MDNS, MEMCACHE, MIH, MIPv6, MNDP, MPLS, MSMMS, MSproxy, MiNT, MobileIP, Modbus/UDP, NAT-PMP, NBDS, NBNS, NCP, NHRP, NTP, NXP 802.15.4 SNIFFER, Nano, OCSP, OLSR v1, OSPF, OpenVPN, PCP v1, PCP v2, PFCP, PKTC, PNIO, POWERLINK/UDP, PTPv2, Pathport, Portmap, QUAKE, QUAKE2, QUAKE3, QUAKEWORLD, RADIUS, RDT, RIP, RIPng, RIPv1, RIPv2, RRoCE, RSIP, RSVP, RTCP, RTPproxy, RakNet, SABP, SAP, SAP/SDP, SCTP, SCoP, SEBEK, SIP, SNA, SNMP, SRVLOC, SSDP, SSH, SSHv2, SSL, SSLv2, STP, STUN, SliMP3, Syslog, TAPA, TC-NV, TCP, TETRA, TFTP, TIME, TIPC, TLSv1, TLSv1.2, TLSv1.3, TPCP, TPKT, TS2, TSP, TZSP, UAUDP, UDP, UDP/MIKEY, ULP, UNKNOWN, VITA 49, Vines IP, Vuze-DHT, VxLAN, WHO, WLCCP, WSP, WTLS+WSP, WTLS+WTP+WSP, WTP+WSP, X.25, XTACACS, XYPLEX, collectd, eDonkey, lw_res, openSAFETY over UDP, packetbb
- **Attacks:** PortMap (12:43 - 12:51), NetBIOS (13:01 - 13:09), LDAP (13:21 - 13:30), MSSQL (13:33 - 13:43), UDP (13:52 - 14:04), UDP-Lag (14:14 - 14:24), SYN (14:28 - 20:35)

Second day – testing day

PCAP filename	Time range (UTC)	
SAT-01-12-2018_0	13:17:10.711517	14:36:06.133219
SAT-01-12-2018_027	14:36:59.617966	14:37:02.505099
SAT-01-12-2018_0188	14:44:33.210758	14:46:30.026952
SAT-01-12-2018_0190	14:48:26.225518	14:51:39.813446
SAT-01-12-2018_0194	14:57:43.395236	15:00:26.604875
SAT-01-12-2018_0195	15:00:26.604876	15:03:06.989875
SAT-01-12-2018_0305	15:11:56.643849	15:12:00.253348
SAT-01-12-2018_0324	15:12:59.381993	15:13:02.627201
SAT-01-12-2018_0381	15:22:58.494906	15:23:07.641045
SAT-01-12-2018_0387	15:23:53.444988	15:24:02.172861
SAT-01-12-2018_0407	15:26:51.191475	15:27:00.259048
SAT-01-12-2018_0414	15:27:56.123811	15:28:05.086642
SAT-01-12-2018_0443	15:32:32.915441	15:37:20.477580
SAT-01-12-2018_0446	15:37:56.549979	15:38:15.028105
SAT-01-12-2018_0467	15:44:53.078912	15:45:12.275065
SAT-01-12-2018_0470	15:45:48.874827	15:46:07.180524
SAT-01-12-2018_0486	16:00:13.902782	16:13:19.200714
SAT-01-12-2018_0501	16:14:53.513548	16:15:00.789394
SAT-01-12-2018_0510	16:15:58.289530	16:16:05.415448
SAT-01-12-2018_0526	16:17:53.645195	16:18:00.844202
SAT-01-12-2018_0535	16:18:57.740588	16:19:04.830961
SAT-01-12-2018_0577	16:28:47.412567	16:29:26.085243
SAT-01-12-2018_0578	16:29:26.085244	16:30:14.334464
SAT-01-12-2018_0584	16:33:24.858564	16:34:12.351220
SAT-01-12-2018_0586	16:34:45.229199	16:35:19.639364
SAT-01-12-2018_0589	16:35:55.110452	16:36:11.265191
SAT-01-12-2018_0817	18:02:49.179574	20:59:05.159078
SAT-01-12-2018_0818	20:59:05.159081	21:16:39.140675

Table 5.3: Important time frames of individual PCAP files of the second day 1.12.2018

- **Number of packets:** 250 783 287
- **Timeline:** 2018-12-01 13:17:10.711517 UTC – 2018-12-01 21:16:39.140675 UTC
- **Involved hosts:** 192.168.0.1, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8
- **Protocols:** 3Com XNS, 3GPP2 A11, 802.11, A21, ADP, ALC, ALC/XML, ALLJOYN, ALLJOYN-ARDP, ALLJOYN-NS, AMS, AMT, ANSI C12.22, AODV, ARP, ASAP, ASF, ASTERIX, ATH, AX4000, AYIYA, Armagetronad, Auto-RP, BACnet-APDU, BAT_BATMAN, BAT_GW, BAT_VIS, BFD Control, BJNP, BOOTP, BROWSER, BSSGP, BVLC, Bundle, CAPWAP-Control, CAPWAP-Data, CAT-TP, CDP, CLDAP, CLNP, CN/IP, CUPS, CoAP, DAYTIME, DB-LSP-DISC, DCC, DCERPC, DCP-AF, DCP-PFT, DHCP, DHCPv6, DIS, DMP, DNP, DNS, DPNET, DPP, DSPv2 , DSR, DTLS, DTP, EAP, EAPOL, EAPOL-MKA, ECAT, ECATF, ECHO, ECMP,

EGD, EIGRP, ENIP, ENRP, ESP, Elasticsearch, Ethernet, FF, GPRS-LLC, GPRS-NS, GSM SIM, GSMTAP, GTP, GTPv2, Geneve, H.225.0, H.248, HART_IP, HCrt, HPEXT, HTTP, HTTP/XML, HiQnet, IAPP, IAX2, ICMP, ICMPv6, ICP, IEEE 802.15.4, IO-RAW, IP, IPMI, IPVS, IPX, IPv4, IPv6, ISAKMP, ISO, KDP, KINK, KNET, KPASSWD, KRB4, KRB5, L2TP, L2TPv3, LDP, LISP, LLC, LLDP, LLMNR, LMP, LTP Segment, LWAPP, MAC-Telnet, MANOLITO, MDNS, MEMCACHE, MIH, MIPv6, MNDP, MPLS, MSMMS, MSproxy, MiNT, MobileIP, Modbus/UDP, NAT-PMP, NBDS, NBNS, NCP, NEMO, NHRP, NTP, NXP 802.15.4 SNIFFER, Nano, NetBIOS, OCSP, OLSR v1, OSCORE, OSPF, OpenVPN, PCP v1, PCP v2, PFCP, PKTC, PN-PTCP, PNIO, POWERLINK/UDP, PTPv2, Pathport, QUAKE, QUAKE2, QUAKE3, QUAKEWORLD, RADIUS, RDT, RIP, RIPng, RIPv1, RIPv2, RMCP, RRoCE, RSIP, RSVP, RTCP, RTPproxy, RX, RakNet, SABP, SAP, SAP/SDP, SCTP, SCoP, SEBEK, SIP, SNA, SNMP, SPX, SRVLOC, SSDP, SSH, SSHv2, SSL, STP, SliMP3, Syslog, TACACS, TAPA, TC-NV, TCP, TETRA, TFTP, TIME, TIPC, TLSv1, TLSv1.2, TLSv1.3, TPCP, TPKT, TS2, TSP, TZSP, Thread, UAUDP, UDP, UDP/MIKEY, ULP, UNKNOWN, VITA 49, Vines IP, Vuze-DHT, VxLAN, WHO, WLCCP, WSP, WTLS+WSP, WTLS+WTP+WSP, WTP+WSP, X.25, XTACACS, XYPLEX, ZigBee IP, collectd, eDonkey, lw_res, openSAFETY over UDP, openSAFETY/UDP, packetbb

- **Attacks:** NTP (13:35 - 13:45), DNS (13:52 - 14:05), LDAP (14:22 - 14:32), MSSQL (14:36 - 14:45), NetBIOS (14:50 - 15:00), SNMP (15:12 - 15:23), SSDP (15:27 - 15:37), UDP (15:45 - 16:09), UDP-Lag (16:11 - 16:15), WebDDoS (16:18 - 16:29), SYN (16:29 - 16:34), TFTP (16:35 - 20:15)

5.1.2 CIC-IDS2017

The dataset "IDS 2017" contains benign and the most up-to-date common attacks. It reflects a realistic background traffic. This dataset contains the built abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols [172].

The captured data are spitted into 5 PCAP files according to the day of the week they were captured. The data are captured from 3.7.2017 12:00 PM UTC (Monday) to 7.7.2017 8:00 PM UTC (Friday), in local time from Monday 9:00 AM to Friday 5:00 PM. This dataset includes Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS attacks [179].

The following table 5.4 contains the network information of the dataset, including firewall, DNS server, attacker network and victim network.

The following description of the individual days is based on the dataset description and the research paper "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization" [172, 179].

Timeline is displayed in UTC (Coordinated Universal Time) format. Involved hosts displayed in each day include only hosts from network information (attackers, victim, firewall).

Network Information	
Firewall	205.174.165.80 172.16.0.1
DC and DNS Server	192.168.10.3 (Win server 2016)
Attackers	205.174.165.73 (Kali) 205.174.165.69, 205.174.165.70, 205.174.165.71 (Win)
Victim	192.168.10.50, 205.174.165.68 (Web server Ubuntu 16) 192.168.10.51, 205.174.165.66 (Ubuntu server 12) 192.168.10.19 (Ubuntu 14.4, 32B) 192.168.10.17 (Ubuntu 14.4, 64B) 192.168.10.16 (Ubuntu 16.4, 32B) 192.168.10.12 (Ubuntu 16.4, 64B) 192.168.10.9 (Win 7 Pro, 64B) 192.168.10.5 (Win 8.1, 64B) 192.168.10.8 (Win Vista, 64B) 192.168.10.14 (Win 10, pro 32B) 192.168.10.15 (Win 10, 64B) 192.168.10.25 (MAC)

Table 5.4: Network information of CIC-IDS2017 dataset [172]

Monday

- **Number of packets:** 11 709 971
- **Timeline:** 2017-07-03 11:55:58.598308 AM – 2017-07-03 8:01:34.472889 PM
- **Involved hosts:** 172.16.0.1, 192.168.10.5, 192.168.10.8, 192.168.10.9, 192.168.10.12, 192.168.10.14, 192.168.10.15, 192.168.10.16, 192.168.10.17, 192.168.10.19, 192.168.10.25, 192.168.10.50, 192.168.10.51,
- **Protocols:** ARP, BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, EPM, Elasticsearch, FTP, FTP-DATA, GQUIC, HTTP, HTTP/XML, ICMP, ICMPv6, IGMPv2, IGMPv3, IPv4, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, MP4, NBNS, NBSS, NTP, OCSP, OpcUa, PKIX-CRL, RPC_NETLOGON, SAMR, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TCP, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, UDP, WebSocket
- **Attacks:** None

Tuesday

- **Number of packets:** 11 551 954
- **Timeline:** 2017-07-04 11:53:32.364079 AM – 2017-07-04 8:00:31.076755 PM
- **Involved hosts:** 172.16.0.1, 172.16.0.10, 192.168.10.50, 205.174.165.68, 205.174.165.73, 205.174.165.80
- **Protocols:** ARP, BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, EPM, FTP, FTP-DATA, HTTP, HTTP/XML, ICMP, ICMPv6, IGMPv2,

IGMPv3, IPv4, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, MP4, NBNS, NBSS, NTP, OCSP, PKIX-CRL, RPC_NETLOGON, SAMR, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TCP, TLSv1, TLSv1.1, TLSv1.2, UDP

- **Attacks:** Brute Force, FTP-Patator (12:20 PM – 1:20 PM), SSH-Patator (5:00 PM – 6:00 PM)

Wednesday

- **Number of packets:** 13 788 878
- **Timeline:** 2017-07-05 11:42:42.084372 AM – 2017-07-05 8:10:19.780725 PM
- **Involved hosts:** 172.16.0.1, 172.16.0.10, 172.16.0.11, 192.168.10.50, 192.168.10.51, 205.174.165.66, 205.174.165.68, 205.174.165.73, 205.174.165.80
- **Protocols:** ARP, BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, DTLS, DTLSv1.2, EPM, FTP, FTP-DATA, HTTP, HTTP/XML, ICMP, ICMPv6, IGMPv2, IGMPv3, IPv4, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, MP4, MPEG, NBNS, NBSS, NTP, OCSP, RPC_NETLOGON, SAMR, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TCP, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, UDP, WebSocket
- **Attacks:** DoS slowloris (12:47 PM – 1:10 PM), DoS Slowhttptest (1:14 PM – 1:35 PM), DoS Hulk (1:43 PM – 2:00 PM), DoS GoldenEye (2:10 PM – 2:23 PM), Heartbleed Port 444 (6:12 PM – 6:32 PM)

Thursday

- **Number of packets:** 9 322 025
- **Timeline:** 2017-07-06 11:58:58.492265 AM – 2017-07-06 8:04:44.364012 PM
- **Involved hosts:** 172.16.0.1, 172.16.0.10, 192.168.10.8, 192.168.10.25, 192.168.10.50, 205.174.165.68, 205.174.165.73, 205.174.165.80
- **Protocols:** ARP, BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, EPM, FTP, FTP-DATA, HTTP, HTTP/XML, ICMP, ICMPv6, IGMPv2, IGMPv3, IPv4, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, MP4, MPEG PES, NBNS, NBSS, NTP, OCSP, PKIX-CRL, RPC_NETLOGON, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TC, TCP, TLSv1, TLSv1.1, TLSv1.2, UDP
- **Attacks:** Web attacks – Brute Force (12:20 PM – 1:00 PM), XSS (1:15 PM – 1:35 PM), Sql Injection (1:40 PM – 1:42 PM), Infiltration attacks – Dropbox download Win Vista (5:19 PM, and 5:20 PM – 5:21 PM, and 5:33 PM – 5:35 PM, 6:04 PM – 6:45 PM), Cool disk MAC (5:53 PM – 6:00 PM)

Friday

- **Number of packets:** 9 997 874
- **Timeline:** 2017-07-07 11:59:39.599128 AM – 2017-07-07 8:02:41.169108 PM
- **Involved hosts:** 172.16.0.1, 192.168.10.5, 192.168.10.8, 192.168.10.9, 192.168.10.14, 192.168.10.15, 192.168.10.50, 205.174.165.68, 205.174.165.69, 205.174.165.70, 205.174.165.71, 205.174.165.73, 205.174.165.80
- **Protocols:** ARP, BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, EPM, FTP, FTP-DATA, H1, HTTP, HTTP/XML, ICMP, ICMPv6, IGMPv1, IGMPv2, IGMPv3, IPv4, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, MPEG PES, NBNS, NBSS, NTP, OCSP, OMAPI, PKIX-CRL, RPC_NETLOGON, SAMR, SCTP, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TCP, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, UDP
- **Attacks:** Botnet ARES (1:02 PM – 2:02 PM), DDoS LOIT (6:56 PM – 7:16 PM), Port Scan (sS, sT, sF, sX, sN, sP, sV, sU, sO, sA, sW, sR, sL and B)

5.2 Nitroba University Harassment Scenario

A "Nitroba University Harassment Scenario" is a hypothetical network forensic scenario created by the "Digital Corpora". The scenario consists of the slides that introduce the problem (PDF, PPT or TXT file), screenshots in PNG format as a part of the problem introduction, and a PCAP file of the captured traffic. There is also available password-protected solution of this scenario [170].

The background of this case is the harassment of the teacher Lily Tuckrige (lilytuckrige@yahoo.com). She thinks that harassing emails are from one of her students (Amy Smith, Burt Greedom, Tuck Gorge, Ava Book, Johnny Coach, Jeremy Ledvkin, Nancy Colburne, Tamara Perkins, Esther Pringle, Asar Misrad, Jenny Kan). The provided information contains screenshots of the harrasing emails (including the email header), the IP from the email (140.247.62.34) that points into *34.62.247.140.in-addr.arpa domain name pointer G24.student.nitroba.org*, this Nitroba dorm room has wifi without password and three women live here (Alice, Barbara, Candice). The PCAP capture file contains traffic from the packet sniffer placed on the ethernet port. The goal of this scenario is to determine who is responsible for the harassing emails [170].

Annotation of the PCAP file

- **Number of packets:** 94 410
- **Timeline:** 2008-07-22 01:51:07.095278 UTC – 2008-07-22 06:13:47.046029 UTC
- **Involved hosts:** 192.168.15.4 (attacker), 140.247.62.34 (attacker), 209.73.187.220 (answers.yahoo.com), 74.125.19.104 (www.google.com), 74.125.19.17 (mail.google.com), 69.80.225.91 (www.sendanonymousemail.net), 65.54.186.77 (login.live.com)
- **Protocols:** YMSG, XMPP/XML, UDPENCAP, UDP, TLSv1, TCP, SSLv3, SSLv2, SSL, SSDP, SIP/SDP, SIP, RTP, RTCP, RIPv2, RIPv1, OCSP, NTP, Messenger, MSNMS, LLC, ISAKMP, IGMPv3, IGMPv2, ICMP, HTTP/XML, HTTP, ESP, DNS, DCERPC, ARP

Case theory – report of the scenario

Gmail user jcoachj@gmail.com logged into the gmail on 22.7.2008 06:01:02 UTC on the computer with IP 192.168.15.4 with operating system Apple iOS. Using the same web browser, the user searched for "how to annoy people", "sending anonymous mail" and "I want to harass my teacher" on Google approximately about 05:57 on 22.7.2008. Then the user search for „can I go to jail for harassing my teacher" on 22.7.2008 05:58.

After that, at 05:59, there was a login on mail.live.com. At 06:00, the mail.google.com was visited by the user jcoachj@gmail.com (used *gmailchat* cookie) witch proves that this user used this computer (IP: 192.168.15.4).

At 06:01, the user visited www.sentanonymousemail.net. Then the user sent two emails using anonymous mail delivery. The first one was sent using www.sentanonymousemail.net on 22.7.2008 06:02:57 UTC. The second one was sent using willselfdestruct.com on 22.7.2008 06:04:24 UTC. After that, the user searched for "where do the cool kids go to play" on Google and visited youtube.com. These actions prove that the Johnny Coach is the person who harassed his teacher Lily Tuckrige.

On 22.7.2008 06:09:59 UTC, the user amy789smith authenticated with Yahoo, but there was used different web browser, and therefore Amy Smith did not send the harassment emails.

5.3 NETRESEC Packet Injection Attacks

Erik Hjelmvik, in his article "Packet Injection Attacks in the Wild," focused on the packet injection attacks that have been running for several months and that was still active in 2016. They attempted to recreate these packet injections and provided PCAP files [177].

The first attack that they recreated was against the *www.02995.com*. It belongs to the "hao" group of the original research "Website-Targeted False Content Injection by Network Operators". The second attack was against the *id1.cn*. This injection attack was based on the BroCon 2015 [177].

The details of the performed attacks are described in the following sections, including annotations of the provided PCAP files.

5.3.1 Packet injection attack against *www.02995.com*

After visiting the website *www.02995.com*, the two responses are generated with the same sequence number (3820080905):

1. "302 Found" — redirect to *http://www.hao123.com/?tn=93803173_s_hao_pg* injected packet; uses only LF as line feed in the HTTP header,
2. "302 Moved Temporarily" — redirect to *http://hao.360.cn/?src=lm&ls=n4a2f6f3a91* real webserver response; uses the standard CR-LF line breaks in the HTTP response

The user is redirected to the *http://www.hao123.com/*, because the injected response arrived before the real webserver response [177].

Annotation of the PCAP file

- **Number of packets:** 202
- **Timeline:** 2016-03-01 08:03:47.560150 UTC – 2016-03-01 08:04:10.149843 UTC

- **Involved hosts:** 103.235.46.234 (www.hao123.com), 122.225.98.197 (www.02995.com), 192.168.1.254 (Windows)
- **Protocols:** HTTP, TCP

5.3.2 Packet injection attack against *id1.cn*

After visiting the website *id1.cn*, three responses are returned:

1. "200 OK" — redirect to *http://id1.cn/rd.s/Btc5n4unOP4UrIfE?url=http://id1.cn/*, real webserver response, client proceeds this website (this is the first response) and gets two injected responses and one real website response:
   (a) "403 Forbidden" — redirect to *http://batit.aliyun.com/alww.html*
   (b) "403 Forbidden" — redirect to *http://batit.aliyun.com/alww.html*
   (c) "200 OK" — redirect to *http://id1.cn/*, real website response
2. "403 Forbidden" — redirect to *http://batit.aliyun.com/alww.html*, injected response,
3. "403 Forbidden" — redirect to *http://batit.aliyun.com/alww.html*, injected response.

Annotation of the PCAP file

- **Number of packets:** 155
- **Timeline:** 2016-03-01 08:00:19.058801 UTC – 2016-03-01 08:00:28.839398 UTC
- **Involved hosts:** 42.96.141.35 (id1.cn), 42.120.158.95 (batit.aliyun.com), 192.168.1.254 (Windows)
- **Protocols:** HTTP, TCP

5.4 ICS Cybersecurity — DoS Attacks against SCADA-based systems

The ICS Cybersecurity PCAP repository is a suite of PCAP captures that includes the "modbus TCP SCADA" dataset created by a team from the University of Coimbra (Portugal), as a part of the ATENA H2020 project. This dataset was generated for the article "Denial of Service Attacks: Detecting the frailties of machine learning algorithms in the Classification Process" using MODBUS/TCP equipment in the SCADA system [174, 176].

The captured data is organized into three folders containing sub-folders based on the type of the attack, including ARP-based, Main-in-the-Middle attack, Modbus query flooding, ICMP flooding, and TCP SYN flooding. In addition, a nominal state with no attack is included. There is a naming convention for the PCAP files — *<capture interface>dump-<attack>-<attack subtype>-<attack duration>-<capture duration>*. Each attack starts 5 minutes after the first captured packet. The PCAP files with 12 hour capture duration are excluded, this project includes only 0.5 h, 1 h, 6 h, and 12 h captures. The individual PCAP files are described in detail in appendix A. This appendix provides lists all PCAPs of this dataset (12 h captures excluded), including the number of packets, timeline, and attack times. The brief overall description for each category is provided in the following sections.

The time is in the UTC format, and flooding attacks hosts do not contain all involved host IP addresses (since many third-party IPs are involved in the DDoS attacks) [174].

The table 5.5 displays the network information about this dataset.

Network Information	
Attackers	172.27.224.50, 172.27.224.80
Victim	172.27.224.11, 172.27.224.70, 172.27.224.250, 172.27.224.251

Table 5.5: Network information of ICS Cybersecurity dataset

5.4.1 Nominal state

- **Number of PCAPs:** 3
- **Total number of packets:** 535 422
- **Total timeline:** 2018-08-23 17:40:48.376131 UTC – 2018-09-09 00:14:03.946853 UTC
- **Total involved hosts:** 172.27.224.11, 172.27.224.70, 172.27.224.250, 172.27.224.251
- **Total protocols:** ARP, BJNP, BROWSER, DHCP, DHCPv6, ICMPv6, IGMPv3, LLMNR, Modbus/TCP, RARP, STP, TCP, UDP
- **Total attacks:** none

5.4.2 ARP-based, Man-in-the-Middle attack

- **Number of PCAPs:** 22
- **Total number of packets:** 4 161 258
- **Total timeline:** 2018-08-23 18:57:01.789547 UTC – 2018-09-04 02:08:55.041070 UTC
- **Total involved hosts:** 172.27.224.70 (00:0c:29:9d:9e:9e), 172.27.224.80 (00:0c:29:e6:14:0d), 172.27.224.250 (00:80:f4:09:51:3b), 172.27.224.251 (48:5b:39:64:40:79)
- **Total protocols:** ARP, BROWSER, DHCP, DHCPv6, ICMP, ICMPv6, IGMPv3, IRC, LLMNR, Modbus/TCP, STP, TCP, UDP
- **Total attacks:** mitm-change, mitm-read

5.4.3 Modbus query flooding

- **Number of PCAPs:** 44
- **Total number of packets:** 19 861 222
- **Total timeline:** 2018-05-22 10:00:59.923334 UTC – 2018-08-14 17:52:17.836726 UTC
- **Total involved hosts:** 172.27.224.50 (Source), 172.27.224.70 (Source), 172.27.224.80 (Source), 172.27.224.250 (Destination)

- **Total protocols:** ALLJOYN-NS, AMQP, AMS, ARP, ASAP, ASF, ATMTCP, AX4000, BACnet-APDU, BEEP, BFD Control, BJNP, BROWSER, BitTorrent, CAPWAP-Control, CLASSIC-STUN, Chargen, DAYTIME, DB-LSP, DCERPC, DHCP, DHCPv6, DIAMETER, DISTCC , DNS, DRDA, ECATF, ECHO, ECMP, ELCOM, EPM, Elasticsearch, FTP, GIOP, Gearman, Gnutella, HTTP, HTTP/XML, HTTP2, IAX2, ICMP, ICMPv6, ICP, IGMPv3, IPA, IPDC, IPSICTL, IPv4, IRC, ISAKMP, ISystemActivator, KPASSWD, KRB4, L2TP, LANMAN, LLMNR, MAC-Telnet, MDNS, MEMCACHE, MIH, Messenger, Modbus/TCP, NAT-PMP, NBNS, NBSS, NDMP, NTP, Nano Bootstrap, Netsync, OMAPI, Omni-Path, OpenVPN, PMPROXY, PTP/IP, Portmap, R3, RADIUS, REMACT, RIPv1, RIPv2, RSIP, RTSP, RX, SABP, SAMR, SIP, SMB, SMB Mailslot, SMB Pipe, SMB2, SNMP, SRVLOC, SSDP, SSH, SSL, SSLv3, STP, TCP, TFP over TCP, TFTP, TLSv1, TLSv1.1, TLSv1.2, TPKT, UDP, VNC, WOW, X11, XDMCP, synergy
- **Total attacks:** Modbus query flooding

5.4.4 ICMP flooding

- **Number of PCAPs:** 33
- **Total number of packets:** 10 092 395
- **Total timeline:** 2018-05-21 09:54:56.811018 UTC – 2018-08-09 00:57:02.800427 UTC
- **Total involved hosts:** 172.27.224.250 (Destination), many more IP addresses from the full range of IP addresses (Source)
- **Total protocols:** ARP, BJNP, BROWSER, DHCP, DHCPv6, ICMP, ICMPv6, IGMPv3, IRC, LLMNR, Modbus/TCP, RARP, STP, TCP, UDP
- **Total attacks:** ICMP flooding

5.4.5 TCP SYN flooding

- **Number of PCAPs:** 33
- **Total number of packets:** 6 849 825
- **Total timeline:** 2018-05-21 15:32:57.012079 UTC – 2018-08-06 17:44:39.865941 UTC
- **Total involved hosts:** 172.27.224.250 (Destination), many more IP addresses from the full range of IP addresses (Source)
- **Total protocols:** 104apci, ALLJOYN-NS, AMQP, AMS, ANSI C12.22, ARP, ASAP, ASF, ATMTCP, AX4000, BACnet-APDU, BEEP, BFD Control, BJNP, BROWSER, BitTorrent, CAPWAP-Control, CLASSIC-STUN, CVSPSERVER, Chargen, DAYTIME, DB-LSP, DCERPC, DHCP, DHCPv6, DIAMETER, DISTCC , DNS, DRDA, ECATF, ECHO, ECMP, ELCOM, Elasticsearch, FTP, GIOP, GTPv2, Gearman, Gnutella, HTTP, HTTP/XML, HTTP2, IAX2, ICAP, ICMP, ICMPv6, ICP, IGMPv3, IPA, IPDC, IPSICTL, IPv4, IRC, ISAKMP, ISystemActivator, KNET, KPASSWD, KRB4, L2TP, LANMAN, LLMNR, MAC-Telnet, MDNS, MEMCACHE, MIH, MQTT, MSNMS, Modbus/TCP, NAT-PMP, NDMP, NTP, Nano Bootstrap, Netsync, OMAPI,

Omni-Path, OpenVPN, PMPROXY, PPTP, PTP/IP, Portmap, R3, RADIUS, RIPv1, RIPv2, RMI, RSIP, RTSP, RX, SABP, SAMR, SIP, SMB, SMB Pipe, SMB2, SNMP, SRVLOC, SSDP, SSH, SSHv2, SSL, SSLv3, STP, Socks, TCP, TFP over TCP, TFTP, TLSv1, TLSv1.1, TLSv1.2, TPKT, UDP, VICP, VNC, WOW, X11, XDMCP, ZEBRA, giFT, kismet, synergy

- **Total attacks:** TCP SYN flooding

5.5 WireShark SampleCaptures

WireShark provides many PCAP capture files in its wiki page. Some packet capture files are described in the following sections [175].

5.5.1 SSL with decryption keys

Wireshark provides a list of PCAP files together with the decryption keys. Some PCAPs from the list are described in the following part of this section. The description and source of the PCAP file is retrieved from the Wireshark wiki page [175].

rsasnakeoil.cap

- **Description:** SSL encrypted HTTPS traffic, example taken from the dev mailinglist, RSA key available
- **Number of packets:** 58
- **Timeline:** 2006-08-24 09:04:15.842911 UTC – 2006-08-24 09:04:28.211338 UTC
- **Involved hosts:** 127.0.0.1
- **Protocols:** HTTP, SSLv2, SSLv3, TCP

dump.pcapng

- **Description:** a openssl's s_client/s_server HTTP GET request over TLSv1.2 with 73 different cipher suites, generated using the openssl-connect
- **Number of packets:** 1 095
- **Timeline:** 2013-09-15 21:52:16.72595 UTC – 2013-09-15 21:52:17.696039 UTC
- **Involved hosts:** 127.0.0.1
- **Protocols:** SSLv2, TCP, TLSv1.2

mysql-ssl.pcapng

- **Description:** MySQL over TLSv1, PCAP from Peter Wu's Wireshark-notes, pre-master keys are available in capture comments; server with MariaDB, database testdb, queries (INSERT, SELECT, deliberate disallowed 'USE mysql' and more) [181, 182]
- **Number of packets:** 59

- **Timeline:** 2015-01-29 10:39:58.578402281 UTC – 2015-01-29 10:40:33.092194163 UTC
- **Involved hosts:** 127.0.0.1
- **Protocols:** MySQL, TCP, TLSv1.2

pop-ssl.pcapng

- **Description:** POP, PCAP from Peter Wu's Wireshark-notes, pre-master keys are available in capture comments; after handshake, "POPA" followed by renegotiation, "POPA" and "QUIT" [181, 183]
- **Number of packets:** 38
- **Timeline:** 2015-01-30 14:49:01.890849939 UTC – 2015-01-30 14:49:13.645704037 UTC
- **Involved hosts:** 127.0.0.1
- **Protocols:** POP, TCP, TLSv1.2

smtp-ssl.pcapng

- **Description:** SMTP, PCAP from Peter Wu's Wireshark-notes, pre-master keys are available in capture comments; "EHLO lekensteyn" was typed and triggered a renegiotation with "R" (which resulted in an error) [181, 184]
- **Number of packets:** 38
- **Timeline:** 2015-01-30 11:31:42.005931161 UTC – 2015-01-30 11:32:41.025768841 UTC
- **Involved hosts:** 127.0.0.1
- **Protocols:** SMTP, TCP, TLSv1.2

5.6 Summary and comparison

This chapter described some available datasets in detail. This datasets survey is also publicly available at https://martinazembjakova.github.io/Network-forensic-tools-taxonomy/ using GitHub pages environment as one of this master's thesis outputs. The design of this website can be seen in appendix C.

The provided datasets differ in the type of captured traffic, the number of packets, and the capture time.

The main issue that differs in the described datasets is the type of the captured data. DoS attacks are included in the CIC-DDoS2019 dataset and "ICS DoS Attacks" dataset. The other use cases include IDS data in CIC-IDS2017 dataset, harassment in the "Nitroba University Harassment" dataset, packet injection attacks by NETRESEC, and SSL network traffic available from Wireshark sample captures wiki page. The summary of appeared attacks in the described datasets can be seen in table 5.6. Table 5.7 contains appeared application protocols according to TCP/IP model. The tables also contains references to sections.

Dataset	Attacks
CIC-DDoS2019 (section 5.1.1)	DoS
First day	PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN
Second day	NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, TFTP
CIC-IDS2017 (section 5.1.2)	
Tuesday	Brute Force, FTP-Patator, SSH-Patator
Wednesday	DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, Heartbleed Port 444
Thursday	Web attacks - Brute Force, XSS, Sql Injection, Infiltration attacks - Dropbox download Win Vista, Cool disk MAC
Friday	Botnet ARES, DDoS LOIT, Port Scan (sS, sT, sF, sX, sN, sP, sV, sU, sO, sA, sW, sR, sL and B)
Nitroba University Harassment Scenario (section 5.2)	harassement emails
NETRESEC Packet Injection Attacks (section 5.3)	Packet Injection attacks
ICS Cybersecurity - DoS Attacks against SCADA-based systems (section 5.4)	DoS
ARP-based, MITM attack (section 5.4.2)	mitm-change, mitm-read
Modbus query flooding (section 5.4.3)	Modbus query flooding
ICMP flooding (section 5.4.4)	ICMP flooding
TCP SYN flooding (section 5.4.5)	TCP SYN flooding

Table 5.6: Summary of attacks of datasets

Secondly, the other difference is the number of packets in the provided datasets. The Canadian Institute for Cybersecurity datasets consists of a huge amount of packets. The CIC-DDos2019 dataset contains together 312 191 170 packets for both capturing days, the CIC-IDS2017 contains 56 370 702 packets for all capturing days. Another large dataset is the "ICS DoS Attacks" dataset with 41 500 122 packets. On the other hand, other described datasets consist of many fewer packets — the "Nitroba University Harassment Scenario" dataset contains 94 410 packets, all WireShark sample captures are less than 10 000 packets, and NETRESEC Packet Injection Attacks are about 200 packets.

Furthermore, the number of packets is also related to the length of data capture. Some datasets include data captured for days, others for just hours or minutes. There is also a difference in the time period when the data was captured. Table 5.8 summarizes the captured time intervals for each described dataset, including the references of the sections that describe these datasets and approximate capture duration (delta timeline). Time is in the UTC format.

The datasets also differ in how the data are stored. The CIC-IDS2017 dataset provides only one PCAP file for the whole day captured data. The other huge datasets (CIC-DDos2019 and "ICS DoS Attacks") split the data into more PCAP files.

Dataset	Application protocols
CIC-DDoS2019 (section 5.1.1)	3Com XNS, 3GPP2 A11, A21, ADP, ALC, ALLJOYN-ARDP, ALLJOYN-NS, AMS, AMT, ANSI C12.22, AODV, ASAP, ASTERIX, ATH, AX4000, AYIYA, Auto-RP, BACnet-APDU, BAT_BATMAN, BAT_GW, BAT_VIS, BFD Control, BJNP, BOOTP, BROWSER, BVLC, Bundle, CAPWAP-Control, CAPWAP-Data, CDP, CLDAP, CLNP, CN/IP, CUPS, CoAP, DAYTIME, DB-LSP-DISC, DCC, DCERPC, DCP-AF, DCP-PFT, DHCP, DHCPv6, DIS, DMP, DNP, DNS, DPNET, DPP, DSR, DTLS, DTP, EAP, EAPOL, ECAT, ECATF, ECHO, ECMP, EGD, EIGRP, ENIP, ENRP, ESP, Elasticsearch, GPRS-NS, GSM SIM, GSMTAP, GTP, Geneve, HART_IP, HCrt, HPEXT, HTTP, HTTP/XML, HiQnet, IAPP, IAX2, ICP, IO-RAW, IPVS, IPX, ISAKMP, ISO, KDP, KINK, KNET, KPASSWD, KRB4, KRB5, L2TP, L2TPv3, LDP, LISP, LLC, LLDP, LLMNR, LMP, LTP Segment, LWAPP, MANOLITO, MDNS, MEMCACHE, MIH, MIPv6, MNDP, MPLS, MSMMS, MSproxy, MiNT, MobileIP, Modbus/UDP, NAT-PMP, NBDS, NBNS, NCP, NHRP, NTP, NXP 802.15.4 SNIFFER, Nano, OCSP, OLSR v1, OSPF, OpenVPN, PCP v1, PCP v2, PFCP, PKTC, PNIO, POWERLINK/UDP, PTPv2, Pathport, QUAKE, QUAKE2, QUAKE3, QUAKEWORLD, RADIUS, RDT, RIP, RIPng, RIPv1, RIPv2, RRoCE, RSIP, RSVP, RTCP, RTPproxy, RakNet, SABP, SAP, SAP/SDP, SCoP, SEBEK, SIP, SNA, SNMP, SRVLOC, SSDP, SSH, SSHv2, SSL, SliMP3, Syslog, TAPA, TC-NV, TETRA, TFTP, TIME, TIPC, TLSv1, TLSv1.2, TLSv1.3, TPCP, TPKT, TS2, TSP, TZSP, UAUDP, UDP/MIKEY, ULP, UNKNOWN, VITA 49, Vines IP, Vuze-DHT, VxLAN, WHO, WLCCP, WSP, WTLS+WSP, WTLS+WTP+WSP, WTP+WSP, X.25, XTACACS, XYPLEX, collectd, eDonkey, lw_res, openSAFETY over UDP, packetbb
First day	AH, LBT-RU, Portmap, SSLv2, STP, STUN
Second day	ALC/XML, ALLJOYN, ASF, Armagetronad, BSSGP, CAT-TP, DSPv2, EAPOL-MKA, FF, GPRS-LLC, GTPv2, MAC-Telnet, NEMO, NetBIOS, OSCORE, PN-PTCP, RMCP, RX, SPX, STP, TACACS, Thread, ZigBee IP, openSAFETY/UDP
CIC-IDS2017 (section 5.1.2)	BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, EMP, FTP, FTP-DATA, HTTP, HTTP/XML, KRB5, LANMAN, LDAP, LLDP, LLMNR, LSARPC, MDNS, NBNS, NBSS, NTP, OCSP, RPC_NETLOGON, SMB, SMB2, SRVSVC, SSDP, SSH, SSHv2, SSL, SSLv2, SSLv3, STUN, TLSv1, TLSv1.1, TLSv1.2,
Monday	Elasticsearch, GQUIC, MP4, OpcUa, PKIX-CRL, SAMR, TLSv1.3, WebSocket
Tuesday	BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, MP4, PKIX-CRL, SAMR
Wednesday	BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, DTLS, DTLSv1.2, MP4, MPEG, SAMR, TLSv1.3, WebSocket
Thursday	BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, MP4, MPEG PES, PKIX-CRL, TC
Friday	BJNP, BROWSER, CDP, CLDAP, DCERPC, DHCPv6, DNS, DRSUAPI, H1, MPEG PES, OMAPI, PKIX-CRL, SAMR, SCTP, TLSv1.3
Nitroba University Harassment Scenario (section 5.2)	DCERPC, DNS, HTTP, HTTP/XML, ISAKMP, MSNMS, Messenger, NTP, OCSP, RIPv1, RIPv2, RTCP, RTP, SIP, SIP/SDP, SSDP, SSL, SSLv2, SSLv3, TLSv1, XMPP/XML, YMSG
NETRESEC Packet Injection Attacks (section 5.3)	HTTP
ICS Cybersecurity - DoS Attacks against SCADA-based systems (section 5.4)	BROWSER, DHCP, DHCPv6, LLMNR, Modbus/TCP
Nominal state (section 5.4.1)	BJNP
ARP-based, MITM attack (section 5.4.2)	IRC
Modbus query flooding (section 5.4.3)	ALLJOYN-NS, AMQP, AMS, ASAP, ASF, ATMTCP, AX4000, BACnet-APDU, BEEP, BFD Control, BJNP, BitTorrent, CAPWAP-Control, CLASSIC-STUN, Chargen, DAYTIME, DB-LSP, DCERPC, DIAMETER, DISTCC, DNS, DRDA, ECATF, ECHO, ECMP, ELCOM, EPM, Elasticsearch, FTP, GIOP, Gearman, Gnutella, HTTP, HTTP/XML, HTTP2, IAX2, ICP, IPA, IPDC, IPSICTL, IRC, ISAKMP, ISystemActivator, KPASSWD, KRB4, L2TP, LANMAN, MAC-Telnet, MDNS, MEMCACHE, MIH, Messenger, NAT-PMP, NBNS, NBSS, NDMP, NTP, Nano Bootstrap, Netsync, OMAPI, Omni-Path, OpenVPN, PMPROXY, PTP/IP, Portmap, R3, RADIUS, REMACT, RIPv1, RIPv2, RSIP, RTSP, RX, SABP, SAMR, SIP, SMB, SMB Mailslot, SMB Pipe, SMB2, SNMP, SRVLOC, SSDP, SSH, SSL, SSLv3, STP, TFP over TCP, TFTP, TLSv1, TLSv1.1, TLSv1.2, TPKT, VNC, WOW, X11, XDMCP, synergy
ICMP flooding (section 5.4.4)	BJNP, IRC, STP
TCP SYN flooding (section 5.4.5)	104apci, ALLJOYN-NS, AMQP, AMS, ANSI C12.22, ASAP, ASF, ATMTCP, AX4000, BACnet-APDU, BEEP, BFD Control, BJNP, BitTorrent, CAPWAP-Control, CLASSIC-STUN, CVSPSERVER, Chargen, DAYTIME, DB-LSP, DCERPC, DIAMETER, DISTCC, DNS, DRDA, ECATF, ECHO, ECMP, ELCOM, Elasticsearch, FTP, GIOP, GTPv2, Gearman, Gnutella, HTTP, HTTP/XML, HTTP2, IAX2, ICAP, ICP, IPA, IPDC, IPSICTL, IRC, ISAKMP, ISystemActivator, KNET, KPASSWD, KRB4, L2TP, LANMAN, MAC-Telnet, MDNS, MEMCACHE, MIH, MQTT, MSNMS, NAT-PMP, NDMP, NTP, Nano Bootstrap, Netsync, OMAPI, Omni-Path, OpenVPN, PMPROXY, PPTP, PTP/IP, Portmap, R3, RADIUS, RIPv1, RIPv2, RMI, RSIP, RTSP, RX, SABP, SAMR, SIP, SMB, SMB Pipe, SMB2, SNMP, SRVLOC, SSDP, SSH, SSHv2, SSL, SSLv3, STP, Socks, TFP over TCP, TFTP, TLSv1, TLSv1.1, TLSv1.2, TPKT, VICP, VNC, WOW, X11, XDMCP, ZEBRA, giFT, kismet, synergy
WireShark SampleCaptures SSL (section 5.5.1)	
rsasnakeoil.cap	HTTP, SSLv2, SSLv3
dump.pcapng	SSLv2, TLSv1.2
mysql-ssl.pcapng	MySQL, TLSv1.2
pop-ssl.pcapng	POP, TLSv1.2
smtp-ssl.pcapng	SMTP, TLSv1.2

Table 5.7: Summary of application protocols of datasets

Dataset	Timeline	Duration
CIC-DDoS2019 (section 5.1.1)		
First day	2018-11-03 12:18:16 - 2018-11-03 20:36:56	8 hours
Second day	2018-12-01 13:17:10 - 2018-12-01 21:16:39	8 hours
CIC-IDS2017 (section 5.1.2)		
Monday	2017-07-03 11:55:58 - 2017-07-03 20:01:34	8 hours
Tuesday	2017-07-04 11:53:32 - 2017-07-04 20:00:31	8 hours
Wednesday	2017-07-05 11:42:42 - 2017-07-05 20:10:19	8 hours
Thursday	2017-07-06 11:58:58 - 2017-07-06 20:04:44	8 hours
Friday	2017-07-07 11:59:39 - 2017-07-07 20:02:41	8 hours
Nitroba University Harassment Scenario (section 5.2)	2008-07-22 01:51:07 - 2008-07-22 06:13:47	4 hours
NETRESEC Packet Injection Attacks (section 5.3)		
against www.02995.com	2016-03-01 08:03:47 - 2016-03-01 08:04:10	23 seconds
against id1.cn	2016-03-01 08:00:19 - 2016-03-01 08:00:28	10 seconds
ICS Cybersecurity - DoS Attacks against SCADA-based systems (section 5.4)		
Nominal state (section 5.4.1)	2018-08-23 17:40:48 - 2018-09-09 00:14:03	16 days
ARP-based, MITM attack (section 5.4.2)	2018-08-23 18:57:01 - 2018-09-04 02:08:55	11 days
Modbus query flooding (section 5.4.3)	2018-05-22 10:00:59 - 2018-08-14 17:52:17	84 days
ICMP flooding (section 5.4.4)	2018-05-21 09:54:56 - 2018-08-09 00:57:02	79 days
TCP SYN flooding (section 5.4.5)	2018-05-21 15:32:57 - 2018-08-06 17:44:39	77 days
WireShark SampleCaptures SSL (section 5.5.1)		
rsasnakeoil.cap	2006-08-24 09:04:15 - 2006-08-24 09:04:28	13 seconds
dump.pcapng	2013-09-15 21:52:16 - 2013-09-15 21:52:17	1 second
mysql-ssl.pcapng	2015-01-29 10:39:58 - 2015-01-29 10:40:33	35 seconds
pop-ssl.pcapng	2015-01-30 14:49:01 - 2015-01-30 14:49:13	12 seconds
smtp-ssl.pcapng	2015-01-30 11:31:42 - 2015-01-30 11:32:41	1 minute

Table 5.8: Summary of time intervals of datasets

Chapter 6

Use cases and demonstration

This chapter describes the common use cases for forensic tools based on the surveys in chapter 3 and chapter 4.

Some of these use cases are demonstrated on the obtained datasets described in chapter 5 and on newly created datasets described in chapter 7.

Firstly, the complex investigation is described together with the possible tools used. It is demonstrated on the Nitroba harassment scenario described in chapter 5. Secondly, some individual use cases are described. In these use cases, some tools are described in more details with using previously described datasets in chapter 5 or newly created datasets described in chapter 7. The use cases such as scanners, sniffers, visualizers, analyzers, network diagnostic tools, IDS/IPS, and SIEMs are discussed. The end of this chapter contains summary and comparison to the use cases from survey of existing taxonomies described in chapter 3.

6.1 Nitroba scenario

The Nitroba scenario was already described in chapter 5 as an existing dataset. This section focuses on the tools that can be used during the investigation.

The investigation of this case can be separated into the following stages based on the OSCAR investigative methodology described in chapter 2:

1. obtaining basic information,
2. strategizing the way of collecting evidence,
3. collecting evidence,
4. analyzing collected evidence,
   (a) obtaining basic information about PCAP,
   (b) identifying important conversations,
   (c) analyzing specified flows,
5. creating report.

The investigation process is described in detail in the following sections, including obtaining necessary information, collecting evidence, and possible ways how to analyze the data. The report of the Nitroba scenario is described in detail in section 5.2.

6.1.1 Obtain and strategize

The basic information is provided by the teacher that reported this incident to IT support:

- *Name of the victim (teacher):* Lily Tuckrige
- *Background of the victim (teacher):* She teaches chemistry CHEM109 that summer at NSU.
- *Incident:* Lily Tuckrige received a harassing email at her personal email address lilytuckrige@yahoo.com
- *Suspect:* She thinks that the email is from one of the students in her class (Amy Smith, Burt Greedom, Tuck Gorge, Ava Book, Johnny Coach, Jeremy Ledvkin, Nancy Colburne, Tamara Perkins, Esther Pringle, Asar Misrad, Jenny Kant).
- *Provided evidence:* a screenshot of the harassing email message

The provided evidence is not enough for the investigators. Since this screenshot does not contain important information, they ask the teacher to provide them with the screenshot of the email, including email headers. It is possible to display the email header in all mail clients; for example, the Gmail web client provides displaying and downloading the original message, including the header.

From the completed screenshot of the email, including the header, under the "Received" part of the header, the sender's IP is seen. To see where this IP address points, the command *host* is used. The result is that the IP 140.247.62.34 points to a nitroba dorm room *G24.student.nitroba.org*.

Obtaining information about the environment is not finished. The investigators require a list of students who live in the G24 room, the type of internet used in this room, and information about the protection. The answers are that Alice, Barbara, and Candice live in this room, the 10 Mbps Ethernet is in all Nitroba rooms, and Nitroba provides no Wi-Fi access, but the G24 room contains a Wi-Fi router with no password.

The investigators decided to capture the network traffic from the G24 room and wait until a new harassing email is sent.

6.1.2 Collect

When the investigators have the necessary information about the case, they can collect the evidence. They use the packet sniffer to catch network traffic in the G24 room. Many packet sniffers can be used, eight hardware or software-based. The hardware-based needs to be manually installed into the network infrastructure, and software-based can be easily applied on the concrete interface or more interfaces. As a packet sniffer, there can be used, for example, Wireshark, tcpdump, NetworkMiner, and others. The detailed comparison of different packet sniffers is provided in the following section 6.3 that focuses on capturing the data use case.

The evidence is collected until a new harassing email is sent to the teacher. The teacher provides the screenshot of the email, that has subject "you can't find us" and is sent via the www.willselfdestruct.com.

6.1.3 Analyze

Now when the evidence is collected, the investigators need to analyze the collected network traffic data. There are many tools known as packet analyzers that can be used. The most commonly used tool is Wireshark, which can provide the investigator with much useful information. In this case, the NetworkMiner is also very helpful since it provides detailed information about involved hosts, messages, cookies, and many more. It is also described how the data can be analyzed using the concrete packet analyzer tools, including the information that can be gained using the tool in this scenario.

Wireshark

Wireshark can provide the basic information about the PCAP file, such as the number of packets, the timeline of the captured data, and used services (protocols).

Wireshark can be used to identify the attacker's IP. It can be done by searching the packets for the email content, for example, the subject "you can't find us". The result can be seen in the following figure 6.1. The attacker's IP is 192.168.15.4.

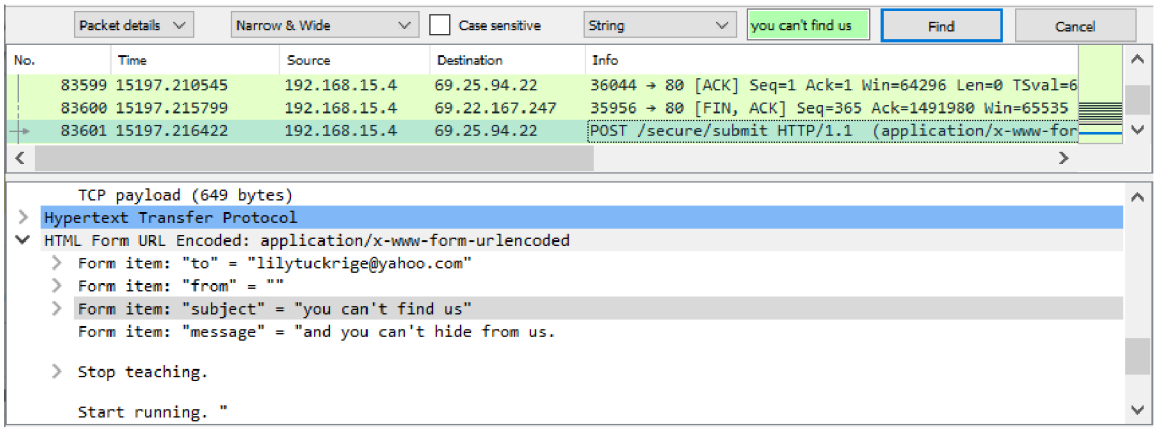


Figure 6.1: Message with attacker's IP gained using *Wireshark* tool

The TCP or HTTP stream can be visualized using "Follow the TCP/HTTP stream." For example, the stream of the harassing email with subject "you can't find us" is displayed in figure 6.2. A user agent can also be interesting for the investigator. In *Wireshark*, the user agent can be obtained from the HTTP header when clicking on the packet or from the TCP/HTTP stream.

The feature of following TCP/HTTP steams can also be used for analyzing meaningful HTTP conversations obtained, for example from the *tshark* tool. *Wireshark* displays the whole stream, and then it is easy to determine the purpose of that HTTP conversation.

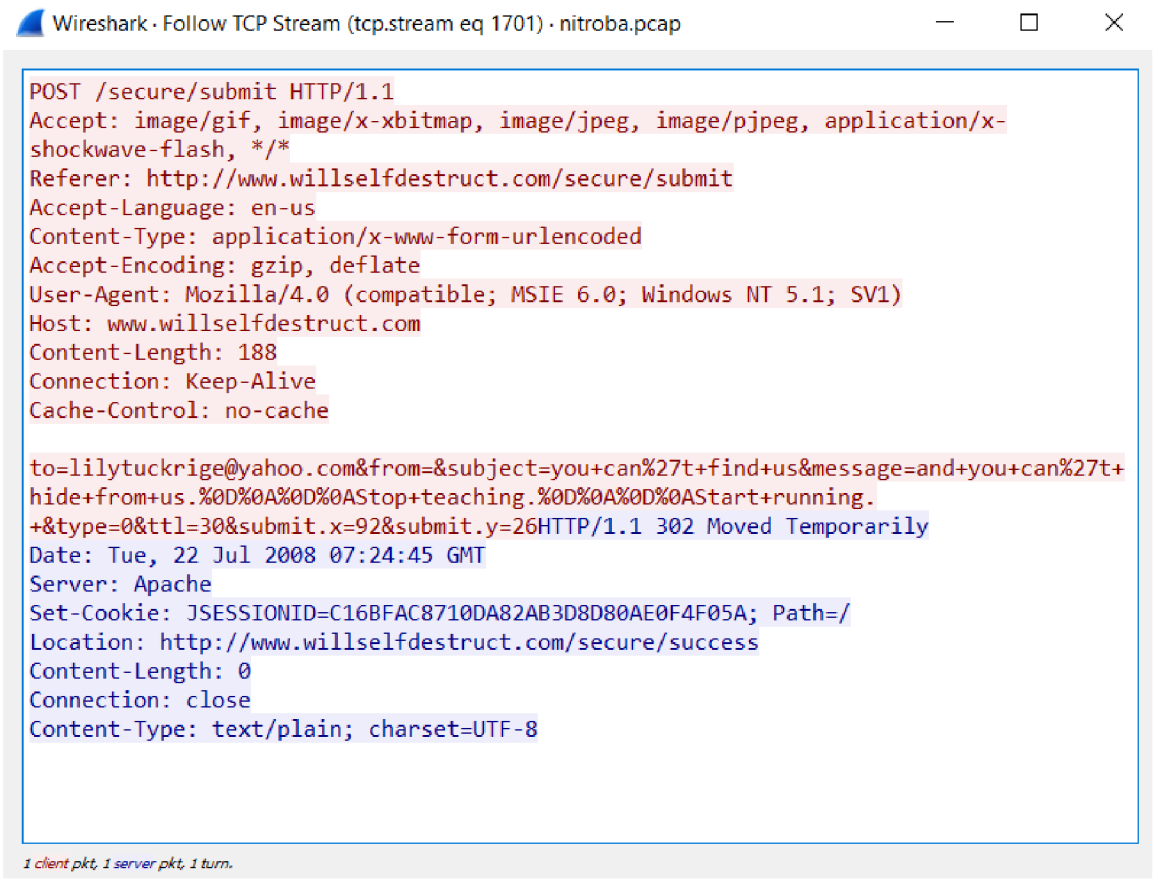
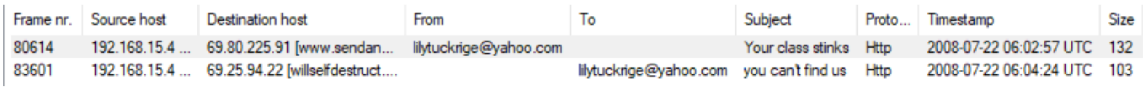


Figure 6.2: TCP stream (1701) of harassing email gained using *Wireshark* tool

NetworkMiner

NetworkMiner can display email messages in one click from the menu bar *Messages*, and from these messages, the source IP can be identified. Visualized messages can be seen in figure 6.3. Compared to the *Wireshark*, obtaining an attacker's IP is easier.



Frame nr.	Source host	Destination host	From	To	Subject	Proto...	Timestamp	Size
80614	192.168.15.4 ...	69.80.225.91 [www.sendan...	lilytuckrige@yahoo.com		Your class stinks	Http	2008-07-22 06:02:57 UTC	132
83601	192.168.15.4 ...	69.25.94.22 [willselfdestruct....		lilytuckrige@yahoo.com	you can't find us	Http	2008-07-22 06:04:24 UTC	103

Figure 6.3: Message with attacker's IP gained using *NetworkMiner* tool

NetworkMiner is a powerful tool that can analyze the PCAP file and display all involved hosts, including detailed information such as assigned MAC addresses, NIC Vendor, MAC Age, hostname, operating system, sent and received packets statistics, incoming and outgoing sessions, queried IP Addresses, queried DNS names, web browser user agents, device category and other host details. The details about the attacker's IP obtained using NetworkMiner are displayed in the following figure 6.4.

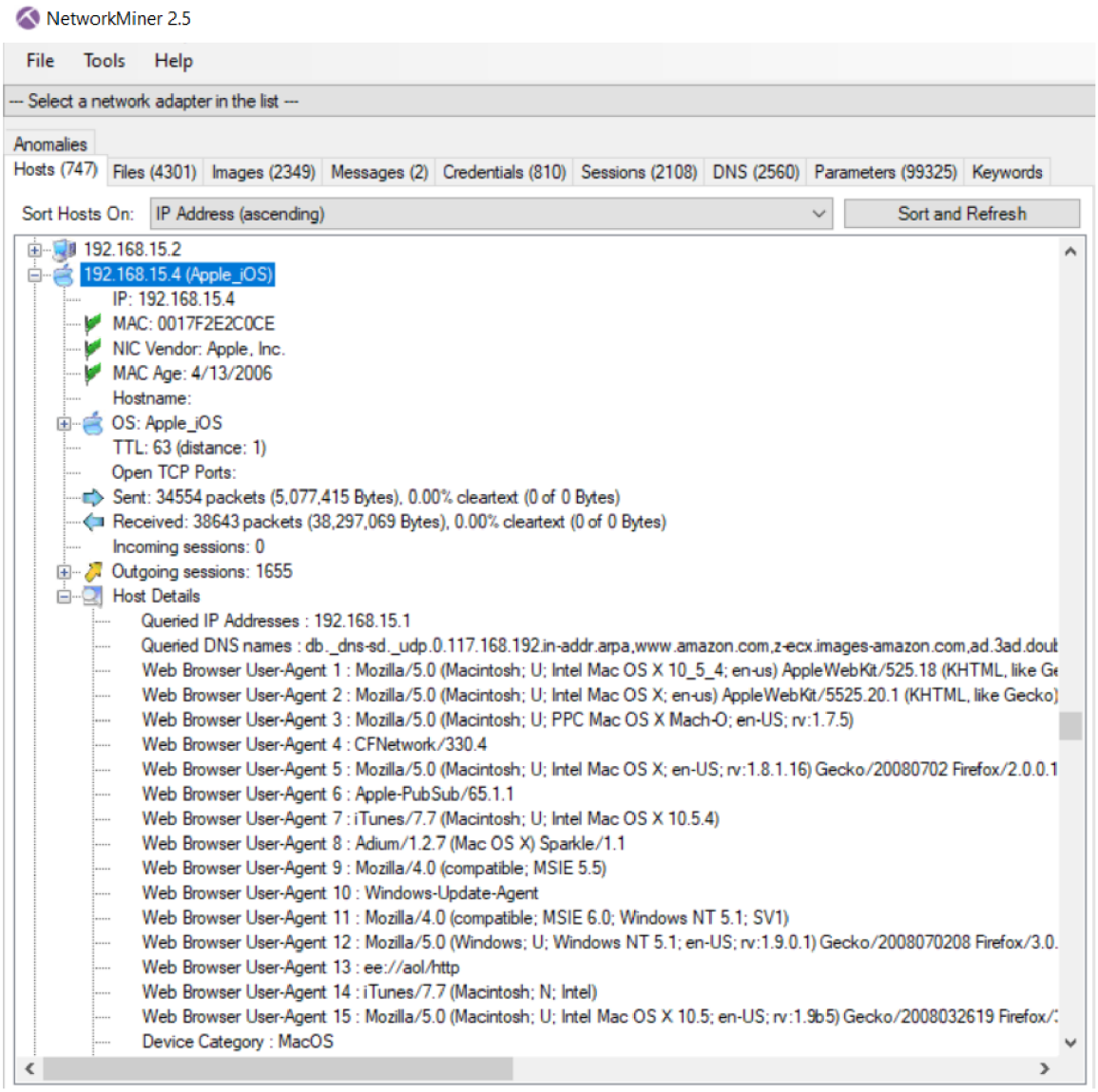


Figure 6.4: Attacker's IP details obtained using *NetworkMiner* tool

Under the *Credentials* tab in *NetworkMiner*, used mail servers can be seen. The Gmail mail server is also present, and therefore the investigators can search for *gmailchat* cookie under the *Parameters* section. The *Parameters* section also displays the timestamp of the parameter and source host in addition to the parameter value. In the Nitroba scenario, this proves that jcoachj@gmail.com was using the computer with IP 192.168.15.4 during the harassment incident. This case's parameters can be seen in the following figure 6.5.

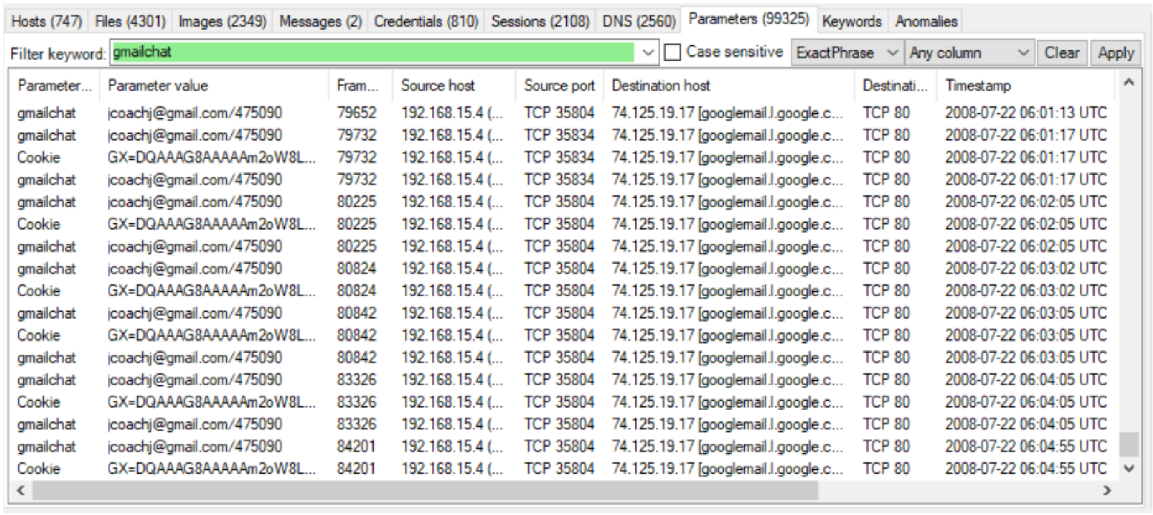


Figure 6.5: Gmailchat cookie from the parameters in *NetworkMiner* tool

tshark

tshak, a console-based tool, can be used for obtaining important conversations. The following commands lists all tcp and udp connections of the attacker's IP:

- `tshark -qn -z conv,tcp -r nitroba.pcap -R "ip.addr == 192.168.15.4" -2`
- `tshark -qn -z conv,udp -r nitroba.pcap -R "ip.addr == 192.168.15.4" -2`

For gaining all HTTP conversations of the attacker, the following command can be used (filtered tcp conversations by the user agent used in HTTP response of harassing email):

- `tshark -qn -z conv,tcp -r nitroba.pcap -R "ip.addr == 192.168.15.4 and" http.user_agent == \„Mozilla/4.0 \(compatible; MSIE 6.0; Windows NT 5.1; SV1\)\" -2`

After applying this filter, HTTP conversations are listed, including the following important ones displayed in table 6.1, which can be further analyzed using also other tools like following streams in *Wireshark*.

IP:port-src	IP:port-dst	Frames (Bytes)	Start	Duration	Protocol
192.168.15.4:35490	74.125.19.104:80	6 (3716)	14787.58927	33.1427	TCP/HTTP
192.168.15.4:35650	209.73.187.220:80	1 (1130)	14845.56531	0	TCP/HTTP
192.168.15.4:35716	65.54.186.77:80	1 (459)	14893.62378	0	TCP/HTTP
192.168.15.4:35804	74.125.19.17:80	65 (77228)	14986.32087	241.9646	TCP/HTTP
192.168.15.4:35848	69.80.225.91:80	3 (1165)	15019.62525	0.3368	TCP/HTTP
192.168.15.4:36054	74.125.19.104:80	4 (3704)	15209.98727	10.3317	TCP/HTTP

Table 6.1: Some important conversations obtained using *tshark* tool

6.2 Scanners

Scanners are the category of tools that can provide the investigator with the scan of the network. The scan can be focused on the active hosts, opened/closed ports, services, web services, vulnerabilities, detection of Wifi networks, and others. Therefore, the scanners can be distinguished as port scanners, web scanners, wifi scanners, and vulnerability scanners. Individual scanner types are described in the following sections, including the demonstration.

6.2.1 Port scanners

Port scanners are useful to identify opened ports and services that are opened on specific ports. To demonstrate port scanners the VUT FIT network was scanned using the popular port scanners. The individual scan reports with possible configuration for each tools is described in more details below. All gained reports are stored in the attached DVD. IP ranges 147.229.13.0/24 and 147.229.14.0/24 are used in the following scenarios.

There were demonstrated the following port scanners – *Nmap* and its GUI version *Zenmap*, and *Solarwinds Port Scanner*. The individual demonstrations are described below.

The results from these demonstration of port scanners are the basis of the following demonstration of web scanners. These port scanners can identify which machines have opened http port, and these machines are then scanned for web vulnerabilities. This

Nmap and Zenmap

Zenmap provides predefined profiles for scans. The GUI of Zenmap can be seen in figure 6.6. These profiles fill the Nmap command with proper parameters.

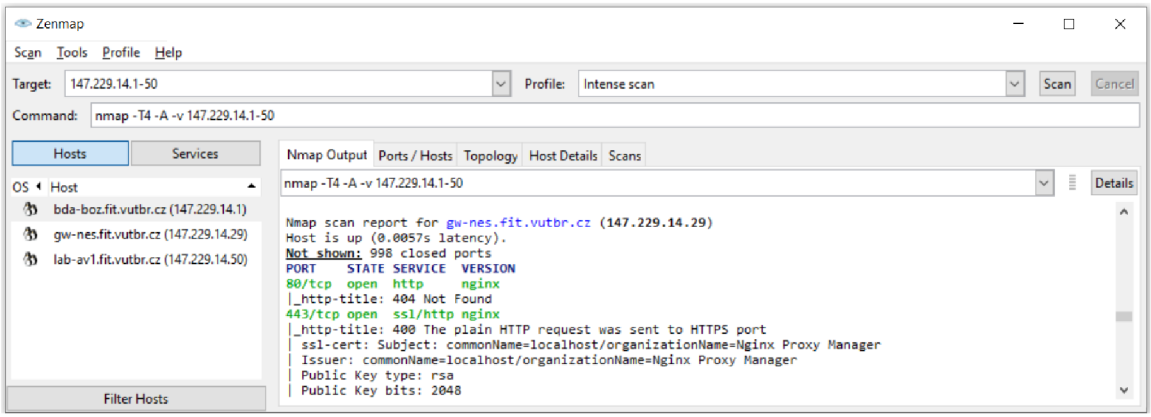


Figure 6.6: Intense scan with results in *Zenmap* tool

In this demonstration the "Quick scan" and "Intense scan" were used. "Quick scan" was performed on 147.229.13.0/24 and 147.229.14.0/24. "Intense scan" was performed on IP range 147.229.14.1-50. The intense scan also provides NSE, Ping scan, Parallel DNS resolution, SYN stealth scan, Service scan, and OS detection. Therefore the report also contains fingerprints, network distance, TCP sequence prediction, IP ID sequence generation, and traceroute. Full scan results are available on the attached DVD. Brief reports from the scans are following:

- Quick scan on 147.229.13.0/24 — `nmap -T4 -F 147.229.13.0/24`
  Start time: 2021-04-15 10:29 EDT
  Number of active hosts: 89
  Number of all opened ports: 274
- Quick scan on 147.229.14.0/24 — `nmap -T4 -F 147.229.14.0/24`
  Start time: 2021-04-15 11:11 EDT
  Number of active hosts: 30
  Number of all opened ports: 26
- Intense scan on 147.229.14.1-50 — `nmap -T4 -A -v 147.229.14.1-50`
  Start time: 2021-04-17 19:08 EDT
  Number of active hosts: 3
  Number of all opened ports: 10

Solarwinds Port Scanner

Solarwinds Port Scanner tool provides a user-friendly GUI for scan configuration and results. It is possible to save results (all or only active) in CSV, XML, or XLSX format. Reports are saved in several formats in the attached DVD for all demonstration scans.

Scan configuration includes specifying hostnames or IP address ranges, port ranges, protocol (TCP, UDP, or both), ping response (port on hosts responding on ping are scanned), resolve DNS (specific DNS servers can be optionally specified) resolve MAC, and detect OS.

This tool's demonstration includes a port scan with basic scan configuration on the defined IP address range. Two scans were performed – one for IP range 147.229.13.0/24 (port range 1 – 1 024), the second for IP range 147.229.14.0/24 (port range 1 – 10 000). Other scan configuration values were the same for both scans – TCP and UDP protocols, enabled ping response, and enabled resolving DNS. The GUI is displayed in figure 6.7.

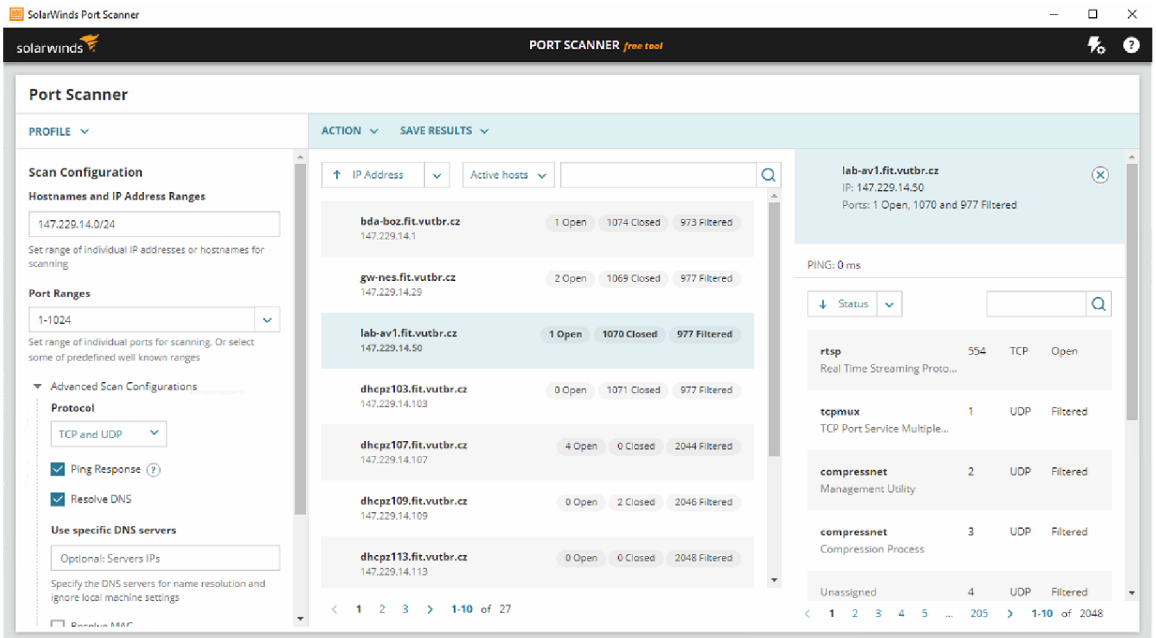


Figure 6.7: GUI of *Port Scanner* tool with results

The full scan results are available on the attached DVD under the scanners repository. The general results of performed scans are following. For each performed scan, the start time of the scan, number of active hosts, number of all opened ports, and recognized services are listed. Firstly, scan marked as *Scan one* is described. This scan focused on IP range 147.229.13.0/24. Secondly, scan for IP range 147.229.14.0/24, marked as *Scan two* is described.

- *Scan one* — 147.229.13.0/24 (port range 1 – 10 000)
  Start time: 2021-04-15 10:10 PT
  Number of active hosts: 88
  Number of all opened ports: 510
  Recognized services: 3d-nfsd, 3ds-lm, abbaccuray, ampr-inter, amqp, apex-mesh, bintec-admin, cisco-sccp, cmtp-mgt, commplex-main, cslistener, ddi-tcp-2, distinct, domain, doom, dsc, d-s-n, epmap, epmd, EtherNet/IP-1, finger, foliocorp, fs-agent, ftnmtp, ftp, ftps, gridgen-elmd, http-alt, https, ideafarm-door, idmaps, intellistor-lm, irdmi, isakmp, jlicelmd, kerberos, mdns, microsoft-ds, minipay, mmcc, mqtt, msmq, ms-wbt-server, mysql, ndl-aas, netbios-ns, netbios-ssn, nfs, nsesrvr, ntp, open-vpn, origo-native, palace-2, pando-pub, pcsync-https, pinghgl, qsnet-cond, rfb, rlm, rockwell-csp2, sdsc-lm, secure-mqtt, sentinelsrm, smtp, snmp, ssh, sunrpc, targus-getdata, targus-getdata1, targus-getdata2, telnet, us-cli, vmrdp, wsdapi, wsm-server, www-http, xmsg, zephyr-clt

- *Scan two* — 147.229.14.0/24 (port range 1 – 1 024)
  Start time: 2021-04-15 8:30 PT
  Number of active hosts: 27
  Number of all opened ports: 29
  Recognized services: device, disclose, doom, epmap, exp2, https, isakmp, mbap-s, microsoft-ds, netbios-ns, netbios-ssn, ntp, rlp, rtsp, ssh, sunrpc, www-http

These scan results were used in the following demonstration of web scanners. These tools were used as a basis for identifying the opened ports with HTTP service. Opened ports with HTTP service were found in both scans. IP addresses with the detected opened ports with HTTP service were scanned using web scanners. The results of web scanners are described in the following section.

6.2.2 Web scanners

After scanning VUT FIT network from previous section, some http opened ports were identified. This section focuses on web scans and some IP addresses with opened http ports are scanned using the tool *Nikto*. The IP addresses that were scanned for vulnerabilities using *Nikto* are within the IP range 147.229.13.0/24 and 147.229.14.0/24. *Nikto* provides output in CSV, TXT, HTM, or XML. To demonstrate various output formats, one scan (IP 147.229.14.29) is exported in all four supported formats. The most user friendly output is HTML, since it provides also host and scan summary, therefore other scans are exported only in HTML format. All scan results are available in attached DVD, and general results are described in appendix B. Table 6.2 displays some general results from appendix B. It can be seen that OSVDB-0, OSVDB-637, OSVDB-877, OSVDB-3092, OSVDB-3233, OSVDB-3268, and OSVDB-5292 were reported.

IP	Errors	Findings	OSVDB	Start time (GMT - 4)	Server
147.229.13.40	0	8	0	2021-04-18 09:37:58	Microsoft-IIS/7.5
147.229.13.41	0	11	0 3092 3233 3268	2021-04-18 09:38:56	Apache/2.4.29 (Ubuntu)
147.229.13.48	0	4	0	2021-04-18 09:40:04	lighttpd/1.4.55
147.229.13.97	0	6	0 877 3233 3268	2021-04-18 09:55:26	Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/8.0.3
147.229.13.220	0	7	0 637 3233 3268	2021-04-18 10:50:53	Apache
147.229.14.124	0	7	0 3233 5292	2021-04-18 09:32:44	nginx/1.14.2

Table 6.2: Some results from scanning VUT FIT network using *Nikto* tool

6.2.3 Wifi scanners

Wifi scanners are type of scanners with focus on the Wifi network. These scanners can detect Wifi networks around the device. Popular wifi scanners include tools such as *Kismet*, *NetStumbler*, and *Wireless Network Watcher*. All these three mentioned tools are open-source.

6.2.4 Vulnerability scanners

Another type of scanners is the vulnerability scanners. These scanners focus on the possible vulnerabilities. Usually, they are part of the other type of scanners. *Nikto* tool (and its Windows version *Wikto*) is a web scanner that scans for vulnerabilities and therefore is also part of the vulnerability scanners. *Nikto* was already demonstrated within the web scanners. Another open-source tool that can scan for vulnerabilities is *Nmap* and its GUI version *Zenmap*. These tools were also already demonstrated as port scanners, but since it uses particular parameters for enabling vulnerability scans, they are also demonstrated for vulnerability scans.

Nmap and Zenmap

Nmap provides vulnerability scan within its scripts. It can be specified using the `-script vuln` parameter. To demonstrate vulnerability scan using *Nmap*, the IP ranges 147.229.13.1-50 and 147.229.14.100-150 were scanned. Full nmap results can are available on the enclosed DVD. Table 6.3 displays vulnerabilities that were found on scanned IP ranges.

IP	Vulnerabilities
147.229.13.41	ssl-dh-params (Anonymous Diffie-Hellman Key Exchange MitM Vulnerability) http-enum (Potentially interesting directory w/ listing)
147.229.13.48	http-csrf (possible CSRF vulnerabilities) http-slowloris-check (Slowloris DOS attack)
147.229.14.124	http-enum (Robots file, Possible information file, Potentially interesting folder)

Table 6.3: Vulnerabilities found on scanned IP ranges 147.229.13.1-50 and 147.229.14.100-150 using *nmap*

6.3 Sniffers

Gathering the data is an essential use case in network forensics and in forensics in general because without the data there is nothing to be analyzed. For this use case, the tools called sniffers are used. There are many packet sniffer that can be used and they mainly differ in the platform they can be run on. However, there are also multiplatform tools. Packet sniffers are also usually the part of the more complex tools. Some sniffers capture whole packets, others focuses on flows.

The following section describes the demonstration of some Windows tools that can be used as packet sniffers. This demonstration contains as simple sniffers like *Windump*, as sniffer that are part of more complex tools like *NetworkMiner*. The dataset obtained by the demonstration of these tools is described in details in section 7.1.

6.3.1 Windows tools

To demonstrate packet capturing in the Windows environment the following tools are used – *Wireshark*, *NetworkMiner*, and *Windump*.

The demonstration of these tools was taken simultaneously to capture the data in the same network interface. In all three tools, they indicated 0 percent of the dropped packets. However, the number of the captured packet differs for each tool. To compare exactly the same time frame, the captured data described in section 7.1 are adjusted into a time range from 2021-03-14 01:07:47.263890 UTC to 2021-03-14 01:08:07.382809 UTC.

The following sections describe the demonstration of the individual windows tools. Firstly, adjusted information is provided. Secondly, the tool-specific information is discussed.

NetworkMiner

- **Filename:** 2021-03-14_networkminer_adjusted.pcap
- **Number of packets:** 1 251
- **Timeline:**
  - capture: 2021-03-14 03:07:47.263890 CET – 2021-03-14 03:08:07.382809 CET
  - adjusted: 2021-03-14 01:07:47.263890 UTC – 2021-03-14 01:08:07.382809 UTC

To be able to capture traffic data using NetworkMiner, it needs to be run as an administrator. The graphical user interface provides a user-friendly intuitive configuration for real-time data capturing. After selecting the interface using the drop-down list, the capturing is managed using the "Start" and "Stop" buttons. The GUI of NetworkMiner can be seen in figure 6.8.

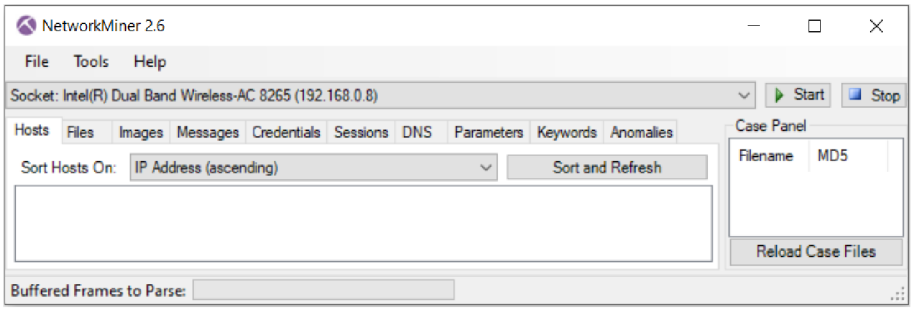


Figure 6.8: GUI of *NetworkMiner* tool for real-time data capturing

Windump

- **Filename:** 2021-03-14_windump_adjusted.pcap
- **Number of packets:** 1 266
- **Timeline:**
  - capture: 2021-03-14 02:07:47.265168 CET – 2021-03-14 02:08:07.382467 CET
  - adjusted: 2021-03-14 01:07:47.265168 UTC – 2021-03-14 01:08:07.382467 UTC

The original Windump uses a WinPcap, but the WinPcap is not supported in Windows 10, the npacp is used instead. Therefore, for Windows 10 the original WinDump will not work. The "WinDump for Npcap" is available, and this tool was used in this demonstration.

Windump provides command line interface. There are several parameters that can be used to specify the required configuration. This demonstration used only the parameter for specifying the interface and parameter for the output file. The whole command used for this demonstration is:

```
WinDump.exe
   -i \Device\NPF_8B337493-B645-4814-8510-4947B08E553A
   -w 2021-03-14_windump.pcap
```

Wireshark

- **Filename:** 2021-03-14_wireshark_adjusted.pcap
- **Number of packets:** 1 266
- **Timeline:**
  - capture: 2021-03-14 02:07:47.265168 CET – 2021-03-14 02:08:07.382467 CET
  - adjusted: 2021-03-14 01:07:47.265168 UTC – 2021-03-14 01:08:07.382467 UTC

Wireshark, similarly to NetworkMiner, provides a graphical user interface. The *Capture Interfaces* window can be seen in figure 6.9. It can also be seen that it can also be specified capture filters for selected interfaces, but only one interface with no capture filters was used in this demonstration. Configuration, as shown in figure 6.9, was used for demonstration. The start button begins the real-time data capturing with defined preferences, and when pressing the stop button, the capturing is finished.

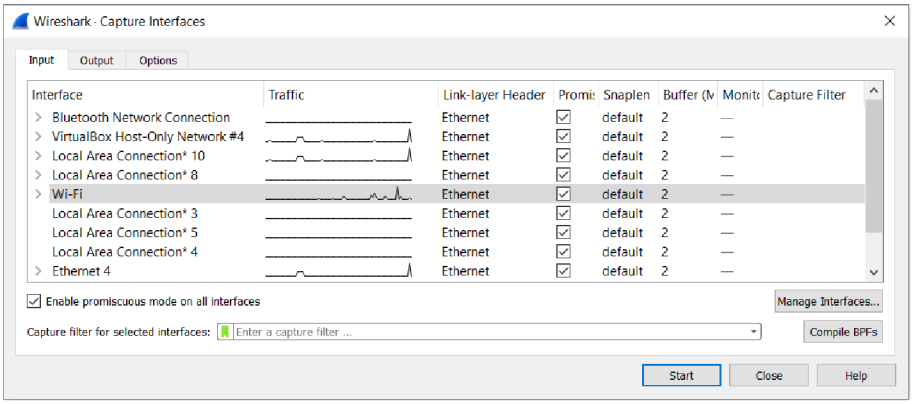


Figure 6.9: GUI of *Wireshark* tool for real-time data capturing

6.3.2 Summary and comparison

Previously the overview of capturing the data using some Windows sniffers was provided. In all tools, the capturing was intuitive, and it did not require special configurations. Another similar feature for all tools is that there was indicated zero packet drop.

The following differences were observed in demonstrated sniffers:

- the user interface — *Windump* is command-line tool, *NetworkMiner* and *Wireshark* provides GUI
- number of captured packets — it can be seen that the captured data using *Windump* and *Wireshark* contain 1 266 packets, while the PCAP file obtained using the *NetworkMiner* only 1 251 packets. Packets that are present in *Windump* and *Wireshark* and are not present in *NetworkMiner* are displayed in table 6.4
- IPv6 support — Although *NetworkMiner* should also process IPv6 traffic even in the free version of the tool, it was not captured. IPv6 traffic is present in the PCAP files obtained using *Windump* and *Wireshark*
- captured time — arrival time of packets differs slightly in milliseconds for *NetworkMiner* — *Windump* and *Wireshark* have the same capture time.
- order of TCP ACK and SYN packets — TCP SYN and ACK packets for *NetworkMiner* and not is in the same order as in PCAPs obtained using *Windump* and *Wireshark*, and therefore some packets are marked as "TCP ACKed unseen segment" and "TCP Spurious Retransmission" in PCAP file obtained using *NetworkMiner*
- packet size limit — *Windump* has limited packet size by default. Therefore some packets for *Windump* are not complete and are marked as "Packet size limited during

capture", for example, the TLSv1.2 Client Hello and Server Hello. There are no truncated packets in PCAPs obtained using *Wireshark* and *NetworkMiner*.

Packet number	Protocol	Source	Destination	Description
43	TCP	54.145.80.191:443	192.168.0.8:56168	[SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM=1 WS=256
285	TCP	54.145.80.191:443	192.168.0.8:56168	[TCP Retransmission] [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM=1 WS=256
572	TCP	35.186.224.47:443	192.168.0.8:49797	[ACK] Seq=1 Ack=36 Win=266 Len=0
573	TLSv1.2	35.186.224.47:443	192.168.0.8:49797	Application Data
426, 575, 756	MDNS	fe80::1860:3d5a:7859:b7c4	ff02::fb	Standard query 0x0000 PTR _companion-link._tcp.local, „QU" question PTR _homekit._tcp.local, „QU" question OPT
746, 747, 903, 904	MDNS	fe80::1860:3d5a:7859:b7c4	ff02::fb	Standard query 0x0000 A netw.local, „QM" question
748, 749, 801, 802	LLMNR	fe80::748e:73a3:45d:b9d8	ff02::1:3	Standard query 0xb2e1 A netw

Table 6.4: Missing packets in PCAP captured using *NetworkMiner*

6.4 Visualizers

When investigating the incident, it is handy to have visualized the traffic or other network data. The tools that can provide the visualized network data are called visualizers. They can provide the investigator with the whole picture, and therefore, may fasten the investigation.

The following sections describe and compare open-source visualizers. It should be mentioned that commercial visualizer tools can provide more information; for example, the Solarwinds NPM tool provides a unified view of network connections, applications, dependency relationships, topology, and ADM information. VisualRoute is also a powerful tool.

To demonstrate the visualizer tools, datasets described in chapter 5 are used. In addition the existing datasets, also newly created data are used. These new datasets are described in detail in chapter 7.

6.4.1 CapAnalysis

CapAnalysis is web-based tool, and for demonstration the demo version is used. The *Nitroba* dataset described in chapter 5 is used to demonstrate how the data are processed and visualized.

Although this tool supports more PCAP to be loaded, the following demonstration uses only one PCAP file. The tool contains visualization tabs that can display flows, overview, statistics, per hour statistics, geo map, source and destination IPs statistics, protocols, and timeline.

The following figures display the data from the *Nitroba* dataset for each visualization tab. The complete screenshots can be seen in the attached DVD.

The *Flow tab* lists of all TCP and UDP streams. It displays date, time, source and destination IP, destination name, source and destination port, protocols (L4 and application protocol), country, sent and received bytes, lost bytes, sent and received packets, and duration. The displaying of the destination name is useful in this *Nitroba* case, since it can be seen visited websites. The figure 6.10 visualizes the flow when harassing email was sent using the www.willselfdestruct.com.

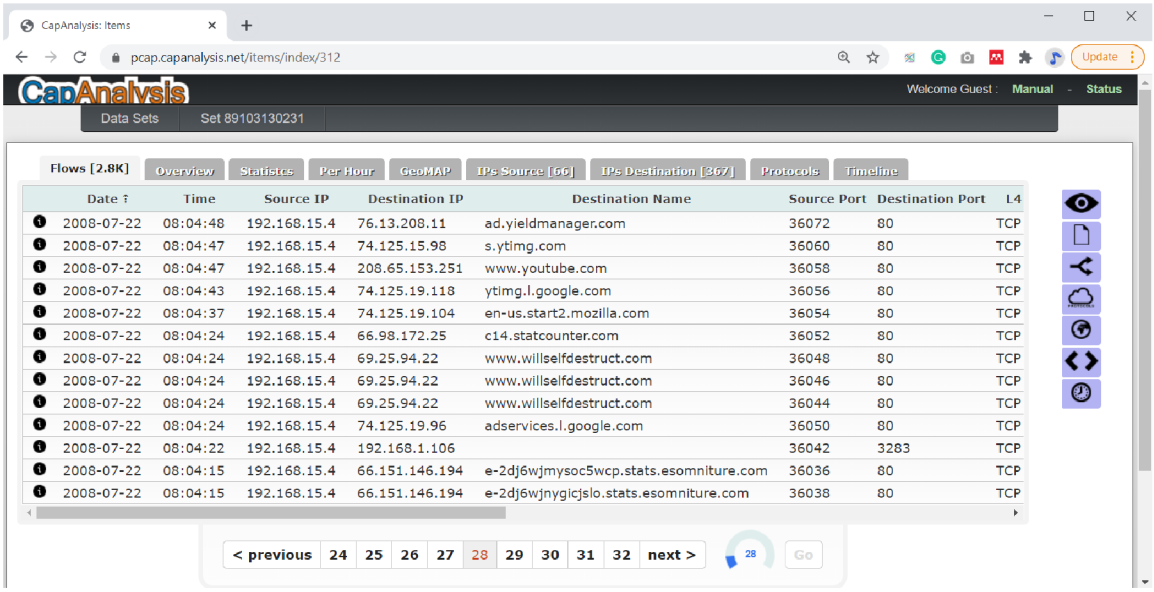


Figure 6.10: Flows tab of *CapAnalysis* tool with *Nitroba* data

The *Overview tab* displays the distribution of destination ports, hourly distribution in map, and protocols according to the countries and days. The part of the data can be seen in figure 6.11.

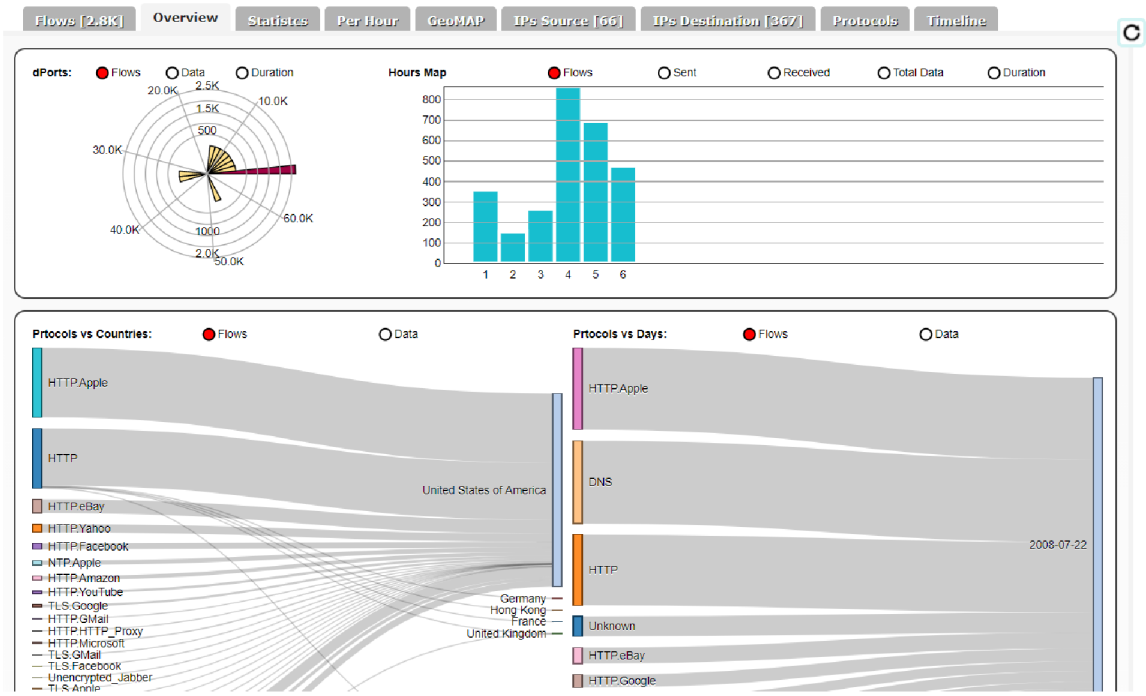


Figure 6.11: Overview tab of *CapAnalysis* tool with *Nitroba* data

The *Statistics tab* provides statistic charts with statistics about Source and Destication IP in flows and sent or received data, protocol and country statistics in flows and duration in flows. Figure 6.12 shows the source IP statistics.

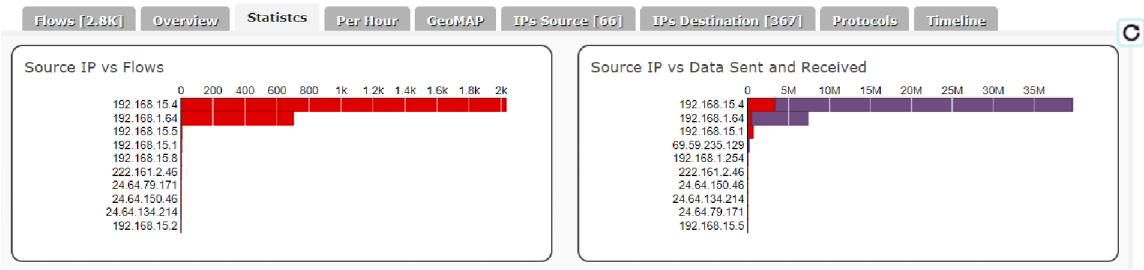


Figure 6.12: Statistics tab of *CapAnalysis* tool with *Nitroba* data

The *Per Hour tab* shows overview per hour, such as number of flows, duration, and number of sent and received bytes. This visualization can be seen in figure 6.13.

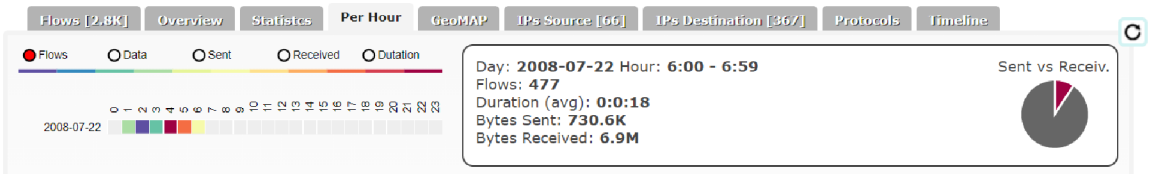


Figure 6.13: Per Hour tab of *CapAnalysis* tool with *Nitroba* data

The *GeoMAP tab* visualize flows, data, sent and received bytes in world map. The details for Unites States are shown in figure 6.14.

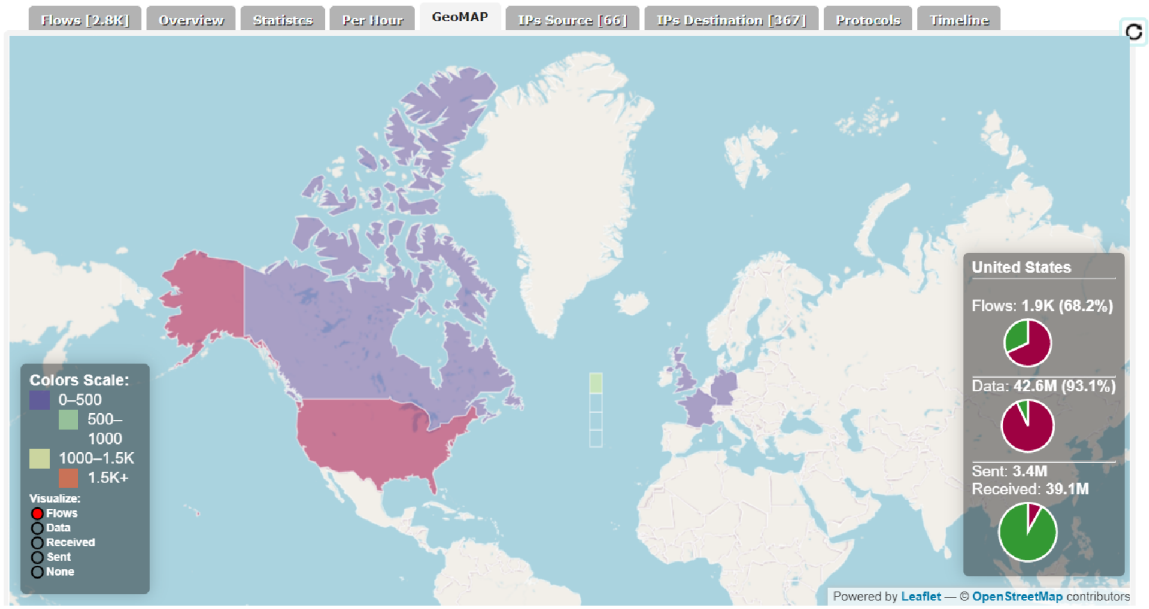


Figure 6.14: GeoMAP tab of *CapAnalysis* tool with *Nitroba* data

The *IPs Source tab* and *IPs Destination tab* displays all source and destination IPs of connections sorted according to the most frequent appearence in flows. The figure 6.15 shows the most used source IPs. It can be seen that the attacker's ip 192.168.15.4 is the most used source IP. The figure 6.16 shows the most used destination IPs.

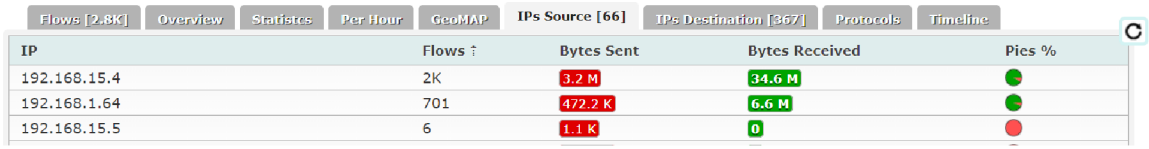


Figure 6.15: IPs Source tab of *CapAnalysis* tool with *Nitroba* data

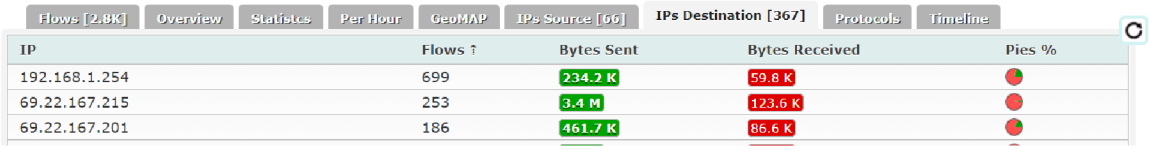


Figure 6.16: IPs Destination tab of *CapAnalysis* tool with *Nitroba* data

The *Protocols tab* visualizes used protocols in the chart based on the source and destination IP. The visualized protocols for *Nitroba* scenario can be seen in figure 6.17.

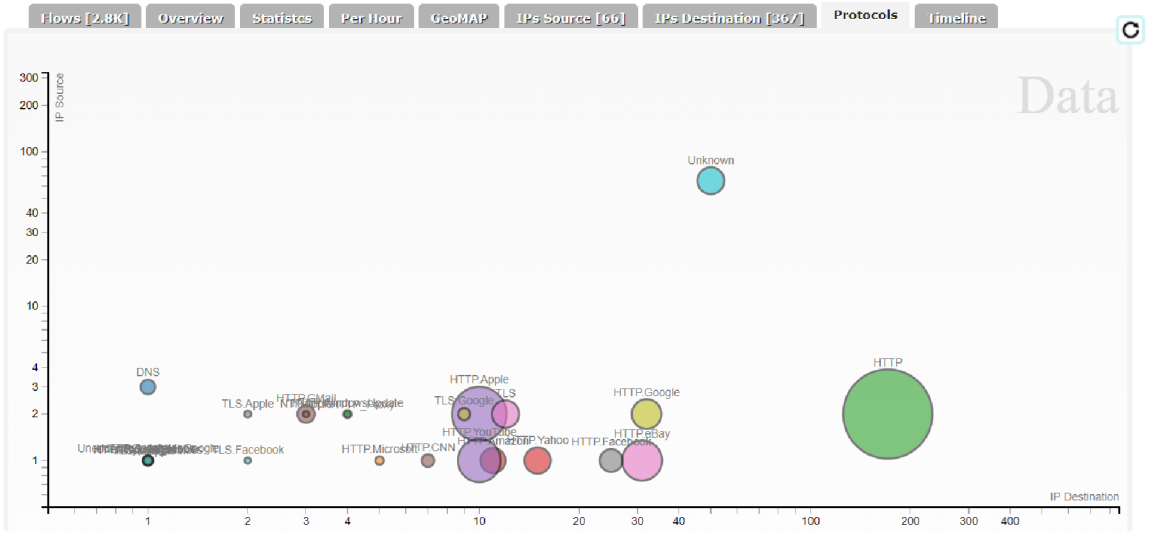


Figure 6.17: Protocols tab of *CapAnalysis* tool with *Nitroba* data

The last visualizetion tab is called *Timeline* and it provides the timeline view on connections, all data, received data, and sent data. The chart for connections timeline can be seen in figure 6.18.

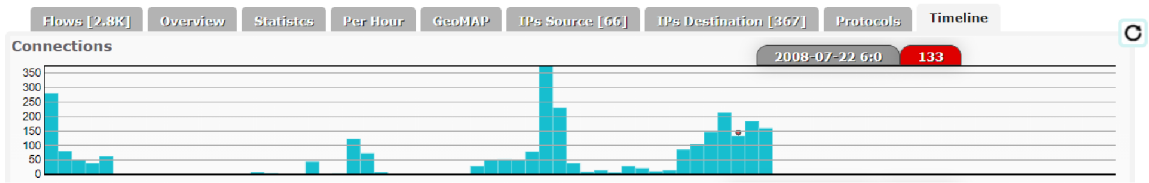


Figure 6.18: Timeline tab of *CapAnalysis* tool with *Nitroba* data

6.4.2 Graphical Ping (NetScanTools)

Graphical Ping is a visualizer tool that is a part of the suite of tools called *NetScanTools*. This visualizer is available in the basic version of *NetScanTools*, but the free version provides only the visualization graph, and more features like printing graph, saving graph, reloading graph, database with reports, and variable time axis are available in commercial *NetScanTools Pro*. The *NetScanTools Pro* is also available in demo version that includes also reports, but other Pro features are not included. Therefore, the *NetScanTools Pro Demo* was used for this demonstration to export some reports.

The *Graphical Ping* provides an intuitive graphical user interface with specifying the target IP address or hostname, the time between, timeout, data size, and TTI before starting the ping. This tool is helpful for displaying the RTT in time for a specified target hostname or IP address. It is possible to apply the autoscale to the RTT axis for better visualization. The reports can provide a list of all missing packets and a list of events with RTT higher than a specified value.

For demonstration the following scenarios were used:

1. Ping on the target hostname "google.com"
2. Ping on the internal IP with destroyed connection after some time

In this demonstration for each scenario the following reports are provided and available on attached DVD:

- list of all missing packets,
- list of all events (events with RTT higher than 0 ms)
- list of events with RTT higher than 2000 ms (RTT 20 ms).

Ping on the target hostname "google.com"

Figure 6.19 shows the graph from *Graphical Ping* on target hostname "google.com". There is also displayed IP address for specified hostname – 172.217.23.238. Total recorded time for this scenario is 33 seconds (from 13:23:17 to 13:23:50). It can be seen that there were no missing replies. RTT was not higher than 60 ms for all recorded events. Only 5 events have RTT higher than 20 ms. This ping was performed from the network using a Wi-Fi connection to the Internet provided by UPS in the Czech republic.

Ping on the internal IP with destroyed connection after some time

Figure 6.20 shows the graph from *Graphical Ping* on target IP 192.168.0.2. Total recorded time for this scenario is 42 seconds (from 13:32:16 to 13:32:58). It can be seen that there were some packets with no response. After approximately the half time of recorded traffic, it can be seen that there are only packets with no response that indicated destroyed connection. RTT was not higher than 300 ms for all recorded events. The most of packets had RTT above 20 ms.

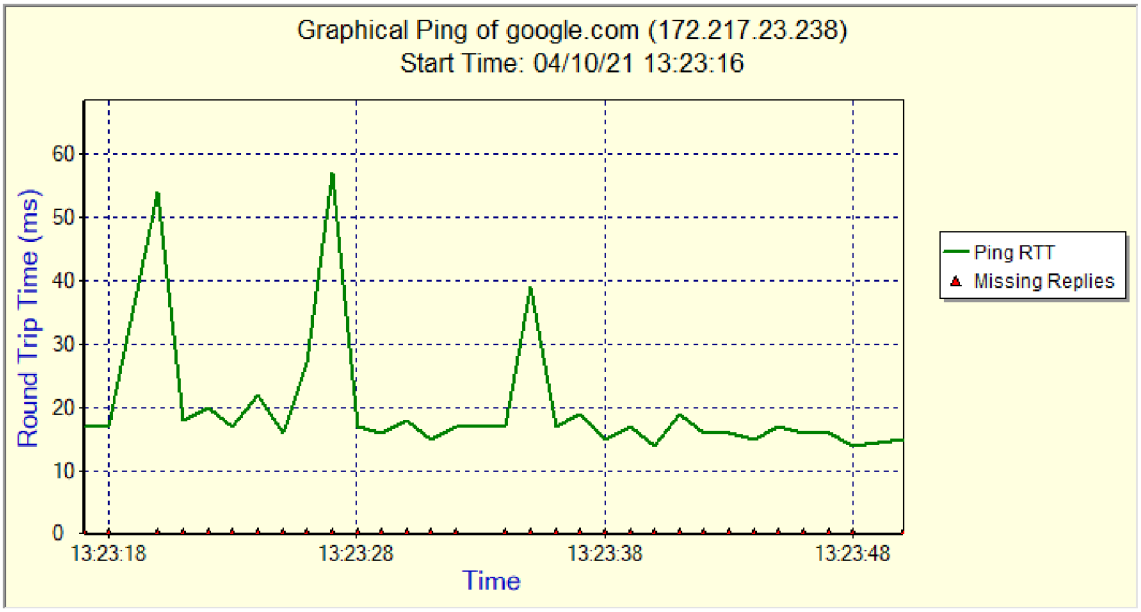


Figure 6.19: Visualized real-time ICMP traffic in *Graphical Ping* tool on target hostname "google.com"

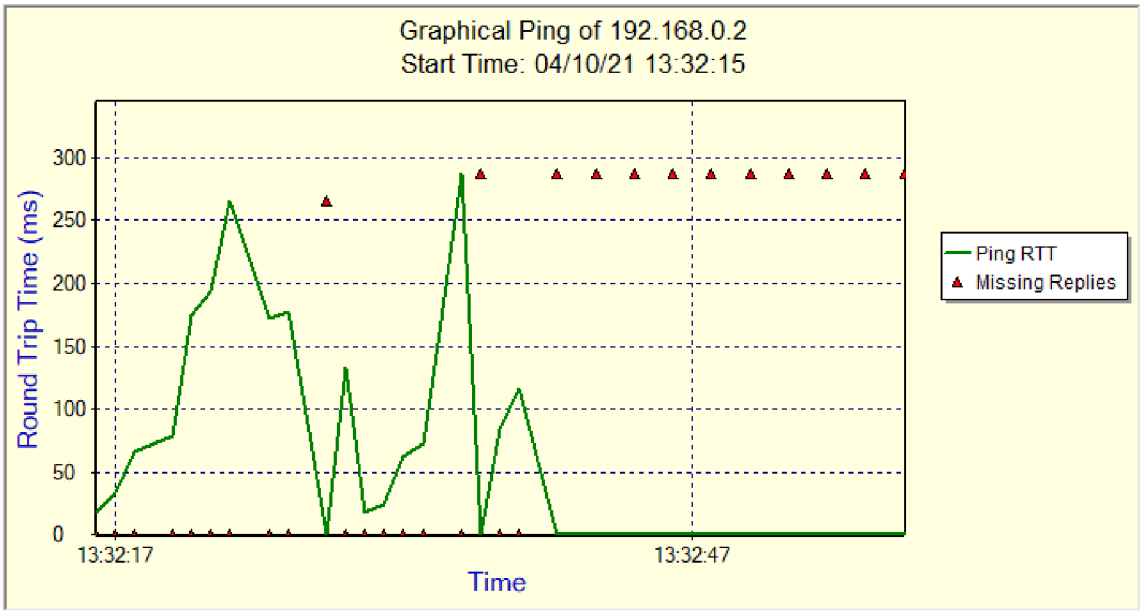


Figure 6.20: Visualized real-time ICMP traffic in *Graphical Ping* tool on target IP 192.168.0.2 with destroyed connection after some time

6.4.3 EtherApe

EtherApe tool visualizes the traffic in real-time. It can be used to visualize online data (monitoring the real traffic), or the data can be read from the capture file. The following demonstration of this visualizer is based on already captured data – the *Nitroba* dataset described in chapter 5 is used. In this demonstration, *EtherApe* was run with replaying data from the *Nitroba* PCAP file with filter on the attacker's IP 192.168.15.4. There can also be set the mode of the reading the data, in this case the *IP mode* is used. The data are actively visualized in the diagram as they are replayed from the PCAP file. During the visualizing the network traffic data, some exports are created. The export is in XML format. The corresponding screenshots and exports are stored in the attached DVD.

The figure 6.21 displays the graphical interface. It provides to pause the visualization, or setup some preferences. The protocols and nodes can also be displayed for the current state. As the data are read, the animated dependencies between hosts and protocols are visualized in the diagram. The bottom status bar informs that the data are read from the file and the selected mode – in this case the IP mode is used.

In figure 6.22, there can be seen the issue of sending harassement email from the attacker's IP 192.168.15.4 to the IP 69.25.94.22 that is the IP of www.willselfdestruct.com as it can be seen in the figure 6.3 from *NetworkMiner* tool.

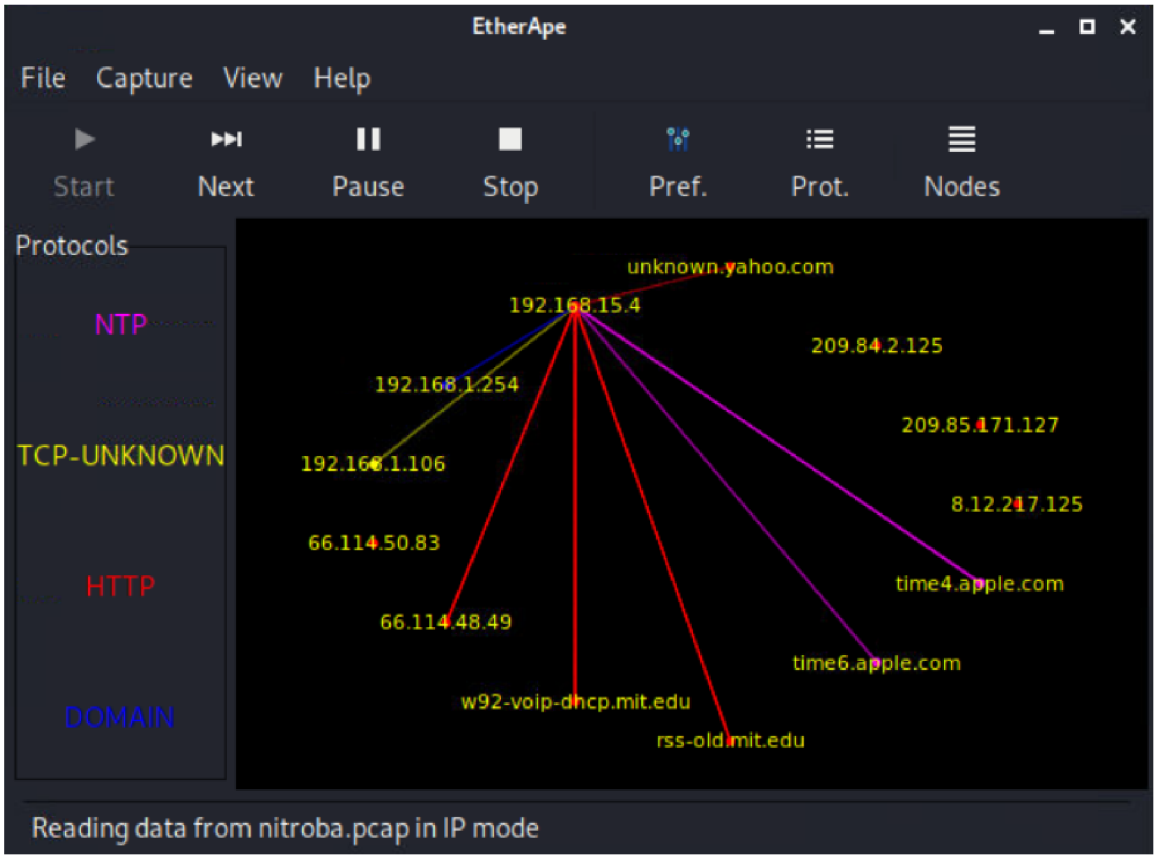


Figure 6.21: Visualized part of traffic from *Nitroba* dataset in *EtherApe* tool

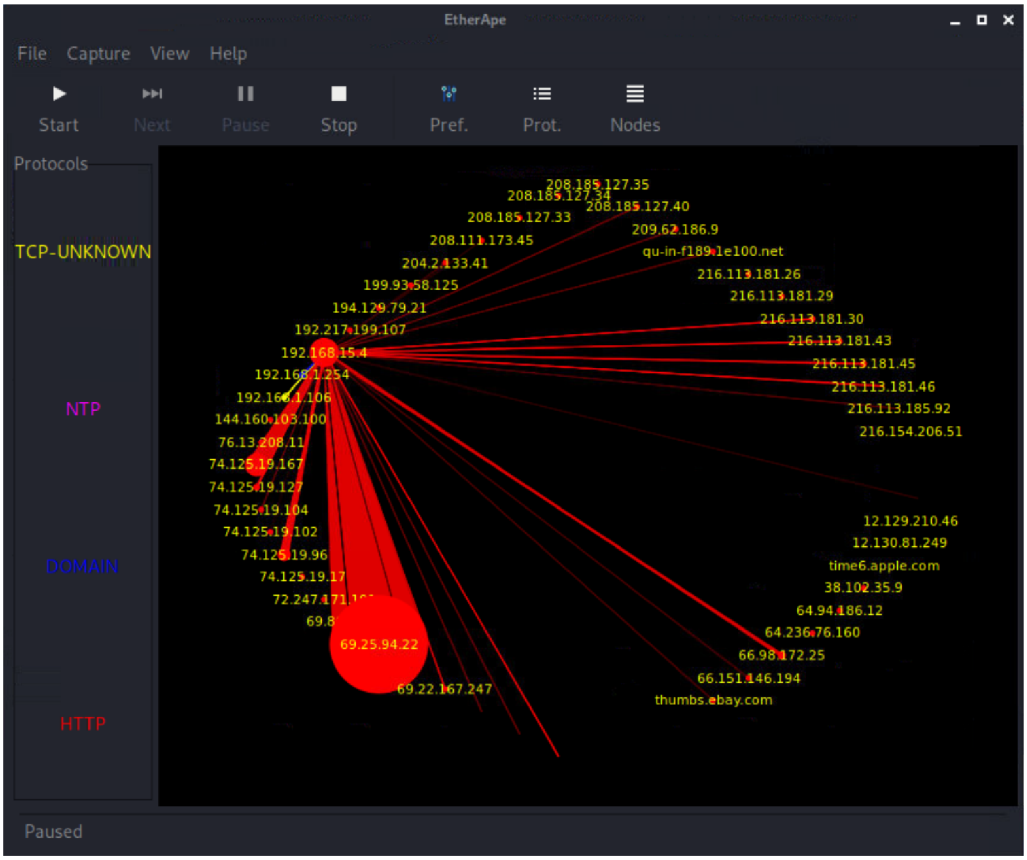


Figure 6.22: Visualized sending harassement email from *Nitroba* dataset in *EtherApe* tool

6.4.4 PcapXray

PcapXray can assist in visualizing the traffic data – the hosts and the connection between them, including the type of traffic. It is possible to filter traffic according to protocols or visualizing only malicious traffic. This tool also provides selecting the source and destination of the data displayed. In addition to the static graph, the interactive graph is also available. The *PcapXray* automatically generates reports based on the visualized data – png file with the static graph, html file with the interactive graph, txt files with the communication, device and packet details. All these report are available on the attached DVD.

This visualizer is demonstrated using the different types of data:

- encrypted data using *Wireshark dumpng* dataset described in section 5.5.1,
- dataset with small number of packets and hosts — *NETRESEC Packet injection attacks* datasets described in section 5.3,
- medium size dataset of generic HTTP traffic described in section 7.1,
- the dataset with more packets and hosts — *Nitroba* dataset described in section 5.2.

This tool is not able to recognize the encrypted traffic, there also is not a possibility to attach decryption key. The figure 6.23 visualizes SSL traffic from the *Wireshark dumpng* dataset, and this traffic is displayed as “Unknown Protocol”.

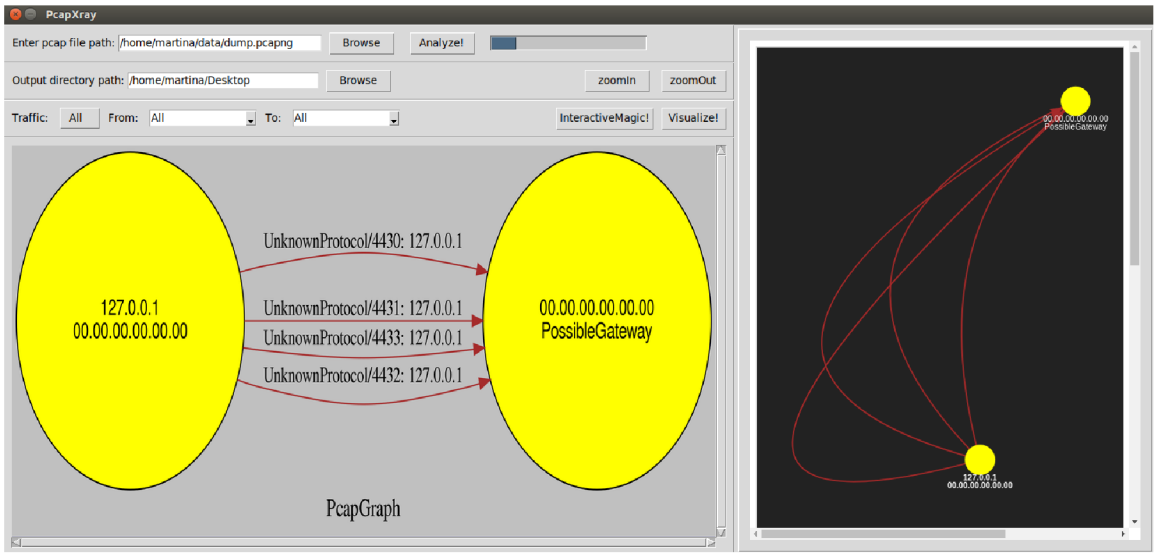


Figure 6.23: Visualized encrypted traffic from *Wireshark dumpng* dataset in *PcapXray* tool

The *PcapXray* is effective for PCAP files with small number of packets and hosts, so that the whole data can visualized in graphs. These data are easy to read and the visualization displays the communication between hosts and other traffic information. The figure 6.24 shows the packet injection attack data against *www.02995.com* described in section 5.3.

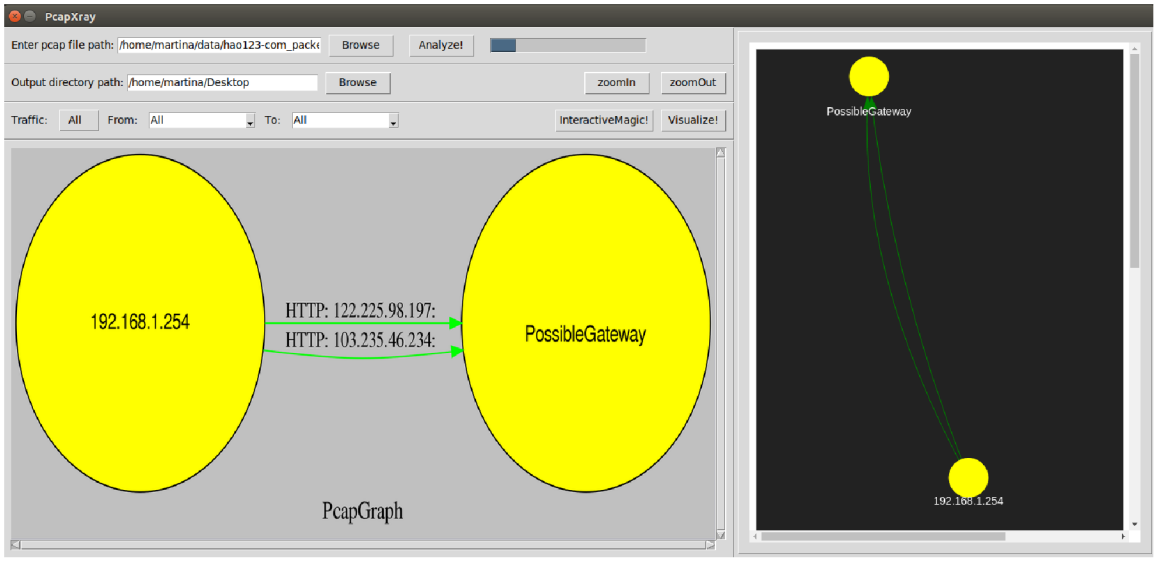


Figure 6.24: Visualized small traffic from *NETRESEC Packet injection attacks* dataset in *PcapXray* tool

The figure 6.25 visualizes simple HTTP traffic of medium size captured in dataset described in chapter 7 section 7.1. It can be seen that the *PcapXray* can work well with medium size traffic. Although the PcapGraph is quite unreadable in the program, the output PNG file can be zoomed in, and the data is easy to read.

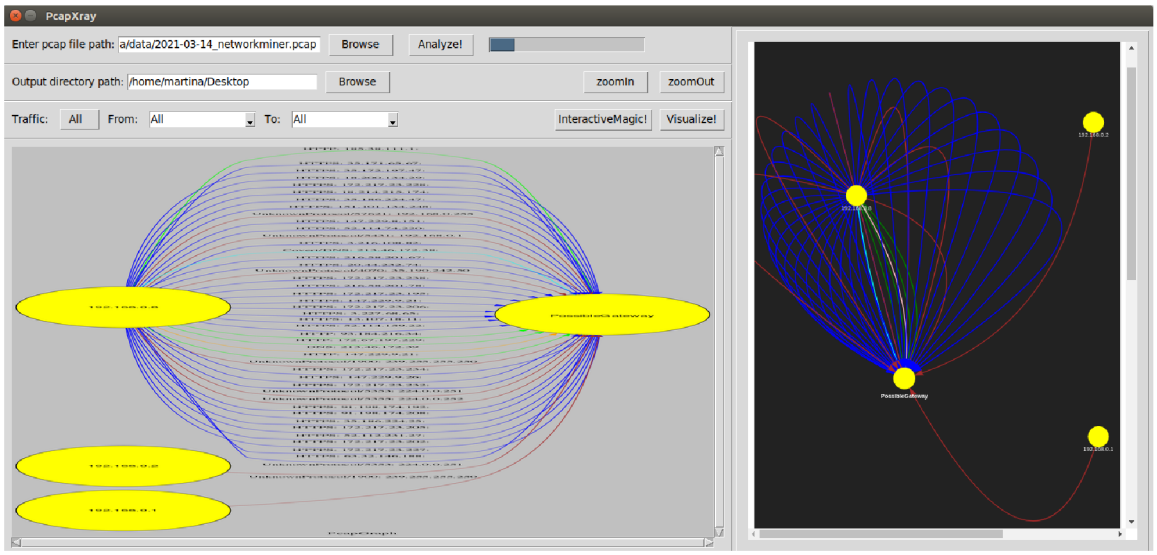


Figure 6.25: Visualized medium size traffic from own HTTP dataset

The visualized data are difficult to read when more hosts are displayed, such as in figures 6.26 and 6.27 of *Nitroba* dataset. It is not easy to read even the report PNG files. The large pcap files also take some time to process the traffic data.

The figure 6.26 displays the traffic data from the attacker's IP 192.168.15.4. It can be seen that although this is not whole traffic data, the visualization is not easy to read.

In figure 6.27 only malicious traffic is visualized – red color is used. The attacker's IP 192.168.15.4 is present in this visualization.

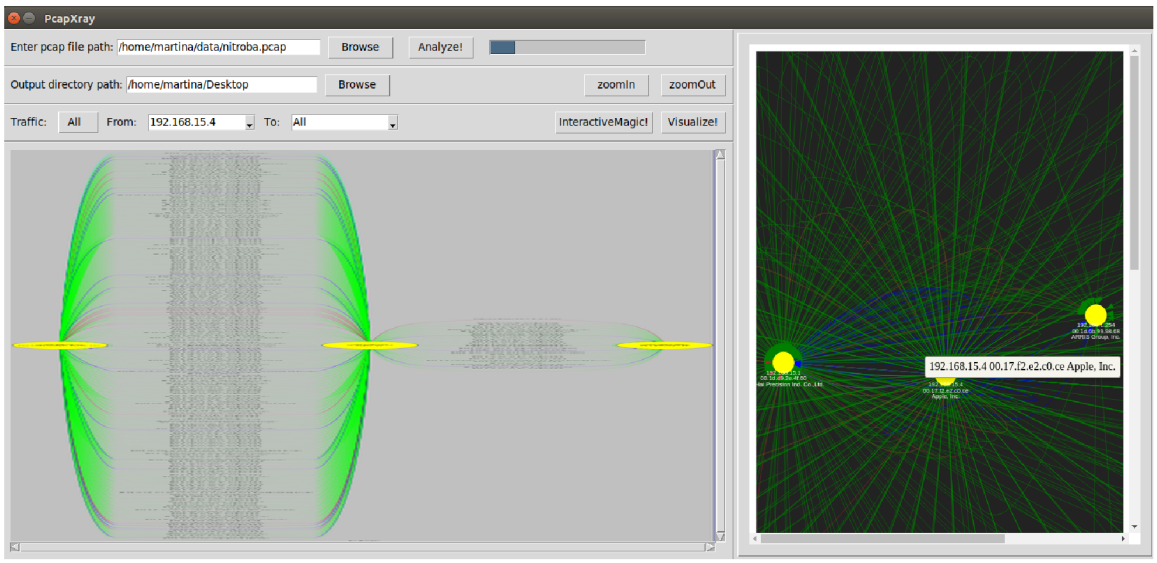


Figure 6.26: Visualized large traffic from *Nitroba* dataset in *PcapXray* tool filtered for attacker's IP 192.168.15.4

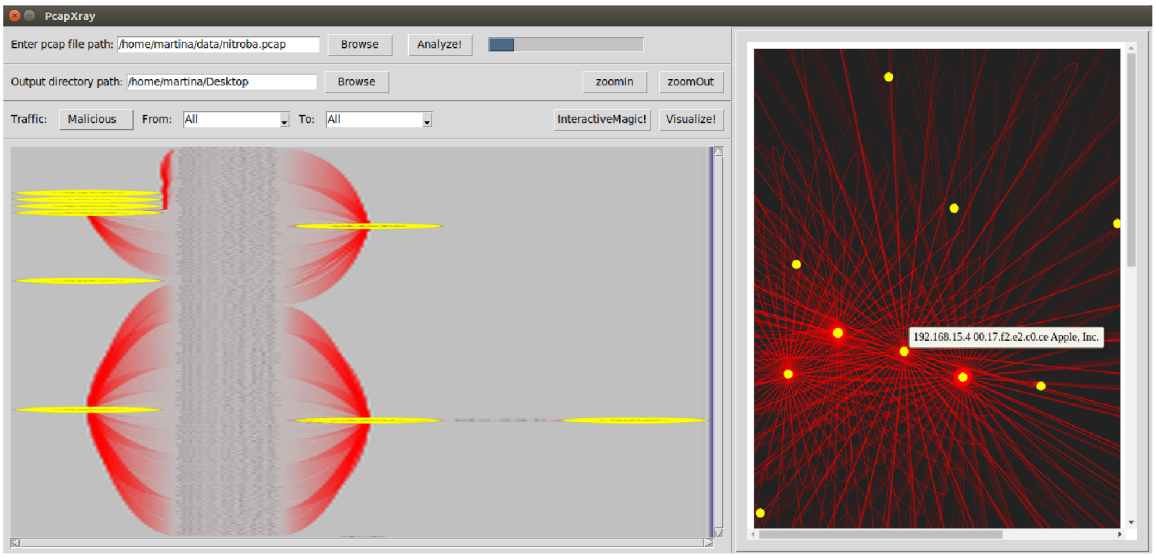


Figure 6.27: Visualized large traffic from *Nitroba* dataset in *PcapXray* tool filtered for malicious traffic

6.4.5 Web Historian

Web Historian is a Google Chrome extension that visualizes web browser history data. The web browser history data can contain sensitive information that can help with the investigation. The visualization of these data helps the investigator process the data that are usually available only as JSON or CSV files.

Google Chrome history data that was used to demonstrate this tool are described in section 7.3. Processed data are available in individual tabs based on visualized category:

- *Home* — provides review of the week, and compares it to the previous week, such as most visited website, and most searched term. Figure 6.28 displays home page with processed data from dataset described in 7.3.

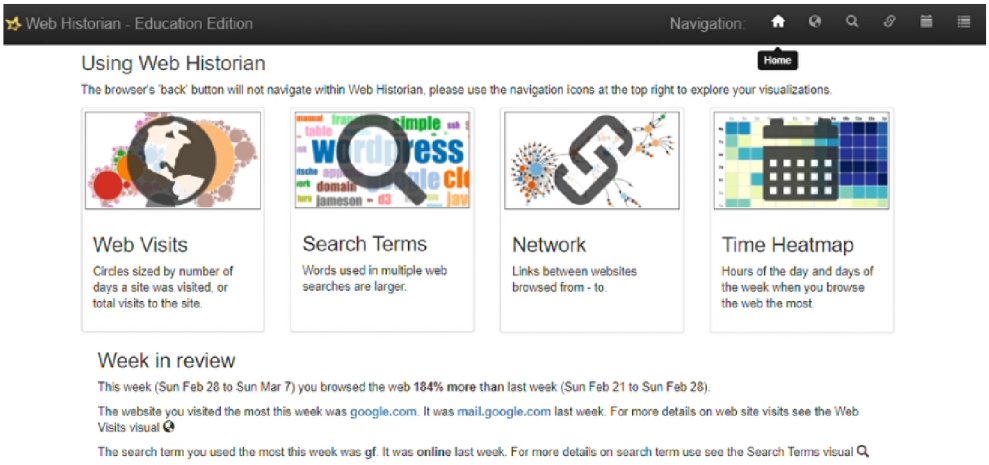


Figure 6.28: Visualized data in *Web Historian* Home tab

- *Web visits* — visualizes the most visited websites in graph, the time range can be set according to the preferences. Figure 6.29 displays the most visited websites from the demonstration dataset for the whole time. Right clicking on individual parts of the graph provides information in other visualisation tabs. It can be seen that "vutbr.cz" was the most visited website from Jan 1, 2021, to Mar 7, 2021. The graphs are animated when changing the time range, and placing the individual parts of the graph is not fix. The whole animation can be seen on the attached DVD under the demonstration repository for *Web Historian*.

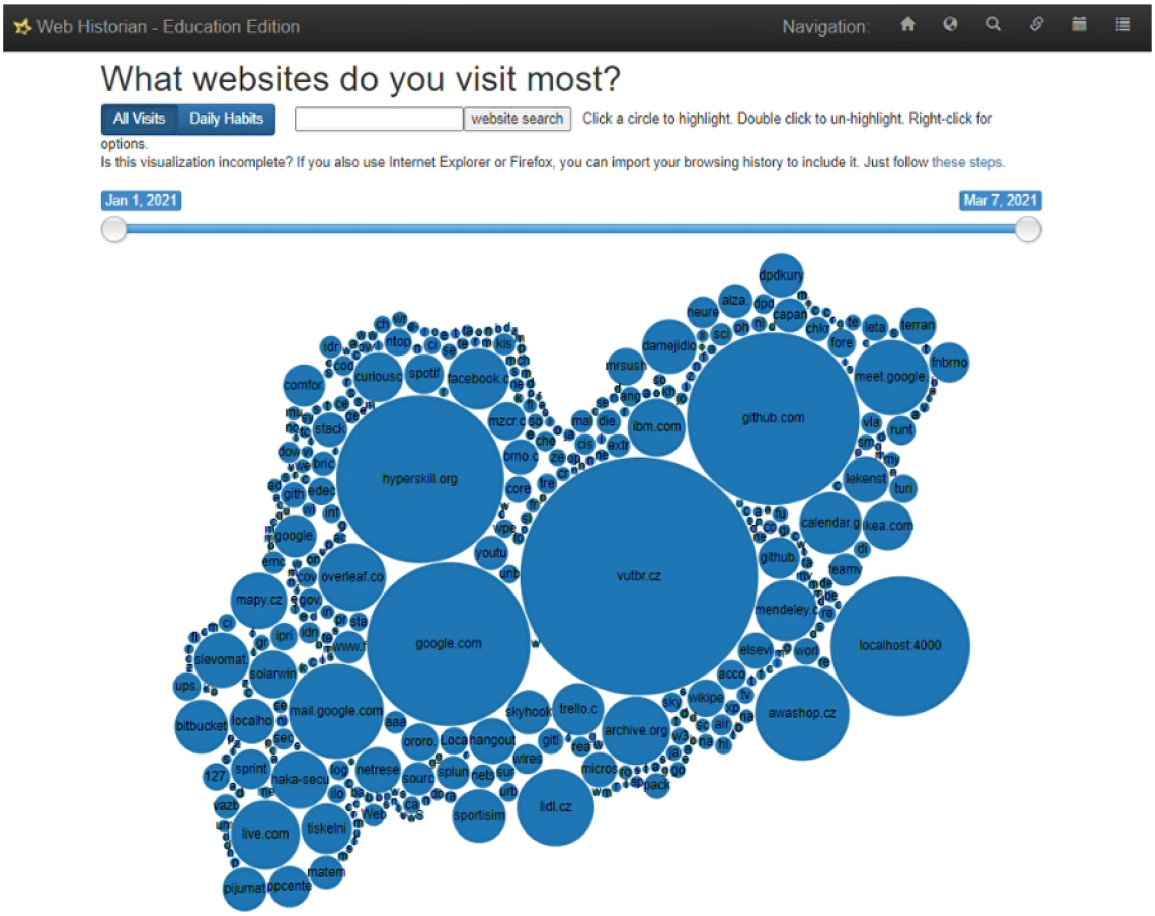


Figure 6.29: Visualized data in *Web Historian* Web Visits tab

- *Search terms* — similar to the web visits tab, the time range can also be specified by the user. The visualized graph contains the most searched terms. The general information about a number of unique search terms and unique words is provided at the top of the visualization. More information can be gained by holding the mouse over the part of the graph or by right-clicking. The more searched term the bigger font. Figure 6.30 visualizes the most searched terms from demonstration dataset during January. It can be seen that the most frequent searched terms include "java", "network", "forensics", and "security".

Figure 6.30: Visualized data in *Web Historian* Search Terms tab

- *Network* — visualizes connections between web sites. General information is provided at the top of the visualization page, and more action can be taken on the visualized graph to obtain additional information. Figure 6.31 visualizes the most searched terms from demonstration dataset during first two days of January.

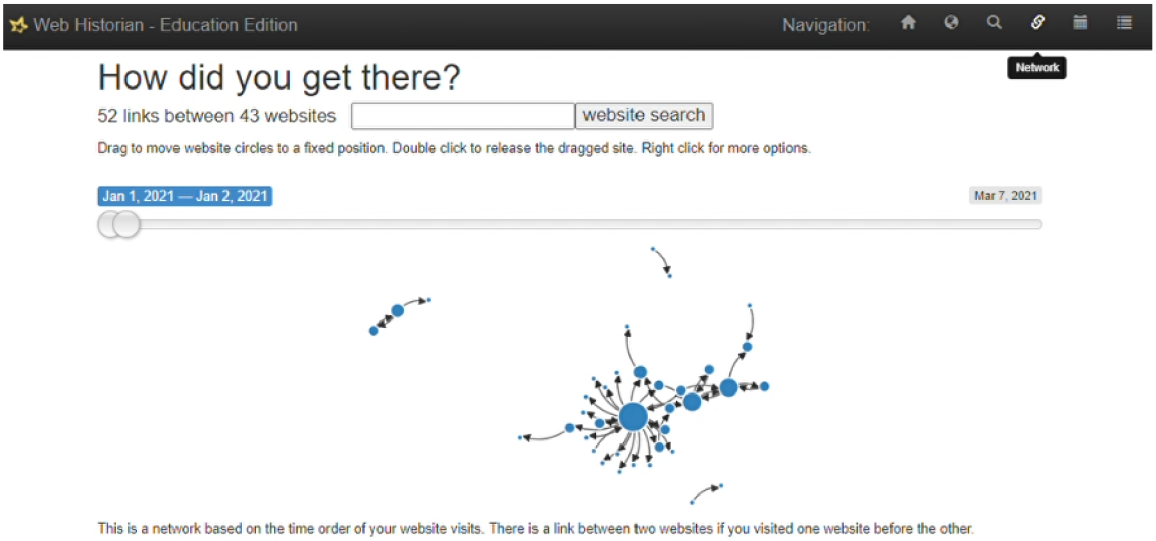


Figure 6.31: Visualized data in *Web Historian* Network tab

- *Time heatmap* — is useful to see how the websites were visited during the specified time frame. It is able to specify all weeks or just one specific one. Like in other

visualization tabs, other actions can be taken on the visualized graph by right-clicking or hovering the mouse over the graph's specific part. Figure 6.32 visualizes the last week of historical data from the demonstration dataset.

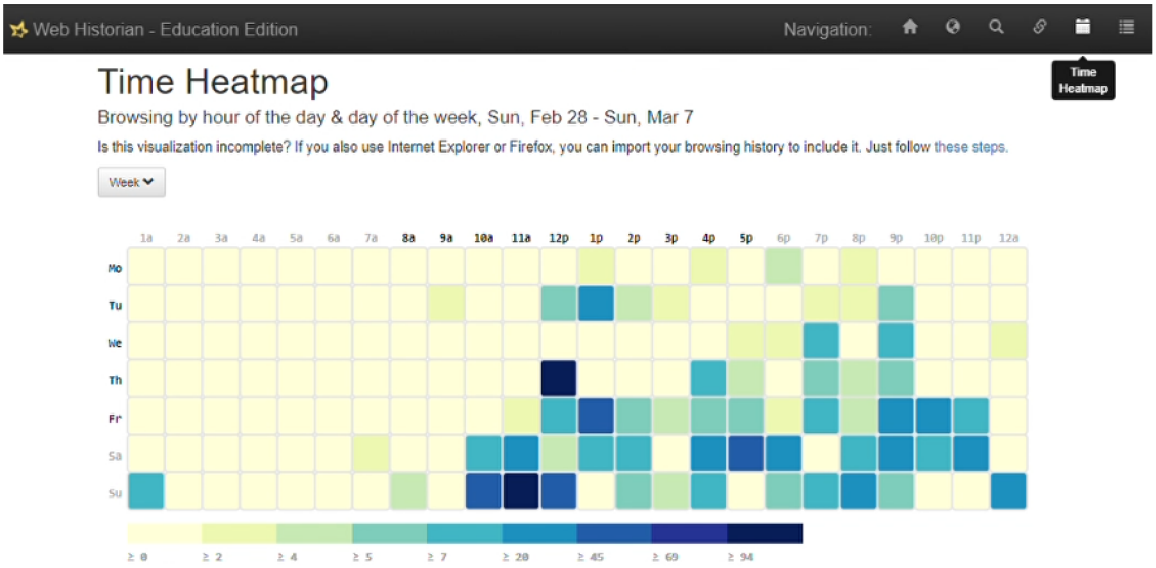


Figure 6.32: Visualized data in *Web Historian* Time Heatmap tab

- *Data table* — displays all visits or just domains in the table view. Figure 6.33 visualizes the first part of the table from processed demonstration dataset.

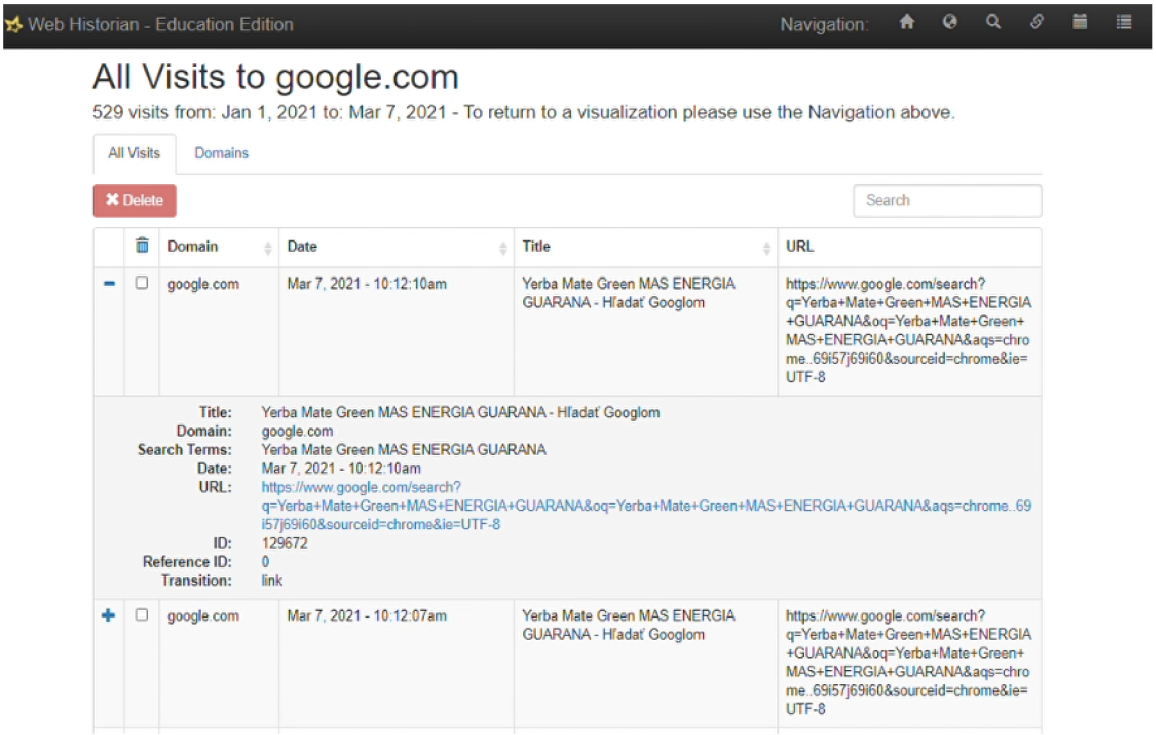


Figure 6.33: Visualized data in *Web Historian* Data table tab

6.5 Analyzers

Analyzers meet the main use case of network forensics. Each analyzer tries to provide the users with the most appropriate analysis and there are some tools that provide a variety of analysis and others that are specified on more specific data.

Some packet analyzers were already demonstrated in the Nitroba scenario in section 6.1. This section will demonstrate other packet analyzers for more specific network data, like attacks or encrypted traffic.

6.5.1 Packet injection attacks

This section demonstrates the detection of the packet injection attacks as it is described in the dataset from section 5.3. This dataset is used to demonstrate tools that can detect packet injection attacks. Both PCAP files from this dataset are used, and all reports are fully available in the attached DVD.

findject.py

findject.py is a python script that can identify injections in a given PCAP file. The results are written to standard output. The output content is the information whether the injection was found or not, 5-touple, sequence number, first, last and modified information.

This tool successfully detected injections in both given PCAP files. After processing the hao123-com_packet-injection.pcap, the injection was found in 5-touple 122.225.98.197:80-192.168.1.254:59360. In the second PCAP file id1-cn_packet-injection.pcap, two injections were found – 42.96.141.35:80-192.168.1.254:59319 and 42.96.141.35:80-192.168.1.254:59320.

HoneyBadger

The *HoneyBadger* is the second tool for the demonstration of detecting packet injections. The same dataset is used. Using the basic configuration of this tool with parameters `-archive_dir=./archive`, `-l=./incoming`, and specified PCAP file, this tool did not detect any injection attack on neither of two tested files.

6.5.2 Encrypted traffic

Wireshark can load the encryption key and then decrypt the encrypted traffic so that it is able to see used protocols. The following demonstration uses *Wireshark* to analyze encrypted traffic datasets described in section 5.5.1. These datasets contain the RSA key or pre-master decryption keys in capture comments. Firstly, the decryption uses RSA key is demonstrated. It is necessary to have the whole handshake captured. Secondly, the demonstration uses a pre-master key for decryption.

rsasnakeoil.cap

This dataset contains an RSA key for the decryption of the SSL traffic. Without using the key in *Wireshark*, whole packets are marked as SSLv3, and it cannot see the encrypted traffic. After loading the correct decryption key into *Wireshark*, the encrypted packet content is visible. The RSA key needs to be added between RSA private keys in Wireshark preferences. The content of this encrypted traffic is HTTP traffic. Decrypted PCAP is saved and available in enclosed DVD.

smtp-ssl.pcapng

The second demonstration of decrypting traffic uses a pre-master key, and the same encrypted SMTP traffic dataset is selected for demonstration. Like the previous dataset, without loading the decryption key into *Wireshark*, the whole traffic is marked as TLSv1.2. After loading the pre-master key, the encrypted SMTP traffic is decrypted and visible as SMTP traffic. Decrypted PCAP is also saved and available in the enclosed DVD.

6.5.3 SMTP traffic

This section demonstrate tool that focuses on SMTP traffic. The *findsmtpinfo.py* is demonstrated. Dataset used in this demonstration is decrypted smtp-ssl.pcapng from section 6.5.2 called smtp-ssl-decrypted.pcap.

findsmtpinfo.py is a tool that processes a given PCAP file, searches for SMTP traffic, and creates a report. This tool uses *tcpflow*. The report contains found flows with detected SMTP traffic together with the raw PCAP file and XML report. From the processed dataset two flows were identified and exported – 127.0.0.1:25 -> 127.0.0.1:58778 and 127.0.0.1:58778 -> 127.0.0.1:25. The report is fully available in the attached DVD.

6.6 Network diagnostic tools

For network forensics, it is also needed to obtain information about the network. The tools that can be marked as diagnostic tools provide the investigator with the requested information of the network. These tools are usually command tools, usually already installed in the OS. The following tools are demonstrated – *ping*, *tracert*, *dig*, *whois*, and *nslookup*. This demonstration is captured in the new created dataset described in section 7.2. The console output of this whole demonstration is also provided in the attached DVD between the demonstration reports.

Each command was performed on IP 147.229.14.1 and hostname google.com.

- *ping* — both targets were reachable; average RTT for IP 147.229.14.1 is 53 ms, for google.com it is 40 ms
- *tracert* — tracing route to 147.229.14.1 took 14 hops, to google.com it was 10 hops
- *dig* — query on 147.229.14.1 returned status NXDOMAIN and one SOA record in authority section; query on google.com returned status NOERROR and one A record in answer section
- *whois* — whois on 147.22914.1 returned inetnum “147.229.0.0 - 147.229.254.255”, netname “VUTBRNET”, desc “Brno University of Technology”, source “RIPE”, and other information; whois on IP 172.217.23.238 (the IP of google.com) returned NetType “Direct Allocation”, OrgName “Google LLC”, and other information
- *nslookup* — nslookup for 147.229.14.1 returned for “1.14.229.147.in-addr.arpa” name “bda-boz.fit.vutbr.cz.”; nslookup for hostname google.com returned non-authoritative answers with IPv4 address 172.217.23.238 and IPv6 address 2a00:1450:4014:80d::200e

6.7 IDS/IPS

IDS and IPS tools are detection and prevention tools that can help identify possible security incidents in real-time by providing some alerts. Another use case of IDS/IPS tools can be analyzing the saved reports and outputs of monitored networks and providing the investigator with possible security incidents. For the demonstration, the *Suricata* tool is used.

Suricata is available for Windows and also Linux. This demonstration uses the Windows version of the *Suricata* tool installed and configured with default settings. This tool works with defined rules — there are several predefined rules, and it is also possible to create customized rules. This demonstration uses only predefined rules.

The following outputs are generated:

- *fast* log — similiar to *snort* fast.log,
- *eve-log* — Extensible Event Format event log in JSON format,
- *stats* — stats for all threads merged together,
- *suricata* log — output of the suricata

Suricata was run on the device connected to the VUT FIT network for a short amount of time — from 30/4/2021 14:56:48 PT to 30/4/2021 15:14:42 PT, and from 30/4/2021 15:17:12 PT to 30/4/2021 15:18:51 PT. The IP of the device is 172.28.106.254. All generated output logs are attached in the enclosed DVD. The following types of events were recorded in the fast.log:

- ET INFO WinHttp AutoProxy Request wpad.dat Possible BadTunnel (Generic Protocol Command Decode),
- ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent (Unknown Traffic),
- ET INFO Windows OS Submitting USB Metadata to Microsoft (Misc activity).

Recorded events with traffic and timestamp can be seen in table 6.5.

Event	Classification	Traffic	Timestamp
ET INFO WinHttp AutoProxy Request wpad.dat Possible BadTunnel	Generic Protocol Command Decode	172.28.106.254:58382 ->147.229.9.23:80	04/30/2021-15:03:33
		172.28.106.254:5838[4-6] ->147.229.9.23:80	04/30/2021-15:03:34
		172.28.106.254:5838[8-9] ->147.229.9.23:80	04/30/2021-15:03:39
		172.28.106.254:58399 ->147.229.9.23:80	04/30/2021-15:03:41
		172.28.106.254:5840[6-7] ->147.229.9.23:80	04/30/2021-15:03:42
		172.28.106.254:584[09-12] ->147.229.9.23:80	04/30/2021-15:03:44
		172.28.106.254:58426 ->147.229.9.23:80	04/30/2021-15:03:47
		172.28.106.254:5843[3-6] ->147.229.9.23:80	04/30/2021-15:03:48
		172.28.106.254:58442 ->147.229.9.23:80	04/30/2021-15:04:41
ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent	Unknown Traffic	172.28.106.254:58383 ->147.229.9.14:3128	04/30/2021-15:03:33
ET INFO Windows OS Submitting USB Metadata to Microsoft	Misc activity	172.28.106.254:58383 ->147.229.9.14:3128	04/30/2021-15:03:33

Table 6.5: Events recorded with *Suricata* tool

6.8 SIEMs

Security Information and Event Management (SIEM) tools are complex tools that can be useful during network forensics since they provide comprehensive security information. In addition to real-time monitoring of many sources, setting rules and actions for specific events, historical search is also available in most SIEMs.

The historical data can be helpful during the investigation since it covers many sources, and using appropriate filters may fasten the investigation. Therefore, SIEMs can act as data sources for network forensics. The filters and rules that can be defined in the SIEMs can cover the analysis phase of the investigation.

Security Event Manager (SEM) tool is used for demonstration of SIEMs. This is a commercial tool that also provides an online demo version. This demo version was used in this demonstration. The demo version is available at https://sem.demo.solarwinds.com/. The user interface contains a dashboard, live and historical events, rules, nodes, and configuration.

The dashboard page provides the user with the most attractive data visualized in graphs and general statistic reports, such as node health, all events, user logons, firewall events, and others. The dashboard is customizable, and each user can edit it according to their requirements — it is possible to add, remove, or move widgets. The dashboard can be seen in figure 6.34.

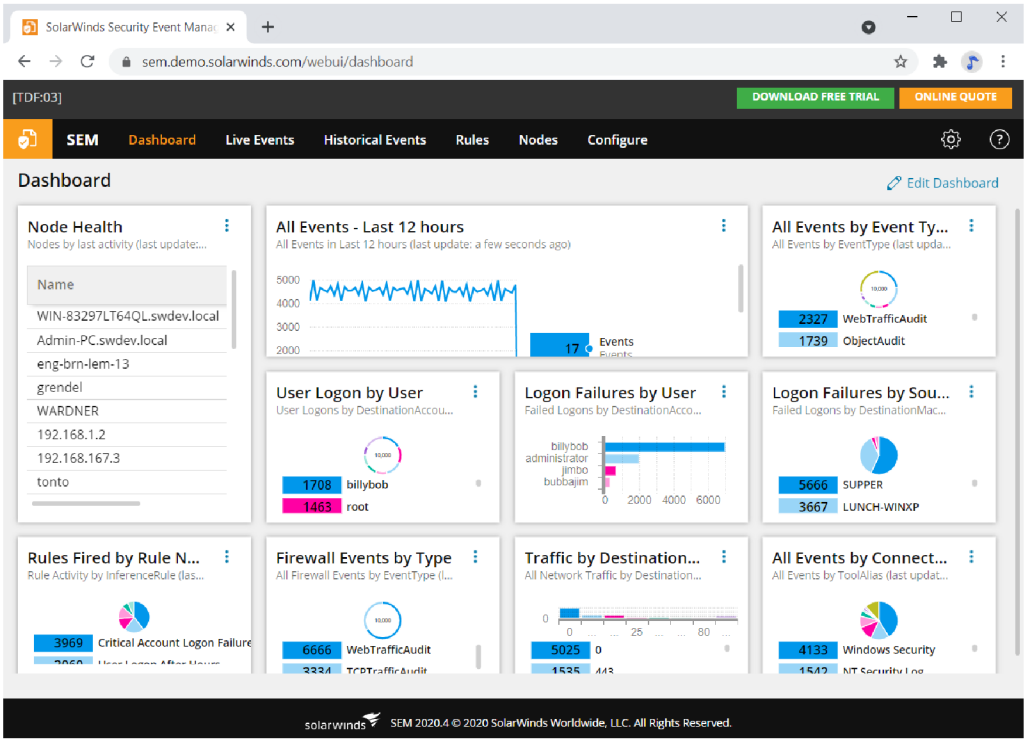


Figure 6.34: Dashboard of *Security Event Manager* tool

In addition to the live capturing and normalization of events, *SEM* also provides historical data. There is also possible to apply filters to these data. Figure 6.35 displays the page with historical data with an applied filter for port scans. *SEM* does not provide only the raw event but uses special alert types for better classification, such as UserLogon, VirusAt-

tack, WebTrafficAudit, TCPPortScan, and others. There is also present the DetectionIP, DetectionTime, EventInfo, SourceAccount, SourceMachine, Protocol, and others based on the specific alert type in the normalized events. Therefore, thanks to this user-friendly normalization, it is easy to understand the events and filter the data.

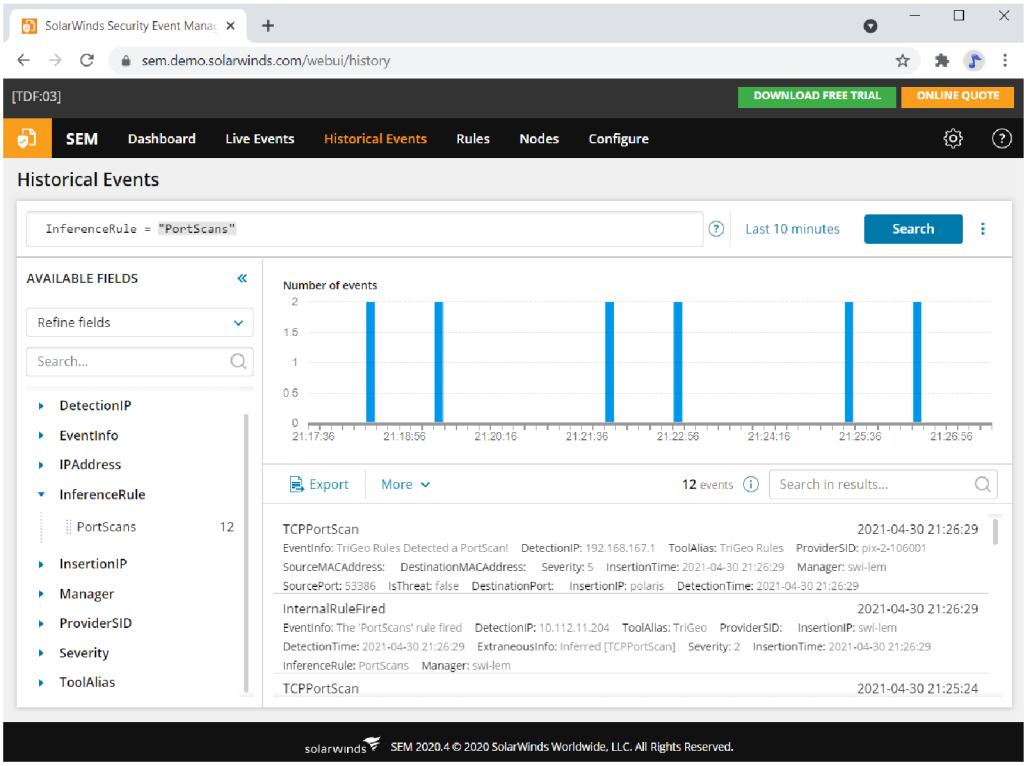


Figure 6.35: Historical data page of *Security Event Manager* tool with applied filter for port scans

The *SEM* also provides the feature of creating and managing rules — performing a specific action based on the appeared events. The action may include blocking the IP address, sending an email, logging off the user, creating an alert, machine shutdown, and others. Figure 6.36 displays the configured rules. For example, it can be seen that rule for logon failures to administrative accounts is set.

Nodes page is used for managing the agent nodes and manager itself — creating nodes, deleting nodes, and managing connectors for these nodes. The node is any device connected to the *SEM* that sends the data. *SEM* supports many types of agent nodes, including Windows, AIX, HP UX, Linux, Mac OS, and Solaris. When the agent node is connected to the manager, several connectors can be configured on this node for data normalization. A connector is a normalizer that normalizes events for a specific device or software. If the user wants to have normalized logs in *SEM*, it is needed to configure the specific connector for that device or software.

In the configuration page, it is able to configure email templates, user-defined groups, users, and directory service groups. Moreover, it is possible to set up log forwarding, threat intelligence, or automatic updates for connectors in the settings.

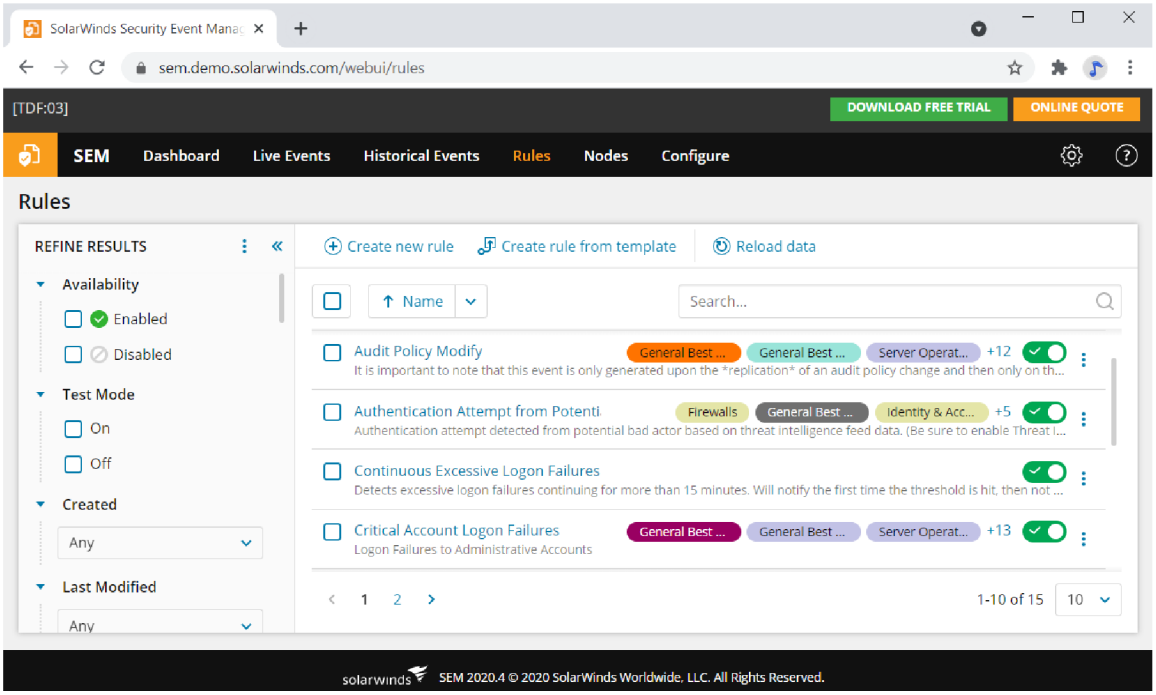


Figure 6.36: Rules page of *Security Event Manager* tool

6.9 Summary and comparison

The common use cases for forensic tools were described. Some of these use cases were also demonstrated, and the results are fully available on the attached DVD. Use cases were demonstrated using the datasets from the survey described in chapter 5. For some use cases, it was needed to create new datasets described in chapter 7. Firstly, network tools were demonstrated in the whole scenario. Secondly, individual designed frequent use cases based on performed surveys were described and demonstrated.

Although the survey of existing taxonomies described in chapter 3 does not implicitly specify use cases of the individual tools, the use cases can be derived from some categories of the taxonomies.

For example, the taxonomy of Davidoff and Ham from section 3.4 describes WAP discovery tools, IDS/IPS, traffic acquisition, and packet and flow analysis. WAP discovery tools were described as a part of the scanners use case, IDS/IPS were also described as a use case in this chapter. Traffic acquisition can be understood as sniffers use case, and packet and flow analysis can be merged into general analyzers use case. Comparing this taxonomy to described use cases, all use cases cover Davidoff and Ham's taxonomy categories.

Moreover, the taxonomy of IJNSA from section 3.2 focuses on their concrete use cases — email forensics, packet sniffers, and web forensics. Comparing to the designed use cases, only sniffers were defined and described. Email and web forensics were not defined as separate use cases but as a part of the general ones.

Taxonomies of Joshi and Pilli from sections 3.5 and 3.3 also contains several use cases that appear between described use cases in this chapter, such as IDS, packet captures, analyzing tools, and vulnerability assessment tool. These taxonomies also mention built-in command tools that are the basis for the described network diagnostic tools in this chapter.

Use cases not mentioned in chapter 3 also appear between these use cases. These are use cases such as visualizers and SIEMs. These use cases were mentioned based on the network tools survey described in chapter 4.

To sum up, described use cases are based on the all performed surveys – network forensic tools taxonomies described in chapter 3, network forensic tools survey described in chapter 4, and network forensic datasets survey described in chapter 5. Existing datasets are the basis for the subsection of the defined use cases. It can also be seen that uses cases also reflect phases of the investigation process described in section 2.2.

Chapter 7

New datasets

This chapter describes newly created datasets that extends the publicly available datasets described in chapter 5. They were used to demonstrate the use cases in chapter 6, and extends datasets with data of today's protocols, application and specific use cases.

Since the existing datasets described in chapter 5 do not contain data captured after 2018, it was needed to create a new dataset with today's traffic that also contains protocols not present in the mentioned datasets. The dataset with today's traffic is described in section 7.2. The applications that appear in this dataset include *WinSCP*, *OpenVPN*, *TeamViewer*, *ESET Secure Browser*, *Zenmap*, *Solarwinds Port Scanner*, *Mail Windows 10*, *ping*, *tracert*, *dig*, and *nslookup*. This dataset also contains traffic of current mobile applications that can contain interesting data. Mobile application like *Ideme vlakom (ZSSK)*, *Mapy.cz*, *Sygic*, *AliExpress*, *Gmail*, *Alza*, and *Strava*, are included. Moreover, a new type of data, such as web browser history data, is included. It is needed to demonstrate specific tools — tools that work with the web browser data. The dataset with web browser history data is described in section 7.3. For a demonstration of sniffers is was necessary to create also dataset that captures the same traffic in the same time frame with multiple tools — this dataset is described in section 7.1.

7.1 Captured traffic with multiple tools

This section describes a dataset of network traffic using multiple tools in the same time. Simple HTTP traffic is captured. This dataset is special in a way that the data are captured with multiple tools in the same time frame. This dataset was creating using the Windows packet capturing tools – NetworkMiner, WinDump, and Wireshark.

In all tools the same interface *Intel(R) Dual Band Wireless-AC 8265* was used for monitoring. The start and the end of the individual captured data differ a bit in seconds, but all of them cover the main period that is from 2021-03-14 01:07:47.265168 UTC to 2021-03-14 01:08:07.382809 UTC.

The table 7.1 displays the basic network information about the interface and the device of captured data. This device is placed in the CET time zone, therefore WinDump and Wireshark use the CET time zone when capturing the data. The NetworkMiner provides only UTC time zone for capturing in the free version, therefore the adjusted time is also displayed in the following annotations of the individual PCAP files. Moreover, the filename of the PCAP file is provided together with the number of packets, used protocols and involved hosts – the common hosts are emphasized.

Network Information		
Interface	Wi-Fi (802.11n) Intel(R) Dual Band Wireless-AC 8265 \Device\NPF_{8B337493-B645-4814-8510-4947B08E553A}	
DNS Servers	213.46.172.38 213.46.172.39	
Gateway	192.168.0.1	
WPAD	185.38.111.1	
Device	**IP**	192.168.0.8
	Operating system	Win 10, 64-bit
	MAC	E4-70-B8-9D-36-EF
	Hostname	TYNNIA

Table 7.1: Network information of the "Simple HTTP" dataset

During the capturing the data the following actions were taken:

1. visited *http://example.com/*
2. visited *https://wis.fit.vutbr.cz/FIT/*; login as a student; visited assessment tab; clicked on "Master's Thesis"; clicked on "Course card" and visited website on FIT server *https://www.fit.vut.cz/study/course/13905/.cs*
3. searched for "network forensics" using *Google*; clicked on first link of *Wikipedia* *https://en.wikipedia.org/wiki/Network_forensics*

The annotation of the individual PCAP files is described in the following part of this section. It includes the filename of the PCAP file that is stored in the attached DVD, the number of packets during the captured time period that is mentioned in the next point of the annotation, and the involved hosts and used protocols.

NetworkMiner

Data are captured in UTC time zone.

- **Filename:** 2021-03-14_networkminer.pcap
- **Number of packets:** 1 761
- **Timeline:**
  - capture: 2021-03-14 03:07:42.871881 CET – 2021-03-14 03:08:12.521182 CET
  - adjusted: 2021-03-14 01:07:42.871881 UTC – 2021-03-14 01:08:12.521182 UTC
- **Hosts:** 3.216.108.82, 3.227.68.65, 13.107.18.11, 18.200.134.29, 18.214.215.174, 20.44.232.74, 35.171.65.67, 35.172.197.47, *35.186.224.25, 35.186.224.47*, 52.112.231.27, 52.114.159.22, *91.198.174.192, 91.198.174.208, 93.184.216.34*, 147.229.8.151, *147.229.9.21, 147.229.9.26*, 151.101.134.248, *172.67.197.229*, 172.217.23.195, 172.217.23.202, 172.217.23.205, *172.217.23.206, 172.217.23.227, 172.217.23.228, 172.217.23.232, 172.217.23.234, 172.217.23.238, 185.38.111.1*, 192.168.0.1, *192.168.0.2, 192.168.0.8, 213.46.172.38, 213.46.172.39*, 216.58.201.67, *216.58.201.78*
- **Protocols:** DNS, HTTP, IGMPv2, LLMNR, MDNS, QUIC, SSDP, TCP, TLSv1, TLSv1.2, TLSv1.3, UDP

WinDump

Data are captured in CET time zone.

- **Filename:** 2021-03-14_windump.pcap
- **Number of packets:** 1 368
- **Timeline:**
  - capture: 2021-03-14 02:07:47.265168 CET – 2021-03-14 02:08:15.620443 CET
  - adjusted: 2021-03-14 01:07:47.265168 UTC – 2021-03-14 01:08:15.620443 UTC
- **Hosts:** *35.186.224.25, 35.186.224.47*, 35.190.242.50, 52.112.231.27, 52.114.104.169, 54.145.80.191, 63.32.146.188, *91.198.174.192, 91.198.174.208, 93.184.216.34*, 142.250.102.188, 147.229.8.151, *147.229.9.21, 147.229.9.26, 172.67.197.229*, 172.217.23.195, 172.217.23.202, 172.217.23.205, *172.217.23.206, 172.217.23.227, 172.217.23.228, 172.217.23.232, 172.217.23.234, 172.217.23.238, 185.38.111.1*, 192.168.0.1, *192.168.0.2, 192.168.0.8, 213.46.172.38, 213.46.172.39, 216.58.201.78*
- **Protocols:** DNS, HTTP, LLMNR, MDNS, QUIC, SSDP, TCP, TLSv1, TLSv1.2, UDP

Wireshark

Data are captured in CET time zone.

- **Filename:** 2021-03-14_wireshark.pcap
- **Number of packets:** 1 493
- **Timeline:**
  - capture: 2021-03-14 02:07:42.124258 CET – 2021-03-14 02:08:09.463342 CET
  - adjusted: 2021-03-14 01:07:42.124258 UTC – 2021-03-14 01:08:09.463342 UTC
- **Hosts:** 3.227.68.65, 13.107.18.11, 34.226.23.237, *35.186.224.25, 35.186.224.47*, 52.114.74.220, 52.114.159.22, 54.145.80.191,*91.198.174.192, 91.198.174.208*, *93.184.216.34, 147.229.9.21, 147.229.9.26*, 151.101.134.248, *172.67.197.229, 172.217.23.206, 172.217.23.227, 172.217.23.228, 172.217.23.232, 172.217.23.234, 172.217.23.238, 185.38.111.1, 192.168.0.2, 192.168.0.8, 213.46.172.38, 213.46.172.39, 216.58.201.78*
- **Protocols:** DNS, HTTP, IGMPv2, LLMNR, MDNS, QUIC, SSDP, TCP, TLSv1, TLSv1.2, TLSv1.3, UDP

7.2 Dataset with today's protocols

This dataset is created to include also today's protocols that are not included in datasets described in chapter 5. *NetworkMiner* tool was used to capture the data in PCAPs with filenames starting with "NM_" and these data are captured in UTC time format. A PCAP "mobile_applications.pcap" was captured using *Wireshark* and data are captured in CET time format.

This dataset reflects common traffic and consists of three PCAP files. First PCAP named "NM_2021-04-17T20-49-16_OpenVPN.pcap" contains mostly OpenVPN traffic. Second PCAP named "NM_2021-04-17T22-00-21_diagnostic-tools.pcap" contains in addition to the mail and WinSCP traffic (the same as in the first PCAP but without VPN) also network traffic generated by some network diagnostic tools. The third PCAP "mobile_applications.pcap" contains the traffic of mobile applications, such as "Ideme vlakom (ZSSK)", "Mapy.cz", "Sygic", "AliExpress", "Gmail", "Alza", and "Strava".

The following subsections describe used scenarios – services and actions taken during the capturing the data, including the annotation of individual PCAP file.

Network information about the local device that was used for data capturing in first two PCAPs is provided in the table 7.2. Table 7.3 displays network information for data capturing of mobile traffic.

Network Information		
Interface	Wi-Fi (802.11n) Intel(R) Dual Band Wireless-AC 8265	
DNS Servers	213.46.172.38 213.46.172.39	
Gateway	192.168.0.1	
WPAD	185.38.111.1	
Device	**IP**	192.168.0.8
	Operating system	Win 10, 64-bit
	MAC	E4-70-B8-9D-36-EF
	Hostname	TYNNIA

Table 7.2: Network information of the dataset with today's protocols

Network Information		
Interface	Wi-Fi Mobile Hotspot from PC Wireless LAN adapter Local Area Connection	
Host PC Device	**IP**	192.168.137.1
	Operating system	Win 10, 64-bit
	MAC	E6-70-B8-9D-36-EF
	Hostname	TYNNIA
Mobile device	**IP**	192.168.137.79
	Operating system	Android
	MAC	74:C1:4F:72:53:56
	Hostname	HUAWEI_P_smart_c751ae46cb

Table 7.3: Network information of the dataset with today's protocols for mobile traffic

7.2.1 OpenVPN traffic

Actions taken in time:

- 20:49 — TeamViewer opened (TeamViewer ID: 813307344) and VM Kali Linux connected (TeamViewer ID: 812526502)
- 20:50 — VM Windows 10 connected (TeamViewer ID: 812564309)

- 20:52 — Opened TeamViewer File Transfer and folder "\home\user\Desktop\scan reports\zenmap\" from TeamViewer Kali VM copied to local PC to destination "C:\Users\marti\FIT\DP\demonstration-reports\Scanners\"
- 20:52 — ESET secure browser opened and visited "google.com" on local PC
- 20:53 — searched for "network forensics" in ESET secure browser and visited first result that directs to wikipedia
- 20:53 — visited "csob.cz" and "Internet banking" in ESET secure browser
- 20:54 — connected to VUT FIT VPN using OpenVPN on local PC
- 20:54 — Zenmap GUI opened and tried to scan 147.229.14.100-110, 147.229.14.1, and 147.229.14.1-10 using Quick scan (scan failed) on local PC
- 20:57 — Solarwinds Port Scanner opened and tried to scan 147.229.14.1 port range 1 – 1024 (ping successful, no result) on local PC
- 20:59 — scanned IP range 147.229.14.140-150 using zenmap on Kali VM on TeamViewer (successful scan)
- 21:00 — Mail Windows 10 opened and sent mail from xzembj00@vutbr.cz to martinka.zembjakova@gmail.com with subject "Test message subject" and body "Test message body" on local PC
- 21:01 — WinSCP opened on local PC
- 21:02 — connected to merlin.fit.vutbr.cz (SFTP, port 22, username: xzembj00)
- 21:03 — create new folder named "delete" in merlin.fit.vutbr.cz at /home/eva/xz/xzembj00/
- 21:05 — transfered two files (test.txt and plan10k.pdf) from local Desktop to "delete" folder
- 21:05 — closed session to merlin.fit.vutbr.cz

Annotation of the PCAP file

- **Filename:** NM_2021-04-17T20-49-16_OpenVPN.pcap
- **Number of packets:** 46 502
- **Timeline:**
  - 2021-04-17 20:49:16.293339 UTC – 2021-04-17 21:06:31.428116 UTC
- **Protocols:** BROWSER, DHCP, DNS, FMTP, HTTP, HTTP/XML, ICMP, IGMPv2, LLMNR, MDNS, NBNS, OCSP, OpenVPN, SMPP, SSDP, SSL, SSLv2, TCP, TLSv1, TLSv1.2, TLSv1.3, UDP

7.2.2 Diagnostic tools traffic

Actions taken in time (all on local PC):

- 22:00 — WinSCP opened and connected to merlin.fit.vutbr.cz (SFTP, port 22, username: xzembj00)
- 22:01 — transfered file plan5k.pdf from local Desktop to "/home/eva/xz/xzembj00/delete" folder on merlin.fit.vutbr.cz
- 22:02 — closed session to merlin.fit.vutbr.cz
- 22:02 — sent mail using Mail Windows 10 application from xzembj00@vutbr.cz to martinka.zembjakova@gmail.com with subject "Test message subject 2" and body "Test message body 2"
- 22:02 — used tool *ping* on IP 147.229.14.1
- 22:03 — used tool *ping* on hostname google.com
- 22:03 — used tool *tracert* on IP 147.229.14.1
- 22:04 — used tool *tracert* on hostname google.com
- 22:05 — used tool *dig* on IP 147.229.14.1 and on hostname google.com using Windows bash console
- 22:06 — used tool *whois* on IP 147.229.14.1 and on hostname google.com using Windows bash console
- 22:07 — used tool *nslookup* on IP 147.229.14.1 and on hostname google.com using Windows bash console

Annotation of the PCAP file

- **Filename:** NM_2021-04-17T22-00-21_diagnostic-tools.pcap
- **Number of packets:** 7 315
- **Timeline:**
  - 2021-04-17 22:00:21.251017 UTC – 2021-04-17 22:07:37.727352 UTC
- **Protocols:** BROWSER, DNS, HTTP, ICMP, IGMPv2, LLMNR, MDNS, NBNS, OCSP, SSDP, SSH, SSHv2, TCP, TLSv1, TLSv1.2, TLSv1.3, UDP, WHOIS

7.2.3 Mobile applications traffic

Actions taken in time (all on mobile device):

- 10:51 — application "Ideme vlakom (ZSSK)" – login to application, browsing for ticket from Bratislava to Kosice, and buying the train ticket
- 10:53 — application "Mapy.cz" – searching for ATMs in the local area, searching for concrete place "Hlavni nadrazi"

- 10:53 — application “Sygic” – searching for banks and ATMs in the local area, creating a route to specific place
- 10:54 — application “AliExpress” – searching for items and putting some items into shopping cart
- 10:59 — application “Gmail” – sending email to “xzembj00@stud.fit.vutbr.cz” with subject “Test message from mobile”
- 11:00 — application “Alza” – searching for items, putting some items into shopping cart, finishing the order
- 11:06 — application “Strava” – looking at activities history

Annotation of the PCAP file

- **Filename:** mobile_applications.pcap
- **Number of packets:** 295 873
- **Timeline:**
  - 2021-05-10 10:51:27.659827 CET – 2021-05-10 11:08:44.728890 CET
- **Protocols:** ARP, BROWSER, CAT-TP, DCP-AF, DNS, HTTP, ICMP, ICMPv6, LLMNR, MDNS, SSDP, SSL, SSLv2, TCP, TLSv1, TLSv1.2, TLSv1.3, UDP

7.3 Web browser history data

This section describes web browser history data. These data are available in attached DVD under the datasets directory in two formats – JSON and XLSX. The filenames of the data are

- “2021-03-07 history.json” and
- “2021-03-07 history.xlsx”.

The date in the filename indicates the date the data was retrieved from the Google Chrome web browser.

The visualized data are described in chapter 6 under the visualizer section 6.4.

The data contains the following information – ID, Last Visit Time, Title, Typed Count, Url, and Visit Count.

The following table 7.4 displays some basic information about the dataset. This table contains the browser type, the number of unique records, first captured record, last captured record, domains from top 10 visited websites, and top 5 typed websites. It can be seen that the time period of the retrieved dataset is approximately two months.

Basic Information	
Browser	Google Chrome (64-bit)
Number of unique records	3 013
Start time	Fri Jan 01 2021 00:42:18 GMT+0100 (CEST)
End time	Sun Mar 07 2021 14:51:21 GMT+0100 (CEST)
Domains from top 10 visited websites	brxt.mendeley.com hyperskill.org jsonformatter.curiousconcept.com gitlab.nesad.fit.vutbr.cz meet.google.com sprintname.cc darce.fnbrno.cz mail.google.com calendar.google.com
Top 5 typed websites	https://www.tiskelnik.cz/ https://www.vutbr.cz/ https://archive.org/ https://mrsushito.cz/ https://darce.fnbrno.cz/

Table 7.4: Basic information about the web browser history dataset

Chapter 8

New taxonomy

This chapter focuses on the newly designed taxonomy. Based on the survey of existing taxonomies described in chapter 3 and their comparison, it was needed to design new taxonomy that would include more tools and contain more classification-based categories and properties that are not present in existing taxonomies.

The taxonomy is based on the existing taxonomies that are described in chapter 3. The main focus is on the use cases that are described in chapter 6. In taxonomies described in chapter 3, categories related to the use of tools are often used. Therefore, this taxonomy is based on the use cases. The following categories are defined:

- *Scanners* — The category based on scanners appears in several existing categories of the survey described in chapter 3 – Davidoff and Ham taxonomy from section 3.4 uses *WAP discovery tools*, and also taxonomy from Joshi and Pilli described in section 3.5 contains *Vulnerability Assessment Tools* category together with the *Network Scanning Tools* category. All these there categories were merged into *Scanners* category and then within this category are distinguished according to the type of scanning.

- *Sniffers* — The category based on the capturing the data appears in all existing taxonomies from chapter 3, and therefore it is present also in the new taxonomy.

- *Visualizers* — None of the existing taxonomies described in chapter 3 focuses on the data visualization as a category. This use case category is deduced from the tools survey in chapter 4 that contains several tool that specializes in visualization itself.

- *Analyzers* — The data analysis also appears in the several taxonomies described in chapter 3. The taxonomy according to ENISA described in section 3.7, Davidoff and Ham taxonomy from section 3.4, and Joshi and Pilli taxonomy from section 3.5 contains category specialized on analyzing.

- *Network diagnostic tools* — This category belong to those ones that are not implicitly specified in the existing taxonomies described in chapter 3. However, the base of this category is on the Joshi and Pilli taxonomy described in section 3.3 that contains command-line tools description.

- *IDS/IPS* — IDS/IPS category also appears in many existing taxonomies from chapter 3. Joshi and Pilli taxonomy from sections 3.5 and 3.5, taxonomy according to ENISA from section 3.7, and taxonomy of Davidoff and Ham from section 3.4 mention the IDS or IPS category.

- *SIEMs* — SIEM tools are not present in the existing taxonomies survey, but according to newly discovered tools described in chapter 4, SIEMs can be useful during the network forensics.

The division of tools into open-source and commercial is also often seen in the existing taxonomies, therefore the *Subscription* category is present in this taxonomy in the most of the use case categories. Other categories that can be denoted as general categories are *User Interface* and *Platform*. The *User Interface* category is based on the taxonomy from IOSR-JCE described in section 3.6 that reflects this classification. Although the classification according to the operating system is not present in taxonomies from chapter 3, in this new taxonomy it belongs to the commonly used generic categories. The *Platform* is included because of the variety of operating systems and platform available for specific tools, based on the tools survey described in chapter 4. This category also provides the possibility of identifying the multiplatformed tools. These three categories are a part of the *Scanners*, *Sniffers*, *Visualizers*, *Analyzers*, and *Network diagnostic tools* use case categories.

The taxonomy structure without the tool examples is visualized in figure 8.1. The structure of this taxonomy consists of one main level that specifies the main category according to the use cases. These categories are visualized with blue color. Each category then contains more specific categories, and one tool can have more than one property from defined categories – can be a part of more than one category. The individual categories are describes in its own section.

The following sections discuss individual categories of the taxonomy. Each category also contains the network tools that are part of that category. Used network tools are tools described in chapter 4 – the tools used in existing network forensic taxonomies discussed in chapter 3 together with some other tools. There were selected only specific tools. Not all tools described in chapter 4 are used in this new taxonomy as examples. Using all the tools described, the taxonomy would be too extensive, and the emphasis is on the categories of taxonomy, not on the tools themselves. However, there is an effort to use more examples for each category than was used for the existing taxonomies in chapter 3. More examples highlight the diversity of tools for each category and expand the possibility of choosing the right tool.

8.1 Scanners

Scanners use case category contains in addition to the three general categories (*Subscription*, *User Interface*, and *Platform*), also the *Type* category that defines the type of the scanner. This category is meant as a division of various types of the scanners, such as wifi, port, web, or vulnerability scanners. The *Scanner* use case category is a general category for all type of scanners that appeared in the existing taxonomies in chapter 3. Scanners category is visualized in table 8.6, including examples of tools.

8.2 Sniffers

Sniffers use case category contains also the mentioned three general categories (*Subscription*, *User Interface*, and *Platform*). Moreover, more specific categories are also included – *Sniffer type*, *Type*, and *Distribution*. The *Sniffer type* specifies whether the tool has only one function and it is a data capturing, or the sniffing is a part of the more complex tool

that can provide the user with more function. The *Type* identifies the type of the capturing – packet sniffers that can capture and store whole packets, sniffers that specialize on flow and stores data in flow format, and sniffers that focus on specific data. Sniffers use case category is visualized in table 8.6, also including examples of tools.

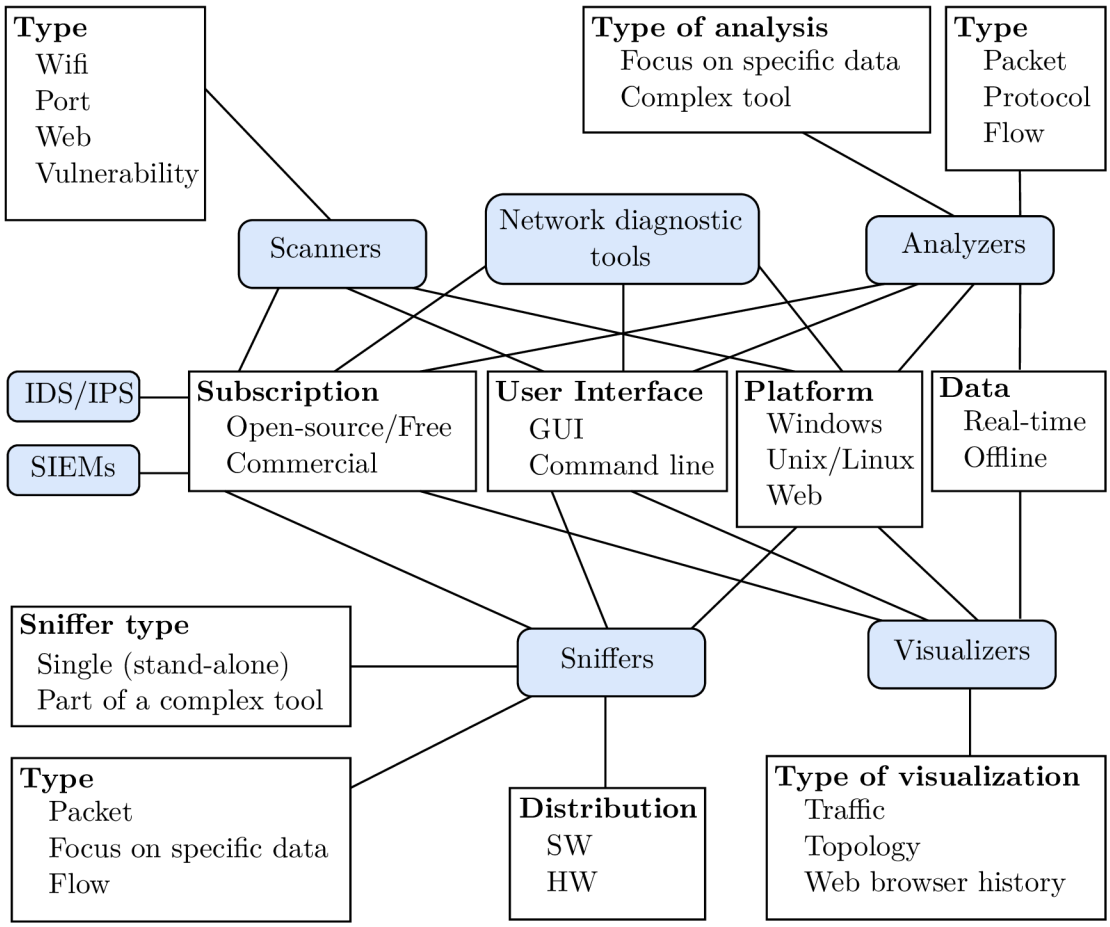


Figure 8.1: New taxonomy structure

8.3 Visualizers

Visualizers use case category is one of the categories that does not appear in the existing categories described in chapter 3. As other categories in this new designed taxonomy, it contains the mentioned three general categories (*Subscription*, *User Interface*, and *Platform*). Another category is the *Data* category that determines whether the tool is capable of processing real-time data, offline data or both. The last category is called *Type of visualization* and it says what type of visualization is provided – visualization of the traffic data, visualization of the network topology, or visualization of the web browser history data. Like other categories, visualizers are also displayed in the table, together with examples of the tools. Table 8.6 provide overview of visualizers within the defined taxonomy.

8.4 Analyzers

Analyzers use case category contains several categories that describe properties of the tools. In addition to the general categories *Subscription*, *User Interface*, and *Platform*, *Type of analysis*, *Type* and *Data* are present. The *Data* category is common with the *Visualizers* use case category.

Type category is based on the "Packet analysis" category of the Davidoff and Ham taxonomy described in section 3.4 – protocol tools, packet tools, and flow tools. *Flow* category contains as tools that can provide flow analysis output as tools that can work with flow data.

Similar to the visualizers, *Data* category appears. This category is useful to identify the tools that are able to work with real-time data, offline data, or both.

Newly created category named *Type of analysis* does not appear in existing taxonomies from the survey. This category specifies the tools that can process many types of data and can provide various types of analysis. These tools can also be called as complex tools and this category also includes tools referred to as NFATs in the existing taxonomies of Pilli and Joshi described in sections 3.3 and 3.5. Other type of analysis contains tools that that focuses on specific type of data, such as specific protocol or specific type of analysis.

Tools that are part of the analyzers use case category are displayed in tables 8.6 and 8.6, including displayed properties of each subcategory as defined in the taxonomy.

8.5 Network diagnostic tools

Network diagnostic tools is a completely new category that includes mostly command-line tools. It is based on the command tools described in Pilli and Joshi taxonomy from section 3.3, but it is also extended to tools with a graphical user interface. All tools present in this category are useful for network diagnostics or obtaining other basic information about the network.

This category contains only the basic categories such as *Subscription*, *User Interface*, and *Platform* categories. Table 8.6 provides an overview of tools that belong to this category within designed taxonomy.

8.6 IDS/IPS and SIEMs

Both *IDS/IPS* and *SIEMs* have only the generic category *Subscription* to identify free and proprietary tools.

Table 8.7 provides an overview of IDS/IPS tools and table 8.8 focuses on SIEM tools that belong to these categories within designed taxonomy.

Scanners	Subscription		User interface		Platform		Type			
	Open-source/ Free	Commercial	GUI	Command line	Windows	Unix/ Linux	Wifi	Port	Web	Vulnerability
Acunetix Web Vulnerability Scanner		X	X		X	X			X	
Angry IP Scanner	X		X		X	X		X		
Kismet	X		X	X	X	X	X			
Metasploit	X	X		X	X	X				X
Nessus	X	X	X		X	X				X
NetStumbler	X		X		X		X			
Nikto	X			X		X			X	X
Nmap	X			X	X	X		X		X
Port Scanner	X		X		X			X		
Wikto	X			X	X				X	X
Wireless Network Watcher	X		X		X		X			
Zenmap	X		X		X	X		X		X

Table 8.1: Scanners category from the taxonomy

Sniffers	Subscription		User interface		Platform		Sniffer type		Distribution		Type		
	Open-source/ Free	Commercial	GUI	Command line	Windows	Unix/ Linux	Single (stand-alone)	Part of a complex tool	SW	HW	Packet	Focus on specific data	Flow
aircrack-ng	X			X	X	X		X	X		X		
AirPcap		X	X		X		X			X	X		
Argus	X			X	X	X		X	X				X
dumpcap	X			X	X	X	X		X		X		
flow-tools	X			X		X		X	X				X
Kismet	X		X	X	X	X		X	X		X		
NetworkMiner	X	X	X		X			X	X		X		
NfDump	X			X		X		X	X				X
ngrep	X			X	X	X		X	X		X		
PassiveDNS	X			X		X		X	X		X	X	
SiLK	X			X		X		X	X				X
Steganographer	X			X		X		X	X		X		
tcpdump	X			X		X	X		X		X		
TCPFlow	X			X	X	X	X		X				X
Windump	X			X	X		X		X		X		
windump	X			X	X		X		X		X		
Wireshark	X		X		X	X		X	X		X		

Table 8.2: Sniffers category from the taxonomy

Visualizers	Subscription		User interface		Platform			Data		Type of visualization		
	Open-source/ Free	Commercial	GUI	Command line	Web	Windows	Unix/ Linux	Real-time	Offline	Traffic	Topology	Web browser history
CapAnalysis	X		X		X		X		X	X		
EtherApe	X		X				X	X	X	X		
NetScanTools (Graphical Ping)	X	X	X			X		X		X		
Network Performance Monitor		X	X		X	X		X		X	X	
Network Topology Mapper		X	X			X		X			X	
NfSen	X		X	X	X			X	X	X		
PcapXray	X		X				X		X	X		
VisualRoute		X	X			X		X		X	X	
Web Historian	X		X		X				X			X

Table 8.3: Visualizers category from the taxonomy

Analyzers	Subscription		User interface		Platform		Type			Data		Type of analysis	
	Open-source/ Free	Commercial	GUI	Command line	Windows	Unix/ Linux	Packet	Protocol	Flow	Real-time	Offline	Focus on specific data	Complex tool
Argus	X			X	X	X			X	X	X	X	
Bless	X		X		X	X	X				X	X	
CapLoader		X	X		X		X		X		X	X	
DoHlyzer	X		X		X	X	X		X	X	X	X	
Dshell	X			X	X	X	X			X	X		X
EmailTrackerPro		X	X		X		X				X	X	
findsmtpinfo.py	X			X	X	X	X				X	X	
flow-tools	X			X		X			X	X	X	X	
FlowTraq		X	X		X	X			X	X	X	X	
Forensics Investigation Toolkit		X	X		X		X			X	X		X
HoneyBadger	X			X		X	X			X	X	X	
Iris		X	X		X		X						X
NetFlow Traffic Analyzer		X	X		X				X	X		X	
Netfox Detective	X		X		X		X			X	X		X
NetworkMiner	X	X	X		X		X			X	X		X

Table 8.4: Analyzers category from the taxonomy (part 1)

Analyzers	Subscription		User interface		Platform		Type			Data		Type of analysis	
	Open-source/ Free	Commercial	GUI	Command line	Windows	Unix/ Linux	Packet	Protocol	Flow	Real-time	Offline	Focus on specific data	Complex tool
ngrep	X			X	X	X	X			X	X	X	
OmniPeek		X	X		X		X	X	X	X	X		X
PassiveDNS	X			X		X	X			X	X	X	
pcapcat	X			X	X	X	X		X		X	X	
SiLK	X			X		X			X		X	X	
smtpdump	X			X		X	X				X	X	
softflowd	X			X		X			X	X	X	X	
SSLsplit	X			X		X	X			X	X	X	
TCPFlow	X			X	X	X			X	X	X	X	
TCPStat	X			X		X	X			X	X	X	
TCPXtract	X			X		X	X		X	X	X	X	
tshark	X			X	X	X	X	X	X	X	X	X	
Wireshark	X		X		X	X	X	X	X	X	X		X
Xplico	X		X			X	X				X		X
YAF	X			X		X			X	X	X	X	

Table 8.5: Analyzers category from the taxonomy (part 2)

Network diagnostic tools	Subscription		User interface		Platform	
	Open-source/Free	Commercial	GUI	Command line	Windows	Unix/Linux
ARP	X			X	X	X
chrootkit	X			X	X	X
dig	X			X	X	X
ifconfig	X			X		X
IP Address Manager		X	X		X	
IP Address Tracker	X		X		X	
ipconfig	X			X	X	
nbtstat	X			X	X	
netcat	X			X		X
NetScanTools	X	X	X		X	
netstat	X			X	X	X
nslookup	X			X	X	X
p0f	X			X		X
ping	X			X	X	X
SmartWhois		X	X		X	
traceroute	X			X		X
tracert	X			X	X	
whois	X			X	X	X

Table 8.6: Network diagnostic tools category from the taxonomy

IDS/IPS	Subscription	
	Open-source/Free	Commercial
Bricata		X
Check Point IPS		X
Corero Network Security		X
Extreme Network IPS		X
PADS	X	
snort	X	
Suricata	X	
Zeek	X	

Table 8.7: IDS/IPS category from the taxonomy

SIEMs	Subscription	
	Open-source/Free	Commercial
LogRhythm NetMon		X
LogRhythm NetMon Freemium	X	
Security Event Manager		X
Splunk		X

Table 8.8: SIEMs category from the taxonomy

Chapter 9

Conclusion

The aim of this project was to research the existing taxonomies of network forensic tools, existing network forensic tools, and available datasets. Firstly, after a brief introduction to network forensics, found taxonomies were described and compared. Secondly, network tools from the literature survey together with other found tools were described. Furthermore, some available datasets were described in detail, including their comparison.

Moreover, frequent use cases for forensic tools were designed and demonstrated using the found datasets, including comparing the results on the use cases mentioned in the literature. Further, new datasets with missing/updated network protocols and use cases were created. Last but not least, this thesis designed new taxonomy based on existing ones and the obtained information. The additional value of the thesis is publicly available output using GitHub pages that represents the overview of available network tools that can be used in network forensics extended with some publicly available datasets and their description.

This work required the study of a large amount of literature. It was necessary to identify existing taxonomies, study them in detail, describe and compare them. Furthermore, it was necessary to get acquainted with many tools and find tools that were not mentioned in existing taxonomies. Moreover, it was needed to describe and analyze tools, demonstrate their use cases on datasets, and compare and describe them in the newly created taxonomy. Besides, gathering, analyzing, and comparing datasets took some time. Information about tools and datasets was collected and clearly described in overviews also published as GitHub Pages website. Overall, this work required a great deal of study, acquisition, analysis, demonstration, and comparison.

9.1 Website representing the results

The website for summarising the gained information was created to represent the results from the performed surveys in chapter 4 and 5. GitHub pages environment was used. This website is publicly available on link https://martinazembjakova.github.io/Network-forensic-tools-taxonomy/ or directly on GitHub[1].

This website acts as an overview of available network tools that can be used during the network forensics described in chapter 4 and provides a brief description of each tool together with the proper links, such as official website, link to GitHub repository, and

[1] https://github.com/MartinaZembjakova/Network-forensic-tools-taxonomy

documentation. This website also contains some available datasets described in chapter 5 with a detailed description of each dataset and links.

The design of this website hosted on GitHub pages can be seen in appendix C.

This website has been posted in several digital forensics communities and has also been indexed on Google search since April 27, 2021. Google Analytics is also configured, and there were 731 page views from 46 countries in the first three weeks of publishing the website.

Bibliography

Network forensic tool references

[1] *IPTraf - An IP Network Monitor*. 2005. Available at: http://iptraf.seul.org/.

[2] *Passive Asset Detection System*. 2005. Available at: http://passive.sourceforge.net/.

[3] *tcpxtract*. 2005. Available at: http://tcpxtract.sourceforge.net/.

[4] *Systenance Software - Index.dat Analyzer*. 2006. Available at: https://www.systenance.com/indexdat.php.

[5] *The GNU Netcat*. 2006. Available at: http://netcat.sourceforge.net/.

[6] *Sebek | The Honeynet Project*. 2008. Available at: https://web.archive.org/web/20190921171854/http://www.honeynet.org/project/sebek/.

[7] *SKYHOOK Wireless*. 2008. Available at: https://web.archive.org/web/20080604163052/http://www.skyhookwireless.com/howitworks/.

[8] *whois linux command man page*. 2009. Available at: https://www.commandlinux.com/man-page/man1/whois.1.html.

[9] *Administration Guide IPS-1 Sensor R71*. 2010. Available at: http://supportcontent.checkpoint.com/documentation_download?ID=10505.

[10] *SilentRunner Sentinel*. AccessData, 2010. Available at: https://silo.tips/download/silentrunner-brochure.

[11] *CIRT: THE ONLY SOLUTION TO INTEGRATE NETWORK ANALYSIS, HOST ANALYSIS and DATA AUDITING*. AccessData, 2011.

[12] *NetStumbler*. 2011. Available at: http://www.stumbler.net/.

[13] *NfSen*. 2011. Available at: http://nfsen.sourceforge.net/.

[14] *SilentRunner Sentinel Network Forensics Software*. 2011. Available at: https://web.archive.org/web/20111128113603/http://accessdata.com/products/cyber-security-incident-response/ad-silentrunner-sentinel.

[15] *Introduction to Cisco IOS NetFlow*. Cisco, 2012. Available at: https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.pdf.

[16] *AirPcap User's Guide*. May. Riverbed Technology, 2013.

[17] *Extreme Networks Announces Agreement to Acquire Enterasys Networks*. Extreme Networks, Inc., september 2013.

[18] *Forensic and Log Analysis GUI*. 2013. Available at: https://sourceforge.net/projects/pyflag/.

[19] *eMailTrackerPro*. 2014. Available at: http://www.emailtrackerpro.com/.

[20] *Haka*. 2014. Available at: http://www.haka-security.org/.

[21] *Haka's User Guide*. 2014. Available at: http://doc.haka-security.org/haka/release/v0.3.0/doc/user/userindex.html.

[22] *OpenCarnivore*. 2014. Available at: https://web.archive.org/web/20141110081046/http://opencarnivore.org/.

[23] *Sourcefire 3D System User Guide*. Cisco, 2014. Available at: https://www.cisco.com/c/dam/en/us/td/docs/security/sourcefire/3d-system/53/Sourcefire_3D_System_User_Guide_v53.pdf.

[24] *The Check Point IPS Solution*. 2014. Available at: https://sc1.checkpoint.com/documents/R76/CP_R76_IPS_AdminGuide/12742.htm.

[25] *HoneyBadger*. 2015. Available at: https://honeybadger.readthedocs.io/en/latest/.

[26] *InfiniStream Appliance*. NetScout Systems, Inc., 2015. Available at: https://www.netscout.com/sites/default/files/2015/06/netscout_ql_infinistream_appliance.pdf.

[27] *NetDetector DATASHEET*. NIKSUN, 2015. Available at: https://www.phoenixdatacom.com/wp-content/uploads/2015/09/NIKSUNDatasheet_NetDetector_pdl.pdf.

[28] *Netfox Detective*. 2015. Available at: http://netfox.fit.vutbr.cz/About.en.html.

[29] *Network forensics - Forensics Wiki*. 2015. Available at: https://forensicswiki.xyz/wiki/index.php?title=Network_forensics.

[30] *TCPDstat*. 2015. Available at: https://web.archive.org/web/20150808152314/https://staff.washington.edu/dittrich/talks/core02/tools/.

[31] *Tools:Network Forensics - Forensics Wiki*. 2016. Available at: https://forensicswiki.xyz/wiki/index.php?title=Tools:Network_Forensics.

[32] *arp*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/arp.

[33] *ipconfig*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ipconfig.

[34] *nbtstat | Microsoft Docs*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/nbtstat.

[35] *NetIntercept DATASHEET*. NIKSUN, 2017. Available at: https://www.neox-networks.com/downloads/NIKSUNDatasheet_NetIntercept.pdf.

[36] *netstat | Microsoft Docs*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/netstat.

[37] *nslookup | Microsoft Docs*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/nslookup.

[38] *tracert | Microsoft Docs*. 2017. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/tracert.

[39] *FireSIGHT System User Guide Version 5.4.1*. 2018. Available at: https://www.cisco.com/c/en/us/td/docs/security/firesight/541/user-guide/FireSIGHT-System-UserGuide-v5401/Intro-Preface.html.

[40] *ping | Microsoft Docs*. 2018. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ping.

[41] *Security Analytics Hardware End of Life*. Symantec, 2018. Available at: http://www.dhitech.co.kr/Service_Resource/pdf/SecurityAnalyticsHardware.pdf.

[42] *WinDump*. 2018. Available at: https://www.winpcap.org/windump/.

[43] *WinPcap*. 2018. Available at: https://www.winpcap.org/.

[44] *Xplico - Open Source Network Forensic Analysis Tool (NFAT)*. 2018. Available at: https://www.xplico.org/.

[45] *CapAnalysis*. 2019. Available at: https://www.capanalysis.net/ca/.

[46] *Netfox Detective*. 2019. Available at: https://github.com/nesfit/NetfoxDetective.

[47] *PcapXray*. 2019. Available at: https://github.com/Srinivas11789/PcapXray.

[48] *Sebek project site*. 2019. Available at: https://web.archive.org/web/20190304212247/http://projects.honeynet.org/sebek.

[49] *Zeek User Manual v3.2.2*. 2019. Available at: https://docs.zeek.org/en/current/intro/index.html.

[50] *Acunetix | Web Application Security Scanner*. 2020. Available at: https://www.acunetix.com/.

[51] *Aircrack-ng*. 2020. Available at: https://www.aircrack-ng.org/.

[52] *Angry IP Scanner - Documentation*. 2020. Available at: https://angryip.org/documentation/.

[53] *Angry IP Scanner - the original IP scanner for Windows, Mac and Linux*. 2020. Available at: https://angryip.org/.

[54] *arp(8)*. 2020. Available at: https://linux.die.net/man/8/arp.

[55] *BIND 9 - ISC*. 2020. Available at: https://www.isc.org/bind/.

[56] *Bricata - Network Detection & Response. Analytics. Threat Hunting.* 2020. Available at: https://bricata.com/.

[57] *Cisco IOS NetFlow.* 2020. Available at: https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

[58] *Corero Products.* 2020. Available at: https://www.corero.com/products/.

[59] *Decision Group - E-Detective.* 2020. Available at: http://www.edecision4u.com/E-DETECTIVE.html.

[60] *dumpcap - The Wireshark Network Analyzer 3.4.0.* 2020. Available at: https://www.wireshark.org/docs/man-pages/dumpcap.html.

[61] *Enterprise Intrusion Prevention (IPS) Software & Solutions.* 2020. Available at: https://www.trendmicro.com/en_us/business/products/network/intrusion-prevention.html.

[62] *EtherApe, a graphical network monitor.* 2020. Available at: https://etherape.sourceforge.io/.

[63] *flow-tools(1) - Linux man page.* 2020. Available at: https://linux.die.net/man/1/flow-tools.

[64] *FlowTraq.* 2020. Available at: https://www.flowtraq.com/.

[65] *FlowTraq Exporter.* 2020. Available at: https://www.flowtraq.com/product/flow-exporter/.

[66] *Forensics Investigation Toolkit (FIT).* 2020. Available at: http://www.edecision4u.com/FIT.html.

[67] *Free WiFi scanner and security software for Mac - KisMAC.* 2020. Available at: https://kismac-ng.org/.

[68] *ifconfig(8): configure network interface - Linux man page.* 2020. Available at: https://linux.die.net/man/8/ifconfig.

[69] *Intrusion Prevention System - IPS | Check Point Software.* 2020. Available at: https://www.checkpoint.com/products/intrusion-prevention-system-ips/.

[70] *Iris Network Traffic Analyzer 5.2.0.74.* 2020. Available at: https://www.malavida.com/en/soft/iris-network-traffic-analyzer/#gref.

[71] *Kismet.* 2020. Available at: https://www.kismetwireless.net/.

[72] *Metasploit | Penetration Testing Software, Pen Testing Security.* 2020. Available at: https://www.metasploit.com/.

[73] *Metasploit Documentation.* 2020. Available at: https://docs.rapid7.com/metasploit/.

[74] *Nessus Vulnerability Assessment.* 2020. Available at: https://www.tenable.com/products/nessus.

[75] *Netcat: the TCP/IP swiss army*. 2020. Available at: https://nc110.sourceforge.io/.

[76] *Netcat – SecTools Top Network Security Tools*. 2020. Available at: https://sectools.org/tool/netcat/.

[77] *NetScanTools Basic Edition*. 2020. Available at: https://www.netscantools.com/nstbasicmain.html.

[78] *NetScanTools Pro Edition*. 2020. Available at: https://www.netscantools.com/nstpromain.html.

[79] *netstat(8) - Linux man page*. 2020. Available at: https://linux.die.net/man/8/netstat.

[80] *NetVCR DATASHEET*. NIKSUN, 2020. Available at: https://www.niksun.com/c/1/ds/NIKSUNDatasheet_NetVCR.pdf.

[81] *Network Flight Recorder*. 2020. Available at: https://github.com/alphasoc/nfr.

[82] *NetworkMiner - The NSM and Network Forensics Analysis Tool*. 2020. Available at: https://www.netresec.com/?page=Networkminer.

[83] *nGeniusONE*. 2020. Available at: https://www.netscout.com/product/ngeniusone-platform.

[84] *Niksun NetDetector - Packet capture & network security forensics*. 2020. Available at: https://www.phoenixdatacom.com/product/niksun-netdetector-packet-capture-network-security-forensics/.

[85] *NIKSUN NetVCR Suite - Application and Network Performance Monitoring*. 2020. Available at: https://www.niksun.com/product.php?id=110.

[86] *Nikto2*. 2020. Available at: https://cirt.net/Nikto2.

[87] *Nmap: the Network Mapper*. 2020. Available at: https://nmap.org/.

[88] *nslookup(1) - Linux man page*. 2020. Available at: https://linux.die.net/man/1/nslookup.

[89] *ntop – High Performance Network Monitoring Solutions based on Open Source and Commodity Hardware*. 2020. Available at: https://www.ntop.org/.

[90] *Omnipeek Network Protocol Analyzer*. 2020. Available at: https://www.liveaction.com/products/omnipeek-network-protocol-analyzer/.

[91] *openargus*. 2020. Available at: https://openargus.org/.

[92] *OWASP WebScarab Project*. 2020. Available at: https://wiki.owasp.org/index.php/Category:OWASP_WebScarab_Project.

[93] *ping(8) - Linux man page*. 2020. Available at: https://linux.die.net/man/8/ping.

[94] *pyflag*. 2020. Available at: https://code.google.com/archive/p/pyflag/.

[95] *PyFlag - Forensics Wiki*. 2020. Available at: https://forensicswiki.xyz/wiki/index.php?title=PyFlag.

[96] *Riverbed AirPcap*. 2020. Available at: https://support.riverbed.com/content/support/software/steelcentral-npm/airpcap.html.

[97] *RSA NetWitness Platform: Threat detection and response*. 2020. Available at: https://www.rsa.com/en-us/products/threat-detection-response.

[98] *sFlow*. 2020. Available at: https://sflow.org/index.php.

[99] *SiLK*. 2020. Available at: https://tools.netsa.cert.org/silk/.

[100] *SmartWhois*. 2020. Available at: https://www.tamos.com/products/smartwhois/.

[101] *Snort - Network Intrusion Detection & Prevention System*. 2020. Available at: https://www.snort.org/.

[102] *softflowd*. 2020. Available at: https://code.google.com/archive/p/softflowd/.

[103] *SplitCap*. 2020. Available at: https://www.netresec.com/?page=SplitCap.

[104] *Stenographer*. 2020. Available at: https://github.com/google/stenographer.

[105] *TCPDUMP/LIBPCAP public repository*. 2020. Available at: http://www.tcpdump.org/.

[106] *tcptrace*. 2020. Available at: http://www.tcptrace.org/.

[107] *The Zeek Network Security Monitor*. 2020. Available at: https://zeek.org/.

[108] *traceroute(8) - Linux man page*. 2020. Available at: https://linux.die.net/man/8/traceroute.

[109] *tshark - The Wireshark Network Analyzer 3.4.0*. 2020. Available at: https://www.wireshark.org/docs/man-pages/tshark.html.

[110] *VisualRoute - Traceroute and Reverse trace - Traceroute and Network diagnostic tools*. 2020. Available at: http://www.visualroute.com/.

[111] *Web Historian*. 2020. Available at: https://www.webhistorian.org/.

[112] *Whois - Windows Sysinternals*. 2020. Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/whois.

[113] *Wikto – how does it work and how do I use it?* 2020. Available at: https://raw.githubusercontent.com/sensepost/wikto/master/Documentation/using_wikto.pdf.

[114] *Wireless Network Watcher - Show who is connected to your wireless network*. 2020. Available at: https://www.nirsoft.net/utils/wireless_network_watcher.html.

[115] *Wireshark*. 2020. Available at: https://www.wireshark.org/.

[116] *Wireshark User's Guide*. 2020. Available at: https://www.wireshark.org/download/docs/user-guide.pdf.

[117] *YAF*. 2020. Available at: https://tools.netsa.cert.org/yaf/.

[118] *CapLoader*. 2021. Available at: https://www.netresec.com/?page=CapLoader.

[119] *Extreme Networks Intrusion Prevention System (IPS)*. 2021. Available at: https://www.netsolutionstore.com/IPS.asp.

[120] *findject.py*. 2021. Available at: https://www.netresec.com/?page=findject.

[121] *IBM Security Network Intrusion Prevention System (IPS) V4.6.2 documentation*. 2021. Available at: https://www.ibm.com/support/knowledgecenter/SSB2MG_4.6.2/com.ibm.ips.doc/NIPS_product_landing.htm.

[122] *IP Address Manager (IPAM)*. 2021. Available at: https://www.solarwinds.com/ip-address-manager.

[123] *IP Address Tracker*. 2021. Available at: https://www.solarwinds.com/free-tools/ip-address-tracker.

[124] *Log Analyzer*. 2021. Available at: https://www.solarwinds.com/log-analyzer.

[125] *LogRhythm NetMon*. 2021. Available at: https://logrhythm.com/products/logrhythm-netmon/.

[126] *LogRhythm NetMon Freemium*. 2021. Available at: https://logrhythm.com/products/logrhythm-netmon-freemium/.

[127] *NetFlow Configurator*. 2021. Available at: https://www.solarwinds.com/free-tools/netflow-configurator.

[128] *NetFlow Traffic Analyzer*. 2021. Available at: https://www.solarwinds.com/netflow-traffic-analyzer.

[129] *Network Performance Monitor*. 2021. Available at: https://www.solarwinds.com/network-performance-monitor.

[130] *Network Topology Mapper*. 2021. Available at: https://www.solarwinds.com/network-topology-mapper.

[131] *NIKSUN NetDetector Suite*. 2021. Available at: https://www.niksun.com/product.php?id=112.

[132] *Port Scanner*. 2021. Available at: https://www.solarwinds.com/free-tools/port-scanner.

[133] *Security Event Manager*. 2021. Available at: https://www.solarwinds.com/security-event-manager.

[134] *Skyhook*. 2021. Available at: https://www.skyhook.com/.

[135] *Splunk*. 2021. Available at: https://www.splunk.com/.

[136] *Suricata*. 2021. Available at: https://suricata-ids.org/.

[137] Adams, J. *tcpdstat*. 2010. Available at: https://github.com/netik/tcpdstat.

[138] B. Claise, B. Trammell and P. Aitken. *RFC 7011 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*. 2013. ISSN 2070-1721. Available at: https://tools.ietf.org/html/rfc7011.

[139] Balas, E., Song, C. and A. Davis, M. *sebek*. 2013. Available at: https://github.com/honeynet/sebek.

[140] Cohen, M. I. PyFlag - An advanced network forensic framework. *Digital Investigation*. Elsevier Ltd. september 2008, vol. 5, p. S112–S120. DOI: 10.1016/j.diin.2008.05.016. ISSN 17422876.

[141] Dignan, L. EMC acquires NetWitness, combines with RSA. *ZDNet: Between the Lines*. april 2011. Available at: https://www.zdnet.com/article/emc-acquires-netwitness-combines-with-rsa/.

[142] Fjellskal, E. B. *PassiveDNS*. 2020. Available at: https://github.com/gamelinux/passivedns.

[143] Frantzis, A. *Bless - Gtk# Hex Editor*. 2020. Available at: https://github.com/afrantzis/bless.

[144] Garfinkel, S. *TCPFLOW 1.5.0*. 2018. Available at: https://github.com/simsong/tcpflow.

[145] Garfinkel, S. L. and Shick, M. *Passive TCP Reconstruction and Forensic Analysis with tcpflow*. September 2013. Available at: http://hdl.handle.net/10945/36026.

[146] Gudjonsson, K. *oftcat*. 2009. Available at: https://github.com/kiddinn/misc/blob/master/scripts/oftcat.

[147] Gudjonsson, K. *pcapcat*. 2009. Available at: https://github.com/kiddinn/misc/blob/master/scripts/pcapcat.

[148] Guenichot, F. *smtpdump*. 2009. Available at: http://malphx.free.fr/dotclear/public/nfpc2/smtpdump.

[149] Haag, P. *nfdump*. 2020. Available at: https://github.com/phaag/nfdump.

[150] Herman, P. *tcpstat*. 2009. Available at: https://frenchfries.net/paul/tcpstat/.

[151] Klassen, F. *tcpreplay*. 2020. Available at: http://tcpreplay.appneta.com/.

[152] Klassen, F. and Turner, A. *Tcpreplay*. 2020. Available at: https://github.com/appneta/tcpreplay.

[153] Kuranda, S. *Trend Micro To Acquire HP TippingPoint For $300M*. October 2015. Available at: https://www.crn.com/news/security/300078537/trend-micro-to-acquire-hp-tippingpoint-for-300m.htm.

[154] Lashkari, A. H. *DoHlyzer*. 2020. Available at: https://github.com/ahlashkari/DoHlyzer.

[155] Murilo Nelson and Steding-Jessen Klaus. *chkrootkit*. 2020. Available at: http://www.chkrootkit.org/.

[156] Omella, A. A. and Berrueta, D. B. *yersinia: A framework for layer 2 attacks*. 2017. Available at: https://github.com/tomac/yersinia.

[157] Ostermann, S. *tcptrace*. 2001. Available at: https://github.com/blitz/tcptrace.

[158] Ovsienko, D., Le Bail, F.-X., Harris, G. and Richardson, M. *tcpslice*. 2020. Available at: https://github.com/the-tcpdump-group/tcpslice/.

[159] Pluskal, J., Breitinger, F. and Ryšavý, O. Netfox detective: A novel open-source network forensics analysis tool. *Forensic Science International: Digital Investigation*. Elsevier Ltd. 2020, vol. 35, p. 1–13. DOI: 10.1016/j.fsidi.2020.301019. ISSN 26662817.

[160] Ritter, J. *ngrep 1.47 (9.7.2017)*. 2017. Available at: https://github.com/jpr5/ngrep/.

[161] Roethlisberger, D. *SSLsplit - transparent SSL/TLS interception*. 2019. Available at: https://www.roe.ch/SSLsplit.

[162] Rossi, J. *findsmtpinfo.py*. 2016. Available at: https://web.archive.org/web/20200522224019/http://forensicscontest.com/contest02/Finalists/Jeremy_Rossi/.

[163] Schmidt, B., Plotts, C., Hanan, D. and Blois, P. *Snoop*. 2020. Available at: https://github.com/snoopwpf/snoopwpf.

[164] Smith J Allen Crider Henry H Perritt, S. P. and Mengfen Shyong Harold Krent Larry Reynolds Stephen Mencik, J. L. *Independent Review of the Carnivore System Final Report*. IIT Research Institute, 2000.

[165] Stainton, D. *HoneyBadger*. 2019. Available at: https://github.com/david415/HoneyBadger.

[166] Sullo, C. and Lodge, D. *nikto: Nikto web server scanner*. 2020. Available at: https://github.com/sullo/nikto.

[167] Temmingh, R., Phillips, G., Villiers, I. d. and White, D. *wikto: Nikto for Windows with some extra features*. 2015. Available at: https://github.com/sensepost/wikto.

[168] United States Government. *Dshell*. 2020. Available at: https://github.com/USArmyResearchLab/Dshell.

[169] Zalewski, M. *p0f v3*. 2014. Available at: https://lcamtuf.coredump.cx/p0f3/#.

Datasets references

[170] *2008 Nitroba University Harassment Scenario*. Available at: https://digitalcorpora.org/corpora/scenarios/nitroba-university-harassment-scenario.

[171] *DDoS Evaluation Dataset (CIC-DDoS2019)*. Available at: https://www.unb.ca/cic/datasets/ddos-2019.html.

[172] *Intrusion Detection Evaluation Dataset (CIC-IDS2017).* Available at: https://www.unb.ca/cic/datasets/ids-2017.html.

[173] *Publicly available PCAP files.* Available at: https://www.netresec.com/?page=PcapFiles.

[174] *ICS_PCAPS release modbus TCP SCADA #1.* January 2019. Available at: https://github.com/tjcruz-dei/ICS_PCAPS/releases/tag/MODBUSTCP%231.

[175] *Wireshark SampleCaptures.* 2020. Available at: https://gitlab.com/wireshark/wireshark/-/wikis/SampleCaptures.

[176] Frazão, I., Abreu, P. H., Cruz, T., Araújo, H. and Simões, P. Denial of Service Attacks: Detecting the frailties of machine learning algorithms in the Classification Process. In: Springer, ed. *13th International Conference on Critical Information Infrastructures Security (CRITIS 2018).* 2018, p. 230–235. DOI: 10.1007/978-3-030-05849-4_19. ISBN 978-3-030-05848-7.

[177] Hjelmvik, E. *Packet Injection Attacks in the Wild.* 2016. Available at: https://netresec.com/?b=163e02b.

[178] Ring, M., Wunderlich, S., Scheuring, D., Landes, D. and Hotho, A. A Survey of Network-based Intrusion Detection Data Sets. *Computers & Security.* march 2019, vol. 86, p. 147 – 167. DOI: 10.1016/j.cose.2019.06.005.

[179] Sharafaldin, I., Lashkari, A. H. and Ghorbani, A. A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In: *4th International Conference on Information Systems Security and Privacy (ICISSP).* 2018, p. 108–116. DOI: 10.5220/0006639801080116. ISBN 9789897582820.

[180] Sharafaldin, I., Lashkari, A. H., Hakak, S. and Ghorbani, A. A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: *International Carnahan Conference on Security Technology (ICCST).* October 2019. DOI: 10.1109/CCST.2019.8888419. ISBN 9781728115764.

[181] Wu, P. *tls - peter/wireshark-notes.* Available at: https://git.lekensteyn.nl/peter/wireshark-notes/tree/tls/.

[182] Wu, P. *mysql-ssl.pcapng - peter/wireshark-notes.* 2015. Available at: https://git.lekensteyn.nl/peter/wireshark-notes/commit/tls/mysql-ssl.pcapng?id=8cfd2f667e796e4c0e3bdbe117e515206346f74a.

[183] Wu, P. *pop-ssl.pcapng - peter/wireshark-notes.* 2015. Available at: https://git.lekensteyn.nl/peter/wireshark-notes/commit/tls/pop-ssl.pcapng?id=860c55ba8449a877e21480017e16cfae902b69fb.

[184] Wu, P. *smtp-ssl.pcapng - peter/wireshark-notes.* 2015. Available at: https://git.lekensteyn.nl/peter/wireshark-notes/commit/tls/smtp-ssl.pcapng?id=9615a132638741baa2cf839277128a32e4fc34f2.

Other references

[185] Dailey, M. Digital forensic tools. *SC Magazine*. 2012, vol. 23, no. 5, p. 50–51.

[186] Davidoff, S. and Ham, J. *Network Forensics: Tracking hackers through cyberspace*. Pearson Education, Inc., 2012. 545 p. ISBN 0132564718.

[187] ENISA. *Introduction to Network Forensics*. European Union Agency for Cybersecurity (ENISA), 2019.

[188] Garfinkel, S. Network Forensics: Tapping the Internet. *O'Reilly Network*. 2002.

[189] Joshi, R. C. and Pilli, E. S. Network Forensic Tools. In: Sammes, A. J., ed. *Fundamentals of Network Forensics*. Springer, 2016, chap. 4, p. 71–93. DOI: 10.1007/978-1-4471-7299-4_4. ISBN 978-1-4471-7297-0.

[190] Joshi, R. C. and Pilli, E. S. Network Forensics. In: Sammes, A. J., ed. *Fundamentals of Network Forensics*. Springer, 2016, chap. 1, p. 3–16. DOI: 10.1007/978-1-4471-7299-4_1. ISBN 978-1-4471-7297-0.

[191] Khan, S., Gani, A., Wahab, A. W. A., Shiraz, M. and Ahmad, I. Network forensics: Review, taxonomy, and open challenges. *Journal of Network and Computer Applications*. may 2016, vol. 66, p. 214–235. DOI: 10.1016/j.jnca.2016.03.005. ISSN 1084-8045.

[192] Lubis, A. and Siahaan, A. P. U. NetworkForensic Application in General Cases. *IOSR Journal of Computer Engineering (IOSR-JCE)*. 2016, vol. 18, no. 6, p. 41–44. DOI: 10.9790/0661-1806044144.

[193] Maras, M.-H. Network Forensics: An Introduction. In: *Computer Forensics: Cybercriminals, Laws, and Evidence*. Secondth ed. Jones & Bartlett Learning, 2015, chap. 12. ISBN 978-1-4496-9222-3.

[194] Meghanathan, N., Allam, S. R. and Moore, L. A. TOOLS AND TECHNIQUES FOR NETWORK FORENSICS. *International Journal of Network Security & Its Applications (IJNSA)*. april 2009, vol. 1, no. 1, p. 14–25.

[195] Nabbali, T. and Perry, M. Going for the throat: Carnivore in an Echelon World - Part I. *Computer Law and Security Report*. Elsevier Ltd. december 2003, vol. 19, no. 6, p. 456–467. DOI: 10.1016/S0267-3649(03)00603-4. ISSN 02673649/03.

[196] Palmer, G. A Framework for Digital Forensic Scrience Part of A Road Map for Digital Forensic Research. In: *Digital Forensic Research Conference (DFRWS)*. 2001, p. 15–20.

[197] Palmer, G. Digital Forensic Science in Networked Environments (Network Forensics) Part of A Road Map for Digital Forensic Research. In: *Digital Forensic Research Conference (DFRWS)*. 2001, p. 27–31.

[198] Pilli, E. S., Joshi, R. C. and Niyogi, R. Network forensic frameworks: Survey and research challenges. *Digital Investigation*. Elsevier Ltd. october 2010, vol. 7, 1-2, p. 14–27. DOI: 10.1016/j.diin.2010.02.003. ISSN 17422876.

[199] Posey, B. Monitor the data packets on your network with eEye's Iris. *Data Centers*. 2003. Available at: https://www.techrepublic.com/article/monitor-the-data-packets-on-your-network-with-eeyes-iris/.

[200] Rouse, M. *What is network forensics?* 2004. Available at: https://searchsecurity.techtarget.com/definition/network-forensics.

[201] Sharevski, F. Network Forensics: Fundamentals. In: *Mobile Network Forensics: Emerging Research and Opportunities*. IGI Global, 2018, chap. 1, p. 1–18. DOI: 10.4018/978-1-5225-5855-2. ISBN 9781522558552.

[202] Sira, R. *Network Forensics Analysis Tools: An Overview of an Emerging Technology*. Security 401. SANS Institute, 2003. 1–11 p.

Acronyms and Abbreviations

AP	Access Point
ARP	Address Resolution Protocol
ASI	Adaptive Service Intelligence
CAM	Content Addressable Memory
CDP	Cisco Discovery Protocol
CWND	Congestion Window
DAST	Dynamic Application Security Testing
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoH	DNS over HTTPS
DoS	Denial of Service
ENISA	European Network and Information Security Agency
FBI	Federal Bureau of Investigation
GUI	Graphical User Interface
HIDS	Host-based IDS
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection and Prevention Systems
IDS	Intrusion Detection System
IE	Internet Explorer
IJNSA	International Journal of Network Security & Its Applications
IOSR	International Organization of Scientific Research
IOSR-JCE	International Organization of Scientific Research Journal of Computer Engineering

IoT	Internet-of-Things
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPS	Intrusion Prevention Systems
ISP	Internet Service Provider
MOTS	Man-on-the-Side
NAT	Network Address Translation
NBT	NetBIOS over TCP/IP
NFATs	Network Forensic Analysis Tools
NFR	Network Flight Recorder
NFT	Network Forensics Technique
NIDS	Network-based IDS
NSM	Network Security and Monitoring
NSSM	Non-Sucking Service Manager
NTA	NetFlow Traffic Analyzer
OFT	Oscar File Transfer
OSVDB	Open Source Vulnerability Database
PADS	Passive Asset Detection System
PCAP	Packet Capture
QoS	Quality of Service
RTT	Round-Trip Time
SDN	Software-Defined Network
SDR	Software Defined Radio
SIEM	Security Information and Event Management
SiLK	System for Internet-Level Knowledge
SIP	Session Initiation Protocol
SNI	Server Name Indication
STP	Spanning Tree Protocol

TTL	Time To Live
UTC	Coordinated Universal Time
VPN	Virtual Private Network
WAP	Wireless Access Point
WIDS	Wireless Intrusion Detection System
WLAN	Wireless Local Area Network
WVS	Web Vulnerability Scanner
YAF	Yet Another Flowmeter

Appendices

Appendix A

Details of the PCAP files from "DoS Attacks against SCADA-based systems" dataset

A.1 Nominal state

captures1\clean\eth2dump-clean-0,5h_1.pcap
Number of packets: 35 430 *Timeline:* 2018-08-23 17:40:48.376131 – 2018-08-23 18:10:47.968813
captures1\clean\eth2dump-clean-1h_1.pcap
Number of packets: 72 150 *Timeline:* 2018-09-08 20:54:30.055515 UTC – 2018-09-08 21:54:29.869752 UTC
captures1\clean\eth2dump-clean-6h_1.pcap
Number of packets: 427 842 *Timeline:* 2018-09-08 18:14:04.978565 UTC – 2018-09-09 00:14:03.946853 UTC

Table A.1: Annotation of the nominal state PCAPs

A.2 ARP-based, Man-in-the-Middle attack

captures1\mitm\eth2dump-mitm-change-1m-0,5h_1.pcap
Number of packets: 35 477 *Timeline:* 2018-08-23 21:14:15.166155 UTC – 2018-08-23 21:44:13.891139 UTC *Attack:* MITM change (21:19:15 – 21:20:15)
captures1\mitm\eth2dump-mitm-change-1m-1h_1.pcap
Number of packets: 70 863 *Timeline:* 2018-08-23 22:15:30.313736 UTC – 2018-08-23 23:15:30.174669 UTC *Attack:* MITM change (22:20:30 – 22:21:30)
captures1\mitm\eth2dump-mitm-change-1m-6h_1.pcap
Number of packets: 422 332 *Timeline:* 2018-08-24 06:20:31.023559 UTC – 2018-08-24 12:20:31.197900 UTC *Attack:* MITM change (06:25:31 – 06:26:31)
captures1\mitm\eth2dump-mitm-change-5m-0,5h_1.pcap

Number of packets: 35 430
Timeline: 2018-08-23 18:57:01.789547 UTC – 2018-08-23 19:27:00.923039 UTC
Attack: MITM change (19:02:02 – 19:07:02)

captures1\mitm\eth2dump-mitm-change-5m-1h_1.pcap
Number of packets: 69 440
Timeline: 2018-08-23 23:15:55.127986 UTC – 2018-08-24 00:15:54.690784
Attack: MITM change (23:20:55 – 23:25:55)

captures1\mitm\eth2dump-mitm-change-5m-6h_1.pcap
Number of packets: 419 377
Timeline: 2018-08-24 12:21:01.090449 UTC – 2018-08-24 18:20:59.990328 UTC
Attack: MITM change (12:26:01 – 12:31:01)

captures1\mitm\eth2dump-mitm-change-15m-0,5h_1.pcap
Number of packets: 35 476
Timeline: 2018-08-23 21:44:49.728673 UTC – 2018-08-23 22:14:48.927035 UTC
Attack: MITM change (21:49:49 – 22:04:49)

captures1\mitm\eth2dump-mitm-change-15m-1h_1.pcap
Number of packets: 70 985
Timeline: 2018-08-24 00:16:22.370615 UTC – 2018-08-24 01:16:21.693555 UTC
Attack: MITM change (00:21:22 – 00:36:22)

captures1\mitm\eth2dump-mitm-change-15m-6h_1.pcap
Number of packets: 418 103
Timeline: 2018-08-24 18:21:26.100597 UTC – 2018-08-25 00:21:24.818414 UTC
Attack: MITM change (18:26:26 – 18:41:26)

captures1\mitm\eth2dump-mitm-change-30m-1h_1.pcap
Number of packets: 71 037
Timeline: 2018-08-24 01:16:47.549512 UTC – 2018-08-24 02:16:46.704859 UTC
Attack: MITM change (01:21:47 – 01:51:47)

captures1\mitm\eth2dump-mitm-change-30m-6h_1.pcap
Number of packets: 421 043
Timeline: 2018-08-25 00:21:49.663518 UTC – 2018-08-25 06:21:48.941567 UTC
Attack: MITM change (00:26:50 – 00:56:50)

captures1\mitm\eth2dump-mitm-read-1m-0,5h_1.pcap
Number of packets: 35 917
Timeline: 2018-09-02 20:34:11.055967 UTC – 2018-09-02 21:04:09.952139 UTC
Attack: MITM read (20:39:11 – 20:40:11)

captures1\mitm\eth2dump-mitm-read-1m-1h_1.pcap
Number of packets: 71 265
Timeline: 2018-09-02 22:05:39.590991 UTC – 2018-09-02 23:05:39.076235 UTC
Attack: MITM read (22:10:39 – 22:11:39)

captures1\mitm\eth2dump-mitm-read-1m-6h_1.pcap
Number of packets: 427 005
Timeline: 2018-09-03 02:07:39.103114 UTC – 2018-09-03 08:07:37.830640 UTC
Attack: MITM read (02:12:39 – 02:13:39)

captures1\mitm\eth2dump-mitm-read-5m-0,5h_1.pcap
Number of packets: 35 634
Timeline: 2018-09-02 21:04:36.994835 UTC – 2018-09-02 21:34:35.955534 UTC
Attack: MITM read (21:09:36 – 21:14:36)

captures1\mitm\eth2dump-mitm-read-5m-1h_1.pcap
Number of packets: 71 359 *Timeline:* 2018-09-02 23:06:04.544099 UTC – 2018-09-03 00:06:03.896178 UTC *Attack:* MITM read (23:11:04 – 23:16:04)
captures1\mitm\eth2dump-mitm-read-5m-6h_1.pcap
Number of packets: 422 962 *Timeline:* 2018-09-03 08:08:06.796976 UTC – 2018-09-03 14:08:06.180506 UTC *Attack:* MITM read (08:13:06 – 08:18:06)
captures1\mitm\eth2dump-mitm-read-15m-0,5h_1.pcap
Number of packets: 35 641 *Timeline:* 2018-09-02 21:34:57.171191 UTC – 2018-09-02 22:04:57.128462 UTC *Attack:* MITM read (21:39:57 – 21:54:57)
captures1\mitm\eth2dump-mitm-read-15m-1h_1.pcap
Number of packets: 69 216 *Timeline:* 2018-09-03 00:06:31.920822 UTC – 2018-09-03 01:06:30.549189 UTC *Attack:* MITM read (00:11:31 – 00:26:31)
captures1\mitm\eth2dump-mitm-read-15m-6h_1.pcap
Number of packets: 425 224 *Timeline:* 2018-09-03 14:08:27.916169 UTC – 2018-09-03 20:08:26.995065 UTC *Attack:* MITM read (14:13:27 – 14:28:27)
captures1\mitm\eth2dump-mitm-read-30m-1h_1.pcap
Number of packets: 70 687 *Timeline:* 2018-09-03 01:06:56.020609 UTC – 2018-09-03 02:06:55.211520 UTC *Attack:* MITM read (01:11:56 – 01:41:56)
captures1\mitm\eth2dump-mitm-read-30m-6h_1.pcap
Number of packets: 426 785 *Timeline:* 2018-09-03 20:08:55.698605 UTC – 2018-09-04 02:08:55.041070 UTC *Attack:* MITM read (20:13:55 – 20:43:55)

Table A.2: Annotation of the MITM PCAPs

A.3 Modbus query flooding

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-0,5h_1.pcap
Number of packets: 59 213 *Timeline:* 2018-05-22 11:02:29.885761 UTC – 2018-05-22 11:32:28.736724 UTC *Attack:* Modbus query flood (11:07:29 – 11:08:29)
captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-1h_1.pcap
Number of packets: 106 602 *Timeline:* 2018-05-23 13:26:45.472829 UTC – 2018-05-23 14:26:45.085052 UTC *Attack:* Modbus query flood (13:31:45 – 13:32:45)
captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-6h_1.pcap
Number of packets: 442 136 *Timeline:* 2018-05-25 18:39:01.630605 UTC – 2018-05-26 00:39:00.757195 UTC *Attack:* Modbus query flood (18:44:01 – 18:45:01)
captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-0,5h_1.pcap
Number of packets: 210 955

Timeline: 2018-05-22 10:31:45.904531 UTC – 2018-05-22 11:01:43.952304 UTC
Attack: Modbus query flood (10:36:45 – 10:41:45)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-1h_1.pcap
Number of packets: 264 284
Timeline: 2018-05-23 12:26:42.792994 UTC – 2018-05-23 13:26:42.169298 UTC
Attack: Modbus query flood (12:31:42 – 12:36:42)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-6h_1.pcap
Number of packets: 615 273
Timeline: 2018-05-26 00:39:34.756860 UTC – 2018-05-26 06:39:33.804054 UTC
Attack: Modbus query flood (00:44:34 – 00:49:34)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-0,5h_1.pcap
Number of packets: 338 375
Timeline: 2018-05-22 10:00:59.923334 UTC – 2018-05-22 10:30:57.905008 UTC
Attack: Modbus query flood (10:05:59 – 10:20:59)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-1h_1.pcap
Number of packets: 691 761
Timeline: 2018-05-23 11:26:39.256021 UTC – 2018-05-23 12:26:38.632629 UTC
Attack: Modbus query flood (11:31:39 – 11:46:39)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-6h_1.pcap
Number of packets: 406 697
Timeline: 2018-05-26 06:40:04.771587 UTC – 2018-05-26 12:40:03.814120 UTC
Attack: Modbus query flood (06:45:04 – 07:00:04)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-1h_1.pcap
Number of packets: 407 414
Timeline: 2018-05-30 03:38:12.370803 UTC – 2018-05-30 04:38:11.892707 UTC
Attack: Modbus query flood (03:43:12 – 04:13:12)

captures1\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-6h_1.pcap
Number of packets: 681 591
Timeline: 2018-05-26 12:40:35.738566 UTC – 2018-05-26 18:40:34.088488 UTC
Attack: Modbus query flood (12:45:35 – 13:15:35)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding1m-0,5h_1.pcap
Number of packets: 73 563
Timeline: 2018-05-22 13:21:03.778587 UTC – 2018-05-22 13:51:02.812021 UTC
Attack: Modbus query flood (13:26:03 – 13:27:03)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding1m-1h_1.pcap
Number of packets: 106 913
Timeline: 2018-05-22 15:55:43.077098 UTC – 2018-05-22 16:55:43.055859 UTC
Attack: Modbus query flood (16:00:43 – 16:01:43)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding1m-6h_1.pcap
Number of packets: 253 185
Timeline: 2018-05-26 18:41:16.088064 UTC – 2018-05-27 00:41:14.625176 UTC
Attack: Modbus query flood (18:46:16 – 18:47:16)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding5m-0,5h_1.pcap
Number of packets: 234 549
Timeline: 2018-05-22 12:50:36.509002 UTC – 2018-05-22 13:20:36.195308 UTC
Attack: Modbus query flood (12:55:36 – 13:00:36)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding5m-1h_1.pcap

Number of packets: 271 113
Timeline: 2018-05-22 14:55:03.442279 UTC – 2018-05-22 15:55:03.070856 UTC
Attack: Modbus query flood (15:00:03 – 15:05:03)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding5m-6h_1.pcap
Number of packets: 416 874
Timeline: 2018-05-27 00:41:55.866591 UTC – 2018-05-27 06:41:54.093822 UTC
Attack: Modbus query flood (00:46:55 – 00:51:55)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding15m-0,5h_1.pcap
Number of packets: 656 786
Timeline: 2018-05-22 12:20:06.996261 UTC – 2018-05-22 12:50:05.695480 UTC
Attack: Modbus query flood (12:25:06 – 12:40:06)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding15m-1h_1.pcap
Number of packets: 686 204
Timeline: 2018-05-22 13:54:18.448701 UTC – 2018-05-22 14:54:17.832912 UTC
Attack: Modbus query flood (13:59:18 – 14:14:18)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding15m-6h_1.pcap
Number of packets: 817 668
Timeline: 2018-05-27 06:42:21.400371 UTC – 2018-05-27 12:42:19.664486 UTC
Attack: Modbus query flood (06:47:21 – 07:02:21)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding30m-1h_1.pcap
Number of packets: 520 524
Timeline: 2018-05-29 21:32:08.422394 UTC – 2018-05-29 22:32:07.751491 UTC
Attack: Modbus query flood (21:37:08 – 22:07:08)

captures1\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding30m-6h_1.pcap
Number of packets: 143 5563
Timeline: 2018-05-27 12:42:45.705220 UTC – 2018-05-27 18:42:44.770975 UTC
Attack: Modbus query flood (12:47:45 – 13:17:45)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-0,5h_1.pcap
Number of packets: 56 753
Timeline: 2018-07-22 22:44:41.608057 UTC – 2018-07-22 23:14:40.219209 UTC
Attack: Modbus query flood (22:49:41 – 22:50:41)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-1h_1.pcap
Number of packets: 78 475
Timeline: 2018-07-23 00:17:00.184711 UTC – 2018-07-23 01:16:58.514300 UTC
Attack: Modbus query flood (00:22:00 – 00:23:00)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-6h_1.pcap
Number of packets: 253 273
Timeline: 2018-08-13 17:50:23.789438 UTC – 2018-08-13 23:50:22.905925 UTC
Attack: Modbus query flood (17:55:23 – 17:56:23)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-0,5h_1.pcap
Number of packets: 220 691
Timeline: 2018-07-22 23:15:08.218987 UTC – 2018-07-22 23:45:07.297839 UTC
Attack: Modbus query flood (23:20:08 – 23:25:08)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-1h_1.pcap
Number of packets: 246 926
Timeline: 2018-07-23 01:17:26.012512 UTC – 2018-07-23 02:17:24.732133 UTC
Attack: Modbus query flood (01:22:26 – 01:27:26)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-6h_1.pcap
Number of packets: 432 438
Timeline: 2018-08-13 23:51:10.492344 UTC – 2018-08-14 05:51:08.216217 UTC
Attack: Modbus query flood (23:56:10 – 00:01:10)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-0,5h_1.pcap
Number of packets: 640 549
Timeline: 2018-07-22 23:45:50.201786 UTC – 2018-07-23 00:15:48.557453 UTC
Attack: Modbus query flood (23:50:50 – 00:05:50)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-1h_1.pcap
Number of packets: 656 017
Timeline: 2018-07-23 02:17:51.823201 UTC – 2018-07-23 03:17:50.083843 UTC
Attack: Modbus query flood (02:22:51 – 02:37:51)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-6h_1.pcap
Number of packets: 824 083
Timeline: 2018-08-14 05:51:50.215279 UTC – 2018-08-14 11:51:48.210847 UTC
Attack: Modbus query flood (05:56:50 – 06:11:50)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-1h_1.pcap
Number of packets: 1 244 893
Timeline: 2018-07-23 03:18:20.083731 UTC – 2018-07-23 04:18:19.257150 UTC
Attack: Modbus query flood (03:23:20 – 03:53:20)

captures2\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-6h_1.pcap
Number of packets: 283 616
Timeline: 2018-08-14 11:52:18.639209 UTC – 2018-08-14 17:52:17.836726 UTC
Attack: Modbus query flood (11:57:18 – 12:27:18)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-0,5h_1.pcap
Number of packets: 77 023
Timeline: 2018-08-10 16:06:06.706802 UTC – 2018-08-10 16:36:06.147050 UTC
Attack: Modbus query flood (16:11:06 – 16:12:06)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-1h_1.pcap
Number of packets: 76 008
Timeline: 2018-08-10 17:37:32.371566 UTC – 2018-08-10 18:37:31.817184 UTC
Attack: Modbus query flood (17:42:32 – 17:43:32)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-6h_1.pcap
Number of packets: 256 682
Timeline: 2018-07-25 23:48:03.669771 UTC – 2018-07-26 05:48:02.844806 UTC
Attack: Modbus query flood (23:53:03 – 23:54:03)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-0,5h_1.pcap
Number of packets: 242 664
Timeline: 2018-08-10 16:36:31.153617 UTC – 2018-08-10 17:06:30.143110 UTC
Attack: Modbus query flood (16:41:31 – 16:46:31)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-1h_1.pcap
Number of packets: 246 087
Timeline: 2018-08-10 18:37:57.886821 UTC – 2018-08-10 19:37:56.307179 UTC
Attack: Modbus query flood (18:42:57 – 18:47:57)

captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding5m-6h_1.pcap
Number of packets: 429 670
Timeline: 2018-07-26 05:48:08.397619 UTC – 2018-07-26 11:48:06.035258 UTC

Attack: Modbus query flood (05:53:08 – 05:58:08)
captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-0,5h_1.pcap
Number of packets: 653 059 *Timeline:* 2018-08-10 17:06:50.441763 UTC – 2018-08-10 17:36:49.226614 UTC *Attack:* Modbus query flood (17:11:50 – 17:26:50)
captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-1h_1.pcap
Number of packets: 650 875 *Timeline:* 2018-08-10 19:38:23.328304 UTC – 2018-08-10 20:38:22.352708 UTC *Attack:* Modbus query flood (19:43:23 – 19:58:23)
captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding15m-6h_1.pcap
Number of packets: 663 287 *Timeline:* 2018-07-26 11:48:11.275041 UTC – 2018-07-26 17:48:09.599156 UTC *Attack:* Modbus query flood (11:53:11 – 12:08:11)
captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-1h_1.pcap
Number of packets: 1 265 588 *Timeline:* 2018-08-10 20:38:48.207131 UTC – 2018-08-10 21:38:46.533915 UTC *Attack:* Modbus query flood (20:43:48 – 21:13:48)
captures3\modbusQueryFlooding\eth2dump-modbusQueryFlooding30m-6h_1.pcap
Number of packets: 665 322 *Timeline:* 2018-07-26 17:48:13.715360 UTC – 2018-07-26 23:48:12.708644 UTC *Attack:* Modbus query flood (17:53:13 – 18:23:13)

Table A.3: Annotation of the Modbus query flooding PCAPs

A.4 ICMP flooding

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS1m-0,5h_1.pcap
Number of packets: 29 692 *Timeline:* 2018-05-21 10:29:42.556747 UTC – 2018-05-21 10:59:40.914366 UTC *Attack:* ICMP flood (10:34:42 – 10:35:42)
captures1\pingFloodDDoS\eth2dump-pingFloodDDoS1m-1h_1.pcap
Number of packets: 47 387 *Timeline:* 2018-05-21 14:01:24.659729 UTC – 2018-05-21 15:01:22.733558 UTC *Attack:* ICMP flood (14:06:24 – 14:07:24)
captures1\pingFloodDDoS\eth2dump-pingFloodDDoS1m-6h_1.pcap
Number of packets: 414 965 *Timeline:* 2018-05-28 15:38:47.982400 UTC – 2018-05-28 21:38:46.789641 UTC *Attack:* ICMP flood (15:43:47 – 15:44:47)
captures1\pingFloodDDoS\eth2dump-pingFloodDDoS5m-0,5h_1.pcap
Number of packets: 77 726 *Timeline:* 2018-05-21 09:54:56.811018 UTC – 2018-05-21 10:24:55.692791 UTC *Attack:* ICMP flood (09:59:56 – 10:04:56)
captures1\pingFloodDDoS\eth2dump-pingFloodDDoS5m-1h_1.pcap
Number of packets: 95 379 *Timeline:* 2018-05-21 13:00:18.544828 UTC – 2018-05-21 14:00:16.660391 UTC *Attack:* ICMP flood (13:05:18 – 13:10:18)
captures1\pingFloodDDoS\eth2dump-pingFloodDDoS5m-6h_1.pcap

Number of packets: 458 694
Timeline: 2018-05-28 21:39:18.576864 UTC – 2018-05-29 03:39:17.648818 UTC
Attack: ICMP flood (21:44:18 – 21:49:18)

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS15m-0,5h_1.pcap
Number of packets: 197 778
Timeline: 2018-05-21 11:12:37.348723 UTC – 2018-05-21 11:42:36.743681 UTC
Attack: ICMP flood (11:17:37 – 11:32:37)

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS15m-1h_1.pcap
Number of packets: 215 393
Timeline: 2018-05-21 11:59:47.918379 UTC – 2018-05-21 12:59:45.235818 UTC
Attack: ICMP flood (12:04:47 – 12:19:47)

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS15m-6h_1.pcap
Number of packets: 575 901
Timeline: 2018-05-29 03:39:47.264889 UTC – 2018-05-29 09:39:46.802228 UTC
Attack: ICMP flood (03:44:47 – 03:59:47)

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS30m-1h_1.pcap
Number of packets: 424 391
Timeline: 2018-05-29 18:28:49.890451 UTC – 2018-05-29 19:28:49.105863 UTC
Attack: ICMP flood (18:33:49 – 19:03:49)

captures1\pingFloodDDoS\eth2dump-pingFloodDDoS30m-6h_1.pcap
Number of packets: 755 159
Timeline: 2018-05-29 09:40:20.358752 UTC – 2018-05-29 15:40:19.836674 UTC
Attack: ICMP flood (09:45:20 – 10:15:20)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS1m-0,5h_1.pcap
Number of packets: 47 172
Timeline: 2018-08-08 23:26:58.620063 UTC – 2018-08-08 23:56:58.240917 UTC
Attack: ICMP flood (23:31:58 – 23:32:58)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS1m-1h_1.pcap
Number of packets: 47 964
Timeline: 2018-07-10 13:29:01.889637 UTC – 2018-07-10 14:29:00.153496 UTC
Attack: ICMP flood (13:34:01 – 13:35:01)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS1m-6h_1.pcap
Number of packets: 431 130
Timeline: 2018-08-04 17:40:29.103039 UTC – 2018-08-04 23:40:28.978928 UTC
Attack: ICMP flood (17:45:29 – 17:46:29)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS5m-0,5h_1.pcap
Number of packets: 94 152
Timeline: 2018-08-08 23:57:01.140910 UTC – 2018-08-09 00:27:00.954303 UTC
Attack: ICMP flood (00:02:01 – 00:07:01)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS5m-1h_1.pcap
Number of packets: 95 841
Timeline: 2018-07-10 14:29:07.169358 UTC – 2018-07-10 15:29:05.492768 UTC
Attack: ICMP flood (14:34:07 – 14:39:07)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS5m-6h_1.pcap
Number of packets: 478 954
Timeline: 2018-08-04 23:40:59.030645 UTC – 2018-08-05 05:40:57.798724 UTC
Attack: ICMP flood (23:45:59 – 23:50:59)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS15m-0,5h_1.pcap
Number of packets: 212 192
Timeline: 2018-08-09 00:27:03.442593 UTC – 2018-08-09 00:57:02.800427 UTC
Attack: ICMP flood (00:32:03 – 00:47:03)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS15m-1h_1.pcap
Number of packets: 215 376
Timeline: 2018-07-10 15:29:13.957275 UTC – 2018-07-10 16:29:10.772865 UTC
Attack: ICMP flood (15:34:13 – 15:49:13)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS15m-6h_1.pcap
Number of packets: 600 269
Timeline: 2018-08-05 05:41:31.347164 UTC – 2018-08-05 11:41:30.798740 UTC
Attack: ICMP flood (05:46:31 – 06:01:31)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS30m-1h_1.pcap
Number of packets: 394 842
Timeline: 2018-07-10 16:29:17.777918 UTC – 2018-07-10 17:29:15.994621 UTC
Attack: ICMP flood (16:34:17 – 17:04:17)

captures2\pingFloodDDoS\eth2dump-pingFloodDDoS30m-6h_1.pcap
Number of packets: 778 206
Timeline: 2018-08-05 11:42:04.595393 UTC – 2018-08-05 17:42:03.772803 UTC
Attack: ICMP flood (11:47:04 – 12:17:04)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS1m-0,5h_1.pcap
Number of packets: 47 371
Timeline: 2018-08-08 16:36:26.379086 UTC – 2018-08-08 17:06:25.783761 UTC
Attack: ICMP flood (16:41:26 – 16:42:26)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS1m-1h_1.pcap
Number of packets: 47 996
Timeline: 2018-07-17 12:43:55.661432 UTC – 2018-07-17 13:43:53.833149 UTC
Attack: ICMP flood (12:48:55 – 12:49:55)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS1m-6h_1.pcap
Number of packets: 434 054
Timeline: 2018-08-06 17:47:06.861936 UTC – 2018-08-06 23:47:05.895651 UTC
Attack: ICMP flood (17:52:06 –17:53:06)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS5m-0,5h_1.pcap
Number of packets: 94 300
Timeline: 2018-08-08 17:06:45.892610 UTC – 2018-08-08 17:36:45.181641 UTC
Attack: ICMP flood (17:11:45 – 17:16:45)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS5m-1h_1.pcap
Number of packets: 95 809
Timeline: 2018-07-16 17:44:38.863054 UTC – 2018-07-16 18:44:37.361466 UTC
Attack: ICMP flood (17:49:38 – 17:54:38)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS5m-6h_1.pcap
Number of packets: 479 046
Timeline: 2018-08-06 23:47:49.517426 UTC – 2018-08-07 05:47:49.087037 UTC
Attack: ICMP flood (23:52:49 – 23:57:49)

captures3\pingFloodDDoS\eth2dump-pingFloodDDoS15m-0,5h_1.pcap
Number of packets: 212 241
Timeline: 2018-08-08 17:37:05.555487 UTC – 2018-08-08 18:07:05.118361 UTC

Attack: ICMP flood (17:42:05 – 17:57:05)
captures3\pingFloodDDoS\eth2dump-pingFloodDDoS15m-1h_1.pcap
Number of packets: 215 363 *Timeline:* 2018-07-16 18:44:42.925577 UTC – 2018-07-16 19:44:40.710443 UTC *Attack:* ICMP flood (18:49:42 – 19:04:42)
captures3\pingFloodDDoS\eth2dump-pingFloodDDoS15m-6h_1.pcap
Number of packets: 601 705 *Timeline:* 2018-08-07 05:48:18.501120 UTC – 2018-08-07 11:48:17.887162 UTC *Attack:* ICMP flood (05:53:18 – 06:08:18)
captures3\pingFloodDDoS\eth2dump-pingFloodDDoS30m-1h_1.pcap
Number of packets: 394 912 *Timeline:* 2018-07-16 19:44:46.753527 UTC – 2018-07-16 20:44:45.579568 UTC *Attack:* ICMP flood (19:49:46 – 20:19:46)
captures3\pingFloodDDoS\eth2dump-pingFloodDDoS30m-6h_1.pcap
Number of packets: 781 035 *Timeline:* 2018-08-07 11:48:58.161399 UTC – 2018-08-07 17:48:58.233502 UTC *Attack:* ICMP flood (11:53:58 – 12:23:58)

Table A.4: Annotation of the ICMP flooding PCAPs

A.5 TCP SYN flooding

captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-0,5h_1.pcap
Number of packets: 45 271 *Timeline:* 2018-05-21 19:53:09.262750 UTC – 2018-05-21 20:23:08.888401 UTC *Attack:* TCP SYN flood (19:58:09 – 19:59:09)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-1h_1.pcap
Number of packets: 46 637 *Timeline:* 2018-05-21 17:34:52.530128 UTC – 2018-05-21 18:34:51.928684 UTC *Attack:* TCP SYN flood (17:39:52 – 17:40:52)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-6h_1.pcap
Number of packets: 406 880 *Timeline:* 2018-05-25 12:12:26.190542 UTC – 2018-05-25 18:12:25.882428 UTC *Attack:* TCP SYN flood (12:17:26 – 12:18:26)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-0,5h_1.pcap
Number of packets: 81 218 *Timeline:* 2018-05-21 19:22:55.739579 UTC – 2018-05-21 19:52:54.021791 UTC *Attack:* TCP SYN flood (19:27:55 – 19:32:55)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-1h_1.pcap
Number of packets: 92 588 *Timeline:* 2018-05-21 16:34:26.566653 UTC – 2018-05-21 17:34:24.530387 UTC *Attack:* TCP SYN flood (16:39:26 – 16:44:26)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-6h_1.pcap
Number of packets: 406 655 *Timeline:* 2018-05-25 06:11:49.771078 UTC – 2018-05-25 12:11:49.117337 UTC *Attack:* TCP SYN flood (06:16:49 – 06:21:49)
captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-0,5h_1.pcap

Number of packets: 190 816
Timeline: 2018-05-21 18:38:22.491841 UTC – 2018-05-21 19:08:22.032780 UTC
Attack: TCP SYN flood (18:43:22 – 18:58:22)

captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-1h_1.pcap
Number of packets: 208 062
Timeline: 2018-05-21 15:32:57.012079 UTC – 2018-05-21 16:32:55.440337 UTC
Attack: TCP SYN flood (15:37:57 – 15:52:57)

captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-6h_1.pcap
Number of packets: 417 519
Timeline: 2018-05-25 00:11:10.904903 UTC – 2018-05-25 06:11:09.752817 UTC
Attack: TCP SYN flood (00:16:10 – 00:31:10)

captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-1h_1.pcap
Number of packets: 91 474
Timeline: 2018-05-29 19:29:56.309589 UTC – 2018-05-29 20:29:55.904111 UTC
Attack: TCP SYN flood (19:34:56 – 20:04:56)

captures1\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-6h_1.pcap
Number of packets: 1 518 310
Timeline: 2018-05-29 21:34:45.894084 UTC – 2018-05-30 06:34:45.150888 UTC
Attack: TCP SYN flood (21:39:45 – 22:09:45)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-0,5h_1.pcap
Number of packets: 19 034
Timeline: 2018-07-22 16:40:18.434460 UTC – 2018-07-22 17:10:17.450384 UTC
Attack: TCP SYN flood (16:45:18 – 16:46:18)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-1h_1.pcap
Number of packets: 36 647
Timeline: 2018-07-22 18:41:21.197530 UTC – 2018-07-22 19:41:18.782451 UTC
Attack: TCP SYN flood (18:46:21 – 18:47:21)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-6h_1.pcap
Number of packets: 217 201
Timeline: 2018-07-23 04:19:44.049726 UTC – 2018-07-23 10:19:41.997739 UTC
Attack: TCP SYN flood (04:24:44 – 04:25:44)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-0,5h_1.pcap
Number of packets: 22 616
Timeline: 2018-07-22 17:10:25.894367 UTC – 2018-07-22 17:40:24.401638 UTC
Attack: TCP SYN flood (17:15:25 – 17:20:25)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-1h_1.pcap
Number of packets: 40 485
Timeline: 2018-07-22 19:42:00.597644 UTC – 2018-07-22 20:41:59.300769 UTC
Attack: TCP SYN flood (19:47:00 – 19:52:00)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-6h_1.pcap
Number of packets: 220 600
Timeline: 2018-07-23 10:20:08.047671 UTC – 2018-07-23 16:20:06.579062 UTC
Attack: TCP SYN flood (10:25:08 – 10:30:08)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-0,5h_1.pcap
Number of packets: 32 185
Timeline: 2018-07-22 17:40:48.080185 UTC – 2018-07-22 18:10:46.946412 UTC
Attack: TCP SYN flood (17:45:48 – 18:00:48)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-1h_1.pcap
Number of packets: 48 846
Timeline: 2018-07-22 20:42:22.492409 UTC – 2018-07-22 21:42:20.886283 UTC
Attack: TCP SYN flood (20:47:22 – 21:02:22)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-6h_1.pcap
Number of packets: 228 273
Timeline: 2018-07-23 16:20:33.649676 UTC – 2018-07-23 22:20:32.693610 UTC
Attack: TCP SYN flood (16:25:33 – 16:40:33)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-1h_1.pcap
Number of packets: 62 900
Timeline: 2018-07-22 21:42:52.269709 UTC – 2018-07-22 22:42:51.466220 UTC
Attack: TCP SYN flood (21:47:52 – 22:17:52)

captures2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-6h_1.pcap
Number of packets: 246 029
Timeline: 2018-07-23 22:21:07.208234 UTC – 2018-07-24 04:21:05.438953 UTC
Attack: TCP SYN flood (22:26:07 – 22:56:07)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-0,5h_1.pcap
Number of packets: 36 352
Timeline: 2018-08-01 16:14:10.364468 UTC – 2018-08-01 16:44:10.011718 UTC
Attack: TCP SYN flood (16:19:10 – 16:20:10)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-1h_1.pcap
Number of packets: 71 870
Timeline: 2018-08-01 17:44:22.298784 UTC – 2018-08-01 18:44:22.207160 UTC
Attack: TCP SYN flood (17:49:22 – 17:50:22)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-6h_1.pcap
Number of packets: 420 139
Timeline: 2018-08-05 17:43:09.120147 UTC – 2018-08-05 23:43:09.246958 UTC
Attack: TCP SYN flood (17:48:09 – 17:49:09)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-0,5h_1.pcap
Number of packets: 38 305
Timeline: 2018-08-01 16:44:13.950838 UTC – 2018-08-01 17:14:12.655620 UTC
Attack: TCP SYN flood (16:49:13 – 16:54:13)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-1h_1.pcap
Number of packets: 73 698
Timeline: 2018-08-01 18:44:27.149105 UTC – 2018-08-01 19:44:27.028327 UTC
Attack: TCP SYN flood (18:49:27 – 18:54:27)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS5m-6h_1.pcap
Number of packets: 424 156
Timeline: 2018-08-05 23:43:37.534300 UTC – 2018-08-06 05:43:37.177630 UTC
Attack: TCP SYN flood (23:48:37 – 23:53:37)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-0,5h_1.pcap
Number of packets: 49 141
Timeline: 2018-08-01 17:14:17.235707 UTC – 2018-08-01 17:44:17.238788 UTC
Attack: TCP SYN flood (17:19:17 – 17:34:17)

captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-1h_1.pcap
Number of packets: 81 504
Timeline: 2018-08-01 19:44:32.834262 UTC – 2018-08-01 20:44:31.606896 UTC

Attack: TCP SYN flood (19:49:32 – 20:04:32)
captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS15m-6h_1.pcap
Number of packets: 430 779
Timeline: 2018-08-06 05:44:09.846494 UTC – 2018-08-06 11:44:08.652592 UTC
Attack: TCP SYN flood (05:49:09 – 06:04:09)
captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-1h_1.pcap
Number of packets: 96 992
Timeline: 2018-08-01 21:44:42.783597 UTC – 2018-08-01 22:44:42.036915 UTC
Attack: TCP SYN flood (21:49:42 – 22:19:42)
captures3\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS30m-6h_1.pcap
Number of packets: 446 643
Timeline: 2018-08-06 11:44:40.698041 UTC – 2018-08-06 17:44:39.865941 UTC
Attack: TCP SYN flood (11:49:40 –İ2:19:40)

Table A.5: Annotation of the TCP SYN flooding PCAPs

Appendix B

Results from Nikto tool

IP	Errors	Findings	OSVDB	Start time (GMT - 4)	Server
147.229.13.40	0	8	0	2021-04-18 09:37:58	Microsoft-IIS/7.5
147.229.13.41	0	11	0 3092 3233 3268	2021-04-18 09:38:56	Apache/2.4.29 (Ubuntu)
147.229.13.48	0	4	0	2021-04-18 09:40:04	lighttpd/1.4.55
147.229.13.55	0	274	0 3092 3233 3268	2021-04-18 09:45:47	Apache/2.4.25 (Debian)
147.229.13.76	0	5	0 3233	2021-04-18 09:48:11	Apache/2.4.29 (Ubuntu)
147.229.13.88	0	5	0	2021-04-18 09:49:39	Apache/2.4.41 (Ubuntu)
147.229.13.92	0	3	0	2021-04-18 09:51:30	nginx
147.229.13.93	0	3	0	2021-04-18 09:53:46	nginx/1.18.0
147.229.13.95	0	11	0 3092 3233 3268	2021-04-18 09:54:19	Apache/2.4.29 (Ubuntu)
147.229.13.97	0	6	0 877 3233 3268	2021-04-18 09:55:26	Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/8.0.3
147.229.13.108	0	11	0 3092 3233 3268	2021-04-18 10:25:49	Apache/2.4.29 (Ubuntu)

147.229.13.123	0	11	0 3092 3233 3268	2021-04-18 10:30:49	Apache/2.4.29 (Ubuntu)
147.229.13.124	0	11	0 3092 3233 3268	2021-04-18 10:32:03	Apache/2.4.29 (Ubuntu)
147.229.13.127	0	10	0 3092 3233	2021-04-18 10:32:56	Apache/2.4.18 (Ubuntu)
147.229.13.140	0	278	0 3092 3233 3268	2021-04-18 10:33:53	Apache/2.4.29 (Ubuntu)
147.229.13.166	0	11	0 3092 3233 3268	2021-04-18 10:34:15	Apache/2.4.29 (Ubuntu)
147.229.13.167	0	11	0 3092 3233 3268	2021-04-18 10:35:11	Apache/2.4.29 (Ubuntu)
147.229.13.174	0	4	0	2021-04-18 10:36:23	Apache/2.4.29 (Ubuntu)
147.229.13.183	0	1	0	2021-04-18 10:36:45	PRTG
147.229.13.185	0	3	0	2021-04-18 10:43:03	nginx/1.18.0
147.229.13.188	0	6	0 3092	2021-04-18 10:43:22	Werkzeug/1.0.1 Python/3.8.5
147.229.13.203	0	11	0 3092 3233 3268	2021-04-18 10:45:06	Apache/2.4.29 (Ubuntu)
147.229.13.204	0	14	0 3092 3233 3268	2021-04-18 10:45:58	Apache/2.4.29 (Ubuntu)
147.229.13.205	0	11	0 3092 3233 3268	2021-04-18 10:46:50	Apache/2.4.29 (Ubuntu)
147.229.13.206	0	4	0	2021-04-18 10:47:45	lighttpd/1.4.35

147.229.13.216	0	11	0 3092 3233 3268	2021-04-18 10:49:59	Apache/2.4.29 (Ubuntu)
147.229.13.220	0	7	0 637 3233 3268	2021-04-18 10:50:53	Apache
147.229.13.221	0	11	0 3092 3233 3268	2021-04-18 10:53:32	Apache/2.4.29 (Ubuntu)
147.229.13.230	0	8	0 3233	2021-04-18 10:55:16	Apache/2.4.29 (Ubuntu)
147.229.13.231	0	9	0 3092 3233 3268	2021-04-18 10:56:09	Apache
147.229.14.29	0	3	0	2021-04-18 08:32:53	nginx
147.229.14.124	0	7	0 3233 5292	2021-04-18 09:32:44	nginx/1.14.2
147.229.14.248	0	3	0	2021-04-18 09:37:04	

Appendix C

GitHub Pages design

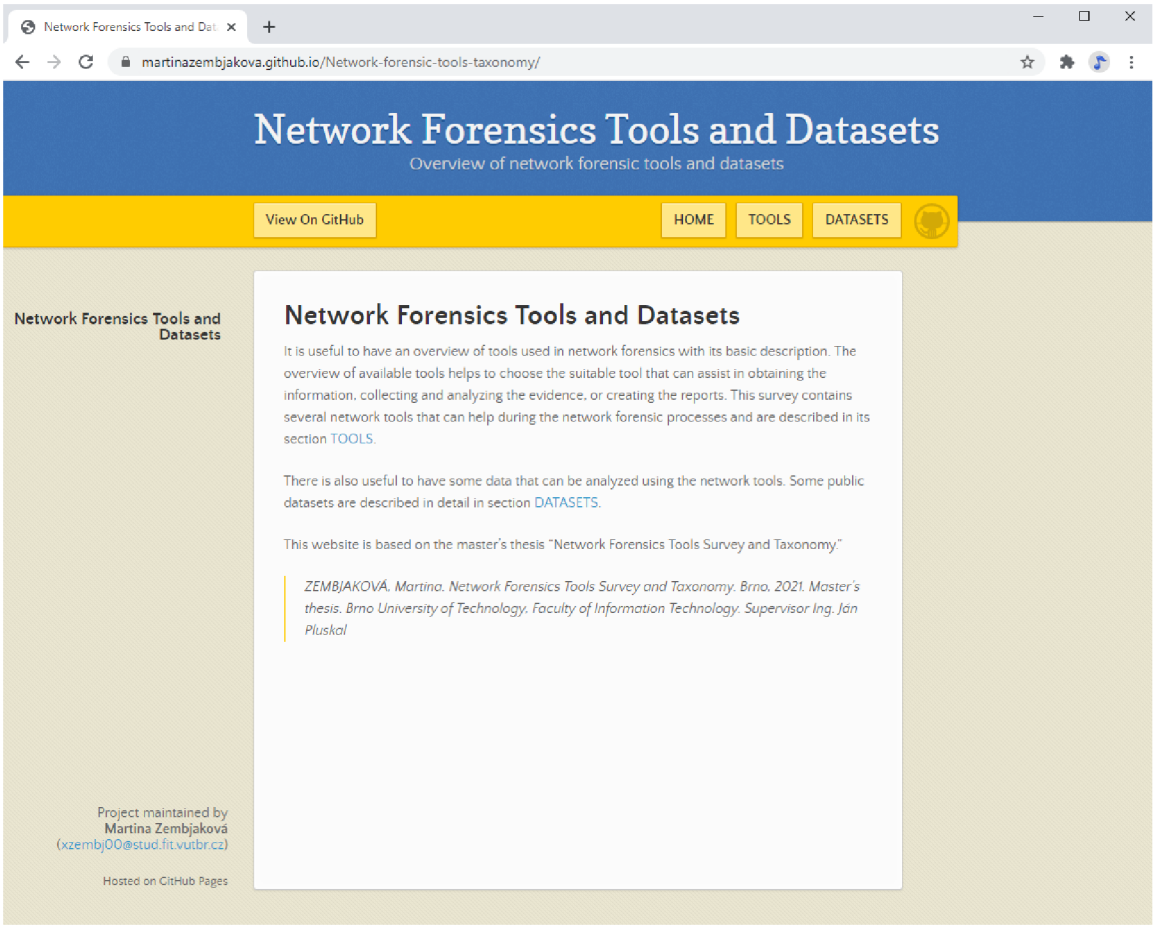


Figure C.1: Home page of the website

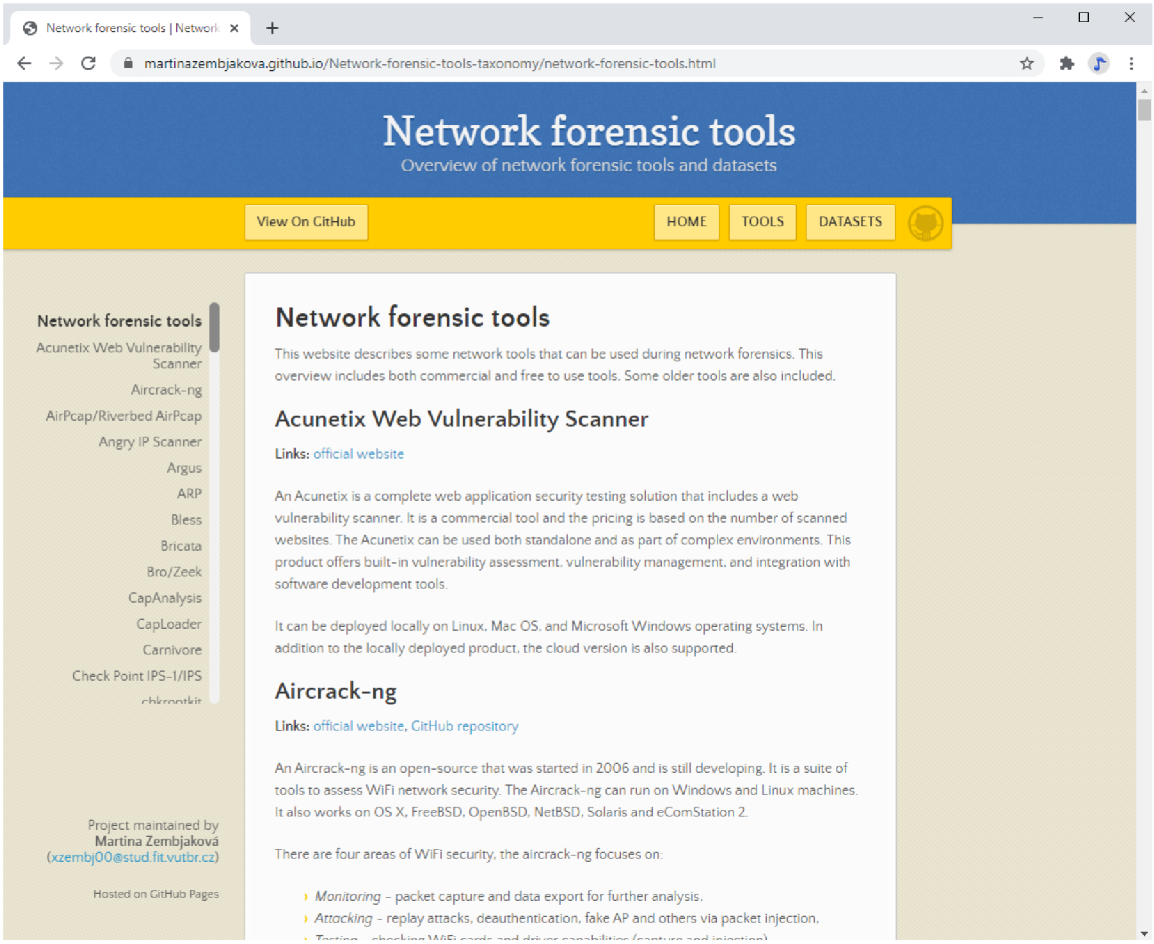


Figure C.2: Network forensic tools page of the website

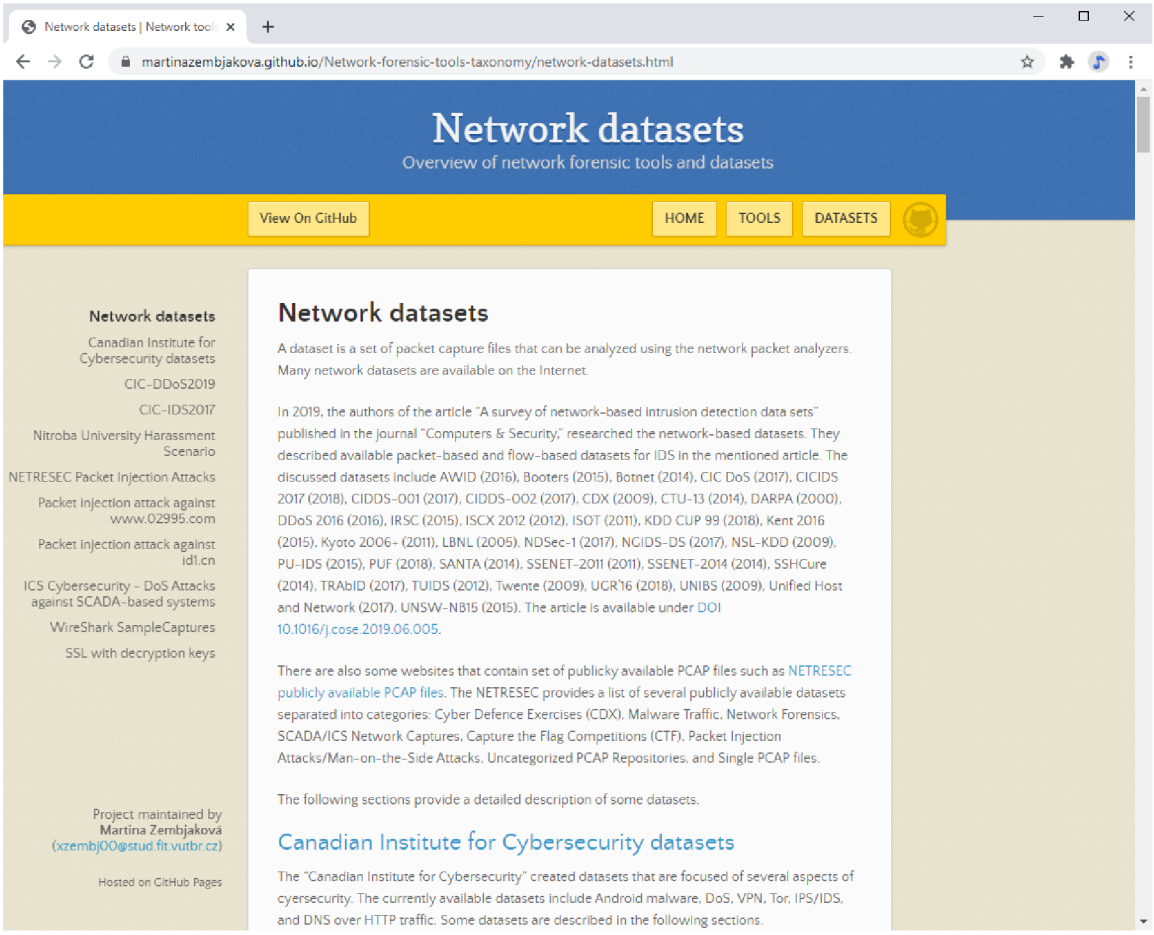
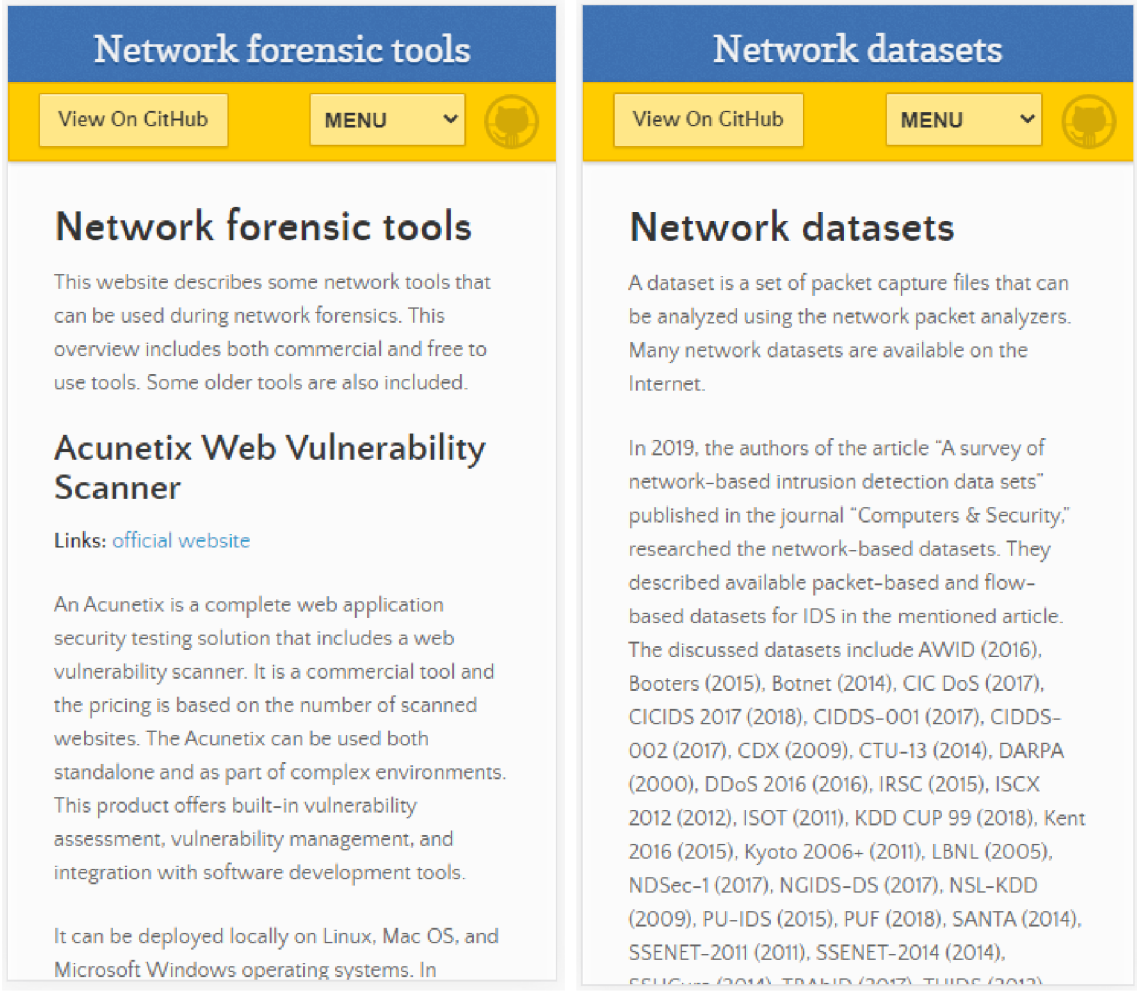


Figure C.3: Network datasets page of the website



(a) Tools page (b) Datasets page

Figure C.4: Mobile design of the website

Appendix D

Contents of the included storage media

An enclosed DVD disc contains following:

- this document in PDF format (xzembj00.pdf file)
- directory "latex-sources" with latex source of this document
- README.md file with detailed description of the DVD content
- directory "demonstration-reports" with results from the tools demonstration
  - directory "Scanners" with results from scanner tool from section 6.2
  - directory "Visualizers" with results from visualizers from section 6.4
  - directory "Analyzers" with results from analyzers from section 6.5
  - directory "Diagnostic tools" with results from diagnostic tools from section 6.6
  - directory "IDS-IPS" with results from IDS/IPS tools from section 6.7
- directory "datasets" with newly created datasets
  - directory "Web browser history data" for the dataset from section 7.3
  - directory "Multiple sniffers" for the dataset from section 7.1
  - directory "Todays traffic" for the dataset from section 7.2
- directory "github-pages"
  - directory "Network-forensic-tools-taxonomy" with source code of GitHub repository
  - directory "offline webpage" with offline HTML webpages
  - screenshots of the website