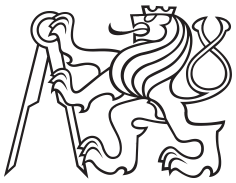
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

The first comprehensive report on the state of the security of mobile phones of civil society.

Jakub Čech

Supervisor: Ing. Sebastián García, Ph.D.
Supervisor–specialist: Ing. Veronica Valeros
Field of study: Open Informatics
Subfield: Cybersecurity
May 2020

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Čech Jakub** Personal ID number: **420855**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Cyber Security**

II. Master's thesis details

Master's thesis title in English:

The first comprehensive report on the state of the security of mobile phones of civil society

Master's thesis title in Czech:

První komplexní zpráva o stavu bezpečnosti mobilních telefonů občanské společnosti

Guidelines:

Analyze the state of the security of mobile phone's traffic capture of civil society. Summarize the findings, Categorize the vulnerabilities/issues, through the analysis of reports or traffic captures of the users of the Civilsphere Emergency VPN. If possible to propose ideas on 'why' this may happen.

Bibliography / sources:

[1] Q. Li and G. Clark, "Mobile Security: A Look Ahead," in IEEE Security & Privacy, vol. 11, no. 1, pp. 78-81, Jan.-Feb. 2013.
[2] Salamon, M. L. (1999). Global civil society.
[3] N. Leavitt, "Mobile Security: Finally a Serious Problem?," in Computer, vol. 44, no. 6, pp. 11-14, June 2011.
[4] Hardy, S., Crete-Nishihata, M., Kleemola, K., Senft, A., Sonne, B., Wiseman, G., ... & Deibert, R. J. (2014). Targeted threat index: Characterizing and quantifying politically-motivated targeted malware. In 23rd {USENIX} Security Symposium ({USENIX} Security 14) (pp. 527-541).

Name and workplace of master's thesis supervisor:

Ing. Sebastián García, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second master's thesis supervisor or consultant:

Ing. Veronica Valeros, Department of Computer Science, FEE

Date of master's thesis assignment: **05.02.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Sebastián García, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student's signature

Acknowledgements

I wish to express my sincere gratitude to my supervisors Ing. Sebastián García, Ph.D., and Ing. Veronica Valeros. They guided me and offered many valuable pieces of advice throughout this project. My thanks extend to everyone at Civilsphere Project for creating the EVPN service, providing the necessary data, and constructive thesis critique. My gratitude also belongs to my family for their material and emotional support during my studies, and to my girlfriend for believing in me. Lastly, to the Czech Technical University in Prague, my alma mater: So long and thanks for all the fish! [1]

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 20. května 2020

Abstract

Civil society members face threats not only in the physical world but in cyberspace. Their critical work leaves them in a permanent risk of surveillance and abuse. Mobile phones are vital for their activities, however these are often vastly unprotected. The lack of a standardized method to measure and analyze these risks hinders the efforts to protect them. The Civilsphere Project at the Czech Technical University in Prague created the Emergency VPN (EVPN) to help civil workers at risk. This free service helps discover data leaks or malware infections through network traffic analysis of mobile devices. The goal of this thesis is to create the first standardized risk measurement score for mobile phones at risk. In order to do so we processed 65 packet captures from the civil society along with the manual assessment reports done by Civilsphere analysts, creating a unique dataset suitable for further analysis. We assessed data leaked from mobile devices to identify potential risks and security threats. We developed a new method to standardize the severity rating and created a metric describing the nature of the reported data leaks. While none of the analyzed devices showed indications of malware presence, we discovered that they leak a lot of data that puts the civil workers at risk, most commonly the user's location.

Keywords: civil society, cybersecurity, mobile security, network analysis

Supervisor: Ing. Sebastián García, Ph.D.
Artifical Intelligence Center, FEE

Abstrakt

Členové občanské společnosti často čelí hrozbám nejen ve fyzickém světě, ale i v kybernetickém prostoru. Jejich důležitá práce je vystavuje neustálému riziku sledování nebo napadení. Mobilní telefony jsou pro jejich aktivitu nezbytné, ale zároveň často nedostatečně chráněné. Snahu o jejich ochranu často ztěžuje nedostatečná standardizace metod pro analýzu a hodnocení rizik. Projekt Civilsphere na Českém vysokém učením technickém v Praze poskytuje ohroženým lidem zdarma službu Emergency VPN (EVPN). Tato služba pomáhá objevit úniky dat a přítomnost malware prostřednictvím analýzy síťové komunikace mobilních telefonů. Cílem této diplomové práce je vytvořit první standardizované skóre popisující nebezpečí plynoucí rizikovým osobám z využívání mobilního telefonu. Zpracovali jsme zachycenou komunikaci ze 65 zařízení, včetně závěrečných hlášení o zranitelnostech, a vytvořili tak unikátní sbírku dat, vhodnou pro další analýzu. Prozkoumali jsme data uniklá z mobilních telefonů a identifikovali rizika z nich vyplývající. Vyvinuli jsme novou metodu hodnocení závažnosti zranitelností a popisu typů uniklých dat. Přestože žádné zařízení nevykazovalo známky přítomnosti malware, zjistili jsme, že z aplikací běžně unikají citlivá data, nejčastěji pozice, ohrožující bezpečí uživatelů.

Klíčová slova: občanská společnost, kybernetická bezpečnost, mobilní bezpečnost, síťová analýza

Překlad názvu: První komplexní zpráva o stavu bezpečnosti mobilních telefonů občanské společnosti

Contents

Figures

Tables

Chapter 1

Introduction

Nowadays, nearly half of the world population owns a smartphone [2]. Smartphones are constantly connected, receiving incoming calls at a nearby cell, asking a remote server for new messages from friends, and downloading the latest weather data and news. They communicate without the need of the user's actions. This constant communication leaves huge traces of our behavior and actions. A determined attacker can leverage these traces to follow, stalk or intercept an individual in the real world. If the communication is not correctly secured, our private information, but more importantly, our lives may be at risk.

To understand the every day risk of civil society workers lets explore a fictional example. Lets imagine our neighbor Bob. He is an investigative journalist at a small newspaper focused on revealing conspiracies and calling out on the government's incompetency. It is Spring 2020, and the world is in the middle of the worldwide COVID-19 pandemic. Bob's local government was not smart in responding to the crisis. Bob dived deep into the subject and revealed that the current government dismantled the country's healthcare infrastructure developed by the previous government. Bob did a thorough research and published his findings. His article went viral overnight and was featured in every major newspaper. The government was furious and desired to silence or discredit Bob. This is unfortunately a very common situation. The government asked the Internet Services Providers to tap and send them all of Bob's mobile traffic, which is legal in most countries with a warrant, and done in many more countries without a warrant. The government now could see everything Bob does on his phone. They read his unencrypted private messages, listened to calls, read SMS, access unencrypted files sent from his device, and see his complete internet behavior activity. Bob was careful, but he made a critical mistake: he checked the weather forecast on his phone. The application he used does not have a good privacy security and retrieves the forecast by sending the exact GPS coordinates in an unencrypted request. The risk of Bob's actions put him in jeopardy and the government has all the information to act by threatening him, harass him, follow him, kidnap him or kill him.

The Civilsphere project at the Czech Technical University in Prague (CTU) is dedicated to provide free services to people at risk to detect targeted cyber security threats[1]. The Emergency VPN is a free service provided by Civilsphere, and offers a chance for people like Bob to see what the attacker sees and how his phone can put him at risk. The most common way to assess the security of a mobile device is through digital forensics which requires physical access to the device. While examining the device physically is undoubtedly the best, it is not necessarily the most practical way. The user needs to get the device to a forensics firm, either send it through an intermediary or deliver it in person. The transfer of ownership poses a

[1] More information available at `https://www.civilsphereproject.org/`

security risk in itself and could do more harm than good. The Emergency VPN service provides a security assessment through the device's network traffic only. Spyware or malware needs to communicate with the outside world to extract sensitive data. While analyzing solely the network traffic can not be as comprehensive as a physical inspection, it is a good alternative to avoid the physical transfer of the device.

Today, Civilsphere considers each device as a completely new project. More than one person creates the reports, and the severity ranking of vulnerabilities is often determined only by intuition and experience. This practice makes it hard to compare the vulnerabilities among devices. Without understanding the common problems, Civilsphere cannot offer targeted advice to its users.

While past studies focused on highlighting cybersecurity issues and attacks on civil society members [3], privacy issues on mobile devices [4] and leaks of personal information from mobile network traffic [5], there is a gap in the understanding of the risks civil society members are exposed through by owning a mobile device. Our research aims to fill this gap.

The aim of this thesis is to understand the current state of data privacy and security of mobile devices owned by civil society members, and develop a Standardized Risk Score, through the analysis of network traffic captured by the Civilsphere Project. We conduct this research in three phases. First, we study and process the network data of individuals at risk to create a dataset. Second, we perform an extensive analysis of the data to understand the threats and risk of users. Third, we use our new understanding of the threats and risks to develop a Standardized Risk Score.

In our research, we include only traffic generated by mobile devices. In today's world, mobile phones act as beacons attached to our bodies. Most people never spend a significant time without their phones at hand. Personal devices can reveal a vast amount of information about their owners. The traffic could hide an answer to questions like "What is their news source?", "Where are they now?", or "Do they drive or use public transport?". We decided to exclude computers because they are not as attached to the user and can be shared. Another reason for this decision was the availability of the captured traffic.

As the basis of our work, we used network captures from Civilsphere Emergency VPN. For further analysis, we expanded the dataset using reports describing vulnerabilities discovered in devices, which were created as part of the Emergency VPN service by Civilsphere members. We converted the dataset into a computer-readable form and imported the data into an SQL

database. We attempted to identify ads and trackers inside the network traffic and annotated the positive domains accordingly. We explored and understood the data through database queries, which we then transformed into graphs. Lastly, we developed a standardized way to rate and describe the privacy-related vulnerabilities discovered in the devices which use Emergency VPN. The agreement between EVPN and its users, protecting the privacy of their data, is of paramount importance to us.

This thesis analyzed the overall state of phones examined with the Emergency VPN and makes several contributions:

- The first contribution is an extensive analysis of network traffic of members of the civil society. In Chapter 6, we describe the analysis of data collected from 65 smartphones, provide an analysis of the risks, and the discovered vulnerabilities.
- The second contribution is creating a new standardized categorization of privacy-related threats with a user risk rating method. Chapter 7 provides details of the tracked metrics and calculations of user risk and exposure scores.
- The third contribution is two compiled lists of domains to identify mobile devices and manufacturers, which are briefly described in Section 6.2. The complete lists are available in the thesis repository. [6]

The Civilsphere project will implement the results of our work in its future projects. We can build on the foundation laid by this thesis after Civilsphere evaluates contributions in the production environment and formulates feedback. The standardized threat table will ease the researchers' burden when rating the threats identified in the Civilsphere Emergency VPN devices. The overview of the past analyses can be used to improve the analysis of future network captures, decreasing the manual analysis time.

Chapter 2

Background

2.1 Civil Society

In this thesis, we plan to analyze the mobile traffic of civil society defendants. Before we can move further, we need to cover some key concepts. First, we need to understand what civil society is, its history, and reasons for existence. We will also examine the challenges and threats the civil society is currently facing. Secondly, we will describe the state of mobile internet and standard network attack methods. In this thesis, we also propose a new way to standardize privacy-related cybersecurity threats. Therefore, in the last part of this chapter, we will explain an existing industry standard for rating vulnerabilities.

2.1.1 Definition

There is no universally accepted definition of the term civil society. A widely cited definition is the one used by the Centre of Civil Society at London School of Economics: "Civil society refers to the arena of uncoerced collective action around shared interests, purposes and values. In theory, its institutional forms are distinct from those of the state, family and market, though in practice, the boundaries between state, civil society, family and market are often complex, blurred and negotiated. Civil society commonly embraces a diversity of spaces, actors and institutional forms, varying in their degree of formality, autonomy and power. Civil societies are often populated by organisations such as registered charities, development non-governmental organisations, community groups, women's organisations, faith-based organisations, professional associations, trade unions, self-help groups, social movements, business associations, coalitions and advocacy groups."[7]

The term civil society is also commonly used with much broader definitions. We can see an example in the Collins English Dictionary: "the organizations within a society that work to promote the common good, usually taken to include state-run institutions, families, charities, and community groups."[8]

2.1.2 History

We can trace the civil society back to the works of ancient Greek and Roman philosophers. For them, civil society equaled the state. The problem of

civil society emerged again in the Age of Enlightenment, a philosophical movement between the 17th and 19th centuries. Influential philosophers John Locke (Two Treatises of Government), Thomas Hobbes (Leviathan), Jean-Jacques Rousseau (Discourse on Inequality, The Social Contract) based their governance philosophies in the social contract theory. The term became less used in the mid-19th century as political philosophers studied the impacts of the industrial revolution on society.

After World War II, Marxist theorist Antonio Gramsci revived the term to portray civil society as the particular nucleus of political activity and a crucial sphere of struggle against tyranny. His books influenced people fighting against dictatorships. Czech, Hungarian, and Polish activists identified themselves with civil society. [9]

The current popularity of the term civil society originates in the 1990s. The world started moving more towards democratic forms of government, which opened the doors to forming civil societies in former dictatorships. The public was tired of the conventional party system and saw the civil society as a social renewal. The rise of the Internet allowed us to forge connections with people all around the globe. Availability of information and accessibility to different worldviews empowered citizens. The civil society became the critical element of the information era zeitgeist [9].

2.1.3 Social Contract Theory

The Social Contract Theory does not study the origins of society but rather the terms on which individual society is governed. According to J. W. Gough in The Social Contract: “The people have made a contract with their ruler which determines their relations with him. They promise him obedience, while he promises his protection and good government. While he keeps his part of the bargain, they must keep theirs, but if he misgoverns the contract is broken and allegiance is at an end.”[10] Nowadays, activists do not believe governments follow the social contract.

If social hierarchies, law, and government are missing, each individual has unlimited freedom. Everyone has the freedom to murder, rape, and steal. Thomas Hobbs argues that the absence of the political community would lead to a "war of all against all" (*bellum omnium contra omnes*). In his most famous book Leviathan, he expands on the problem of the state of nature, i.e., life without government: “In such condition, there is no place for industry; because the fruit thereof is uncertain: and consequently no culture of the

earth; no navigation, nor use of the commodities that may be imported by sea; no commodious building; no instruments of moving, and removing, such things as require much force; no knowledge of the face of the earth; no account of time; no arts; no letters; no society; and which is worst of all, continual fear, and danger of violent death; and the life of man, solitary, poor, nasty, brutish, and short."[11]

2.1.4 Dangers to Civil Society

There is an increasing number of records suggesting some states are attempting to suppress civil society organizations or create unnecessary hurdles to prevent them from functioning effectively.

In 2018, the European Union Agency for Fundamental Rights (FRA) published a report detailing the challenges civil society organizations are facing in the EU [12]. The FRA discovered some states have too restrictive public assembly rules and thus limiting the free expression of potential assembly participants. Several member states maintain criminal laws banning defamation or insult of state officials and the state itself. If the laws are applied too strictly, it may prevent civil society actors from criticizing the state officials, knowing they may face criminal sanctions. An example of such law outside the European Union is The Political Parties, Groups and Movements Act in the Dominican Republic. It penalizes negative comments on social media against candidates in political campaigns with a prison sentence up to 10 years [13].

FRA discovered that some countries lead negative media campaigns against organizations or individuals that receive foreign funding. The campaigns require them to brand themselves as foreign-funded organizations on all their materials. A recent example outside of the EU is Russia's "foreign agent" law, introduced in 2012 and amended in 2019. Independent journalists, bloggers, and non-governmental organizations which receive funding from abroad and engage in political activity can be labeled as "foreign agents" [14].

Many organizations face obstacles when attempting to participate in the law or decision-making process. According to the FRA report, civil society organizations have limited access to information about policies and legal initiatives. In the Czech Republic, there is a notable competition among state departments and non-governmental organizations, further fueling the lack of trust and mutual respect [12].

Violations	Percentage
Detention/arrest	22%
Legal action	20%
Physical attack	13%
Threats	10%
Raid/break in	6%
Disappearance	4%
Torture/ill-treatment	3%
Questioning/interrogation	3%
Smear campaign	3%
Verbal abuse	2%
Travel ban	1%
Sexual violence	<1%

Table 2.1: Types of violations (excluding killings) against human rights defenders as reported by the Front Line Defenders organization [13]

2.1.5 Threats to Human Rights Defenders

Human rights defenders (HRDs) fight to protect the civil rights, and to preserve and improve the freedom of individuals. The leaderships of many countries specifically target HRDs. Front Line Defenders, an organization that seeks to protect human rights defenders at risk, published its statistics on how HRDs are targeted [13]. The data are based on 895 reported violations from January 1st through December 18th, 2019. They only represent violations where the human right defender requested the Front Line Defenders to conduct public advocacy of their case. Table 2.1 presents the most common reported violations, and Table 2.2 presents the distribution of violations by region. Neither table includes killings.

The statistics show that more than 42% of incidents involved detention or legal action. What is even more crucial to observe, 26% of violations included physical attacks, break-ins, disappearances, and torture.

According to the Front Line Defenders' report, 304 human rights defenders were killed in the year 2019 alone. More than a third of deaths, 103, occurred in Colombia, followed by the Philippines, 43, and then by Honduras and Mexico, both with 23 HRDs killed. Detailed statistics are shown in Figure 2.1. The most dangerous sectors are land rights, indigenous peoples' rights, and environmental rights "due to the profit-driven exploitation of natural resources, combined with rampant corruption, weak governments, and systemic poverty."

Violation	Africa	Americas	Asia	MENA	ECA
Detention/arrest	16%	15%	10%	31%	7%
Legal action	11%	12%	13%	17%	17%
Threats/verbal abuse	7%	25%	11%	3%	7%
Physical attack	6%	12%	6%	15%	6%
Raid/break in	35%	5%	5%	3%	7%

Table 2.2: Most commonly reported violations (excluding killings) against human rights defenders by region reported by the Front Line Defenders organization [13]

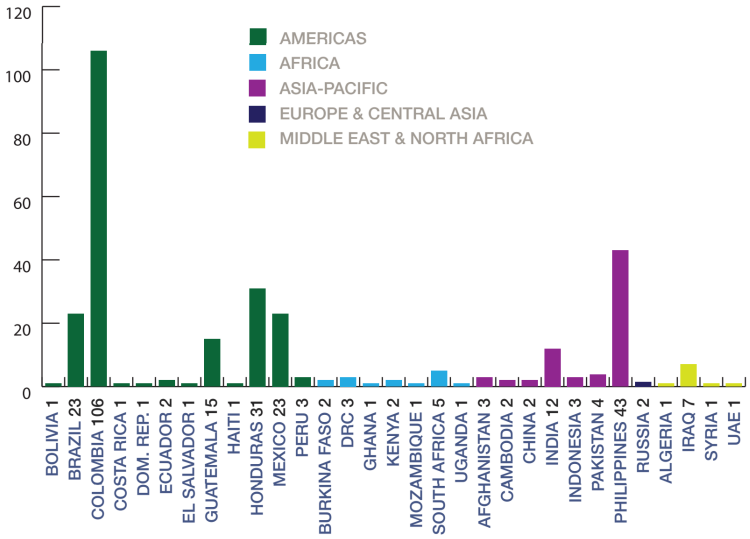


Figure 2.1: Killed human rights defenders by country reported by the Front Line Defenders organization [13]

The danger to activists, and people involved in non-governmental organization is real. The digital world, internet, and mobile devices increase the risk as it became easier for governments to act in the cyber space.

2.2 Mobile Internet Communication

2.2.1 Mobile Internet

According to the data published by the International Telecommunication Union (ITU), internet usage is steadily rising [15]. (Figure 2.2) In 2019,

the ITU estimated that more than 53 % of the world's population used the internet; this results in about 4.1 billion users worldwide.

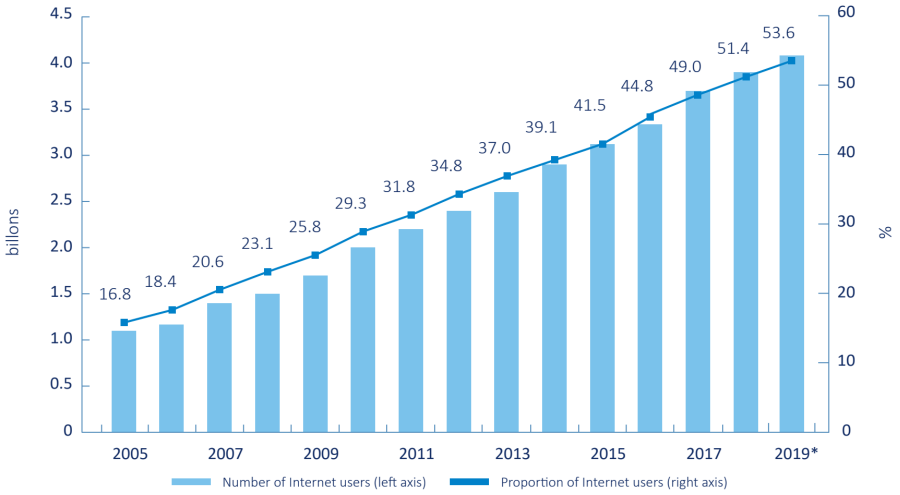


Figure 2.2: Individuals using the Internet, including the fraction of hte total Earth's population, reported by the International Telecommunication Union [15]

Mobile phones are making the internet more accessible than before. As we can see in Figure 2.3, in 2019, developing countries had 103.8 mobile-cellular subscriptions per 100 inhabitants and 75.2 active mobile-broadband subscriptions; that means about 72 % of all cellular subscriptions in developing countries have access to always-on high-speed internet. The number of mobile internet users is even higher in developed countries, where out of 128.9 mobile cellular subscriptions, about 94 % have high-speed internet access.

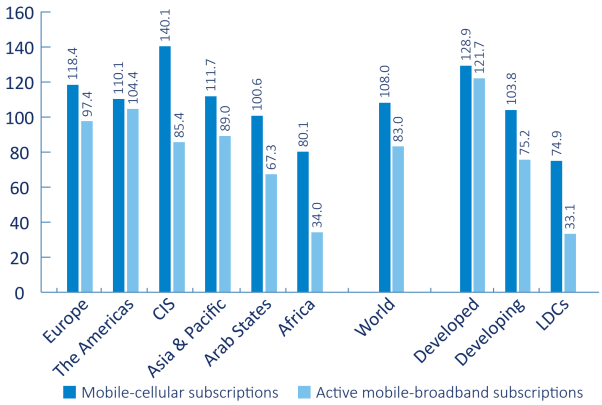


Figure 2.3: Mobile-cellular and mobile-broadband subscriptions per 100 inhabitants in 2019 reported by the International Telecommunication Union [15]

2.2.2 Sources of Network Traffic

There are two primary sources of internet traffic on mobile devices. The first one is the operating system (OS). A user usually has very little control over the OS components or the system itself. While it is technically possible to install Android on an iPhone[1], because Android is open-source, the phone would become unusable in daily life. In the case of Android devices, it is usually possible to change Android flavor without switching to a different device or compromising the device's usability. Changing an operating system requires a particular computer skillset, and we cannot expect a standard user to be able to do so; therefore, an average user can change the OS only by changing the device itself.

The second source of internet traffic is mobile applications. There are two types of applications: pre-installed applications by the manufacturers, and applications installed by the user. Pre-installed applications are expected to be secure, but often present risks [16]. Most of the applications on a mobile phone are installed by the users. Most popular applications, published on the app stores of both platforms, include advertisement and trackers. Advertising companies, such as Google or Facebook, provide ready to use frameworks that developers can include in their mobile application. A downside of this approach is that users need to worry not only about what data their phone sends to the application publisher but also to unknown third parties.

Mobile phones generate large amounts of network traffic, even when they are not in use. Figure 2.4 presents the number of DNS requests we collected in a small home network with ten connected devices - five Android phones, two iOS devices, and three computers. During the night, despite everyone being asleep, and laptops turned off, mobile phones generated 470 DNS requests per hour on average. Pi-hole[2] determined close to half of the requests as ads or trackers. While we can expect some communication, for example from instant messengers or e-mail clients, often the phone is communicating unnecessarily.

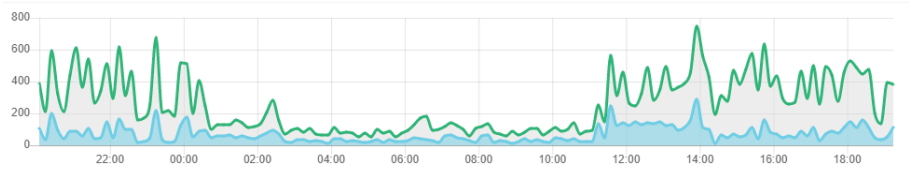


Figure 2.4: DNS queries in 24 hour period in our small home network. Blue line represents ads and trackers.

[1]Project Sandcastle. Available from: `https://projectsandcastle.org/`

[2]The Pi-hole® is a DNS sinkhole that protects your devices from unwanted content, without installing any client-side software. Avalailable from: `https://pi-hole.net/`

2.2.3 Wireless Network Attacks

Phones rarely use a wired connection as opposed to Wi-Fi, which opens them up to a range of attack vectors. For the purpose of this thesis, the most important type of attacks are those in which the adversary can capture the target's network communication or even alter its content. In other words, when an attacker is in the middle of the communication. This type of attack is called Man-in-the-Middle (MITM). An example of MITM attack is depicted in Figure 2.5. An attacker secretly positions himself in between two parties. Then he can, for example, start eavesdropping. He makes two independent connections, one to the victim and the second one to the intended (or even unintended) server. Then, the attacker relays and possibly alters the network traffic.

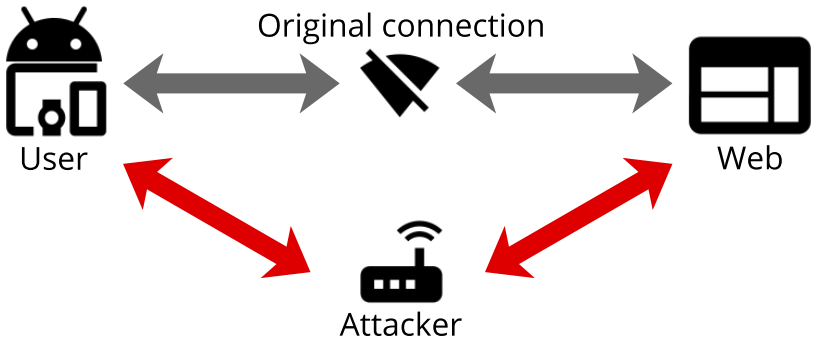


Figure 2.5: Diagram of Man-in-the-middle attack.

Any subject on a path from the origin to the target is "in the middle." One, always present subject, is the internet service provider. There are reports of Comcast, a large telecommunication company in the USA, injecting its code to visited websites, which then displayed advertisements or account notices [17, 18]. Nokia is behind another notable instance of a MITM attack in 2013. Its Xpress Browser redirected the web traffic to Nokia's servers, where it decrypted data transported via HTTPS connections, giving the company access to the clear text content of the traffic. Therefore Nokia could read, for example, users' credentials, banking information, and credit cards [19].

To perform a MITM attack, an external attacker, i.e., not an ISP, needs to find a way to route a phone's network traffic through his device. There are two easy ways to do that, one which requires user action and one that does not. The first method is a fake access point. An attacker goes to a public place, such as an airport or a restaurant. Then he creates a wireless access point with a believable name - AIRPORT FREE WIFI or La Tortilla - Public. Customers or passengers might not think twice before connecting to an unknown access point. The second way is the Evil Twin attack, in which the attacker replicates parameters of an existing access point that the

victim has already used. The target's device may connect to the malicious access point instead of the legitimate one after it is forced to disconnect several times. The malicious access point connected to the internet is no different from any other, from a user's perspective. However, an attacker has access to all traffic flowing through his device. He can read and intercept any unencrypted traffic.

The form of man-in-the-middle attack is the most likely method an attacker would choose to capture the target's network traffic. The Emergency VPN server captures the network traffic for future analysis but, to protect the security of civil society members, it cannot intercept and alter the traffic in any way. However, the packet captures we study in this thesis are very similar to ones, that an attacker would obtain during a MITM attack.

2.3 Common Vulnerability Scoring System

CVSS is one of the most used and recognized metrics worldwide. Its purpose is to describe general threats but it does not suit well the needs of this thesis. Nevertheless, we need to understand this unofficial industry standard before we can develop our specialized metric.

"The Common Vulnerability Scoring System (CVSS) is an open framework for communicating the characteristics and severity of software vulnerabilities. CVSS consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic qualities of a vulnerability that are constant over time and across user environments, the Temporal group reflects the characteristics of a vulnerability that change over time, and the Environmental group represents the characteristics of a vulnerability that are unique to a user's environment. The Base metrics produce a score ranging from 0 to 10, which can then be modified by scoring the Temporal and Environmental metrics. A CVSS score is also represented as a vector string, a compressed textual representation of the values used to derive the score." [20] CVSS version 3.1, released in June 2019, contains three metric groups. There are 22 individual metrics in total.

In CVSS, security analysts assign values to Base and Temporal metrics. The end-user is then responsible for assigning the Environmental metrics, which modify the scores of Base and Temporal metrics, because they are environment-specific and depend on the actual implementation of the vulnerable component in the target system. We won't cover the Environmental metrics in this thesis,

as we cannot know the conditions of the user's real-world situation. The total CVSS score is calculated using standardized and well-defined equations, which are published in the specification document.

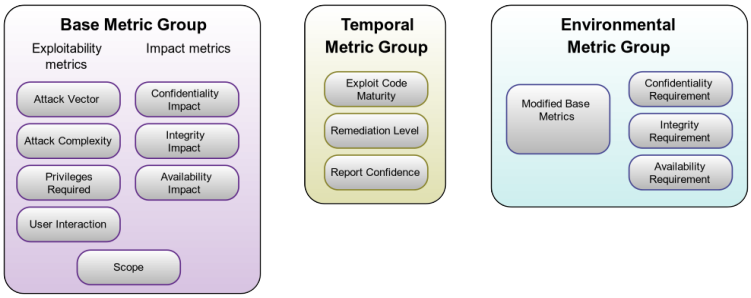


Figure 2.6: Structure of CVSS metrics. [20]

2.3.1 Base Metric Group

The Base Metric Group is the most important of the three. Every metric in the base group is mandatory. The metrics describe the intrinsic qualities of a vulnerability that are constant, no matter when or where an attacker exploits the vulnerability. The scoring should be independent of the attacker's knowledge of the target system, and the analyst should assume the attacker has advanced knowledge of the target system's configuration and defense mechanisms. The Base Metric Group is divided into three subgroups, as depicted in Figure 2.6: exploitability metrics, Impact metrics, and Scope.

The first subgroup in the Base metric group is Exploitability. Metrics belonging to the subgroup reflect the properties that can lead to a successful exploit. The metrics are:

- Attack Vector (AV) - describes how remote the attacker must be from the system
  - Network - remotely exploitable, e.g., from the Internet
  - Adjacent - same physical or logical network, e.g., LAN
  - Local - path via read/write/execute capabilities, e.g., SSH access to a console, user interaction through social engineering
  - Physical - physical access to the component, e.g., access to FireWire
- Attack Complexity (AC) - describes whether there are conditions which must exist and which the attacker cannot control

  - Low - specialized conditions or circumstances do not exist, i.e., deterministic, repeatable success can be expected during the exploitation
  - High - success depends on conditions the attacker cannot control
- Privileges Required (PR) - describes the level of privileges required for successful exploitation
  - None - no authorization required
  - Low - basic user privileges, e.g., read-write access to the user's files, or access to non-sensitive data
  - High - significant privileges, e.g., administrator or root account
- User Interaction (UI) - describes whether another human needs to perform an action to exploit the vulnerability
  - None - no action needed
  - Required - action must be taken by a user, e.g., the user must install an application

The second subgroup is Impact. Metrics in this subgroup describe the effects and consequences of a successful exploit. They consider the worst reasonable outcome and are scored based on the increase of the privilege level, access, or other adverse outcomes. Metrics belonging to the category are:

- Confidentiality - describes the disclosure of confidential information to an unauthorized attacker
  - High - all resources within the component are accessible by the attacker, or the disclosed information has a severe impact, e.g., private encryption keys
  - Low - access to a limited set of information, the attacker cannot control what data can access
  - None - no loss of confidentiality
- Integrity - describes the impact on the trustworthiness of the data after a successful exploitation
  - High - total loss of integrity or only some files can be modified, but with severe consequences, e.g., adding a key to the list of trusted SSH clients
  - Low - the attacker can modify the data but cannot control the consequences of the edit
  - None - no loss of integrity

- Availability - describes the effect on the availability of the component
  - High - total loss of availability, can be in effect only during the attack or even after the attacker stops any actions, e.g., a Denial-Of-Service attack
  - Low - the exploitation leads to reduced availability or performance but not to a total denial of service
  - None - no impact on the availability

Last but not least is the Scope metric, which describes whether the successful exploitation of a vulnerability in a component impacts resources in a different security scope. Security scope is the collection of subjects under one security authority, a mechanism that controls the access to objects and resources, e.g., a sandbox environment. The value of the metric can be either Changed or Unchanged.

2.3.2 Temporal Metric Group

Temporal metrics can change over time and describe the current state of an exploit. For example, when a vendor releases a patch fixing the problem, the temporal score decreases. When code for the exploit is released into the wild, the temporal score increases.

The following metrics belong to the Temporal metrics group:

- Exploit Code Maturity - describes the current state of exploit techniques, exploit code availability, and whether there are active cases of exploitation
  - Not Defined - there is an insufficient amount of information to choose other values
  - High - functional, reliable, autonomous code exists, and the exploit always works or actively autonomously delivered, e.g., exploit included in automated tools, or viruses
  - Functional - functional code that works in most cases is available
  - Proof-of-Concept - the code is only a demonstration, or the attack is not practical in most system
  - Unproven - no code is available, or the exploit is only theoretical
- Remediation Level - describes the state of remediation of the exploit

- Not Defined - there is an insufficient amount of information to choose other values
- Unavailable - the exploit has no, or impossible to apply, remediation available
- Workaround - users, affected by the vulnerability, published an unofficial solution
- Temporary Fix - the vendor published official temporary hot-fix or workaround
- Official Fix - the vendor published official patch or an upgrade

- Report Confidence - describes the confidence in the existence of the vulnerability
  - Not Defined - there is an insufficient amount of information to choose other values
  - Confirmed - the vulnerability was officially confirmed, or detailed reports are verifying its existence
  - Reasonable - security researches are reasonably confident, and the vulnerability is reproducible with some verified impact
  - Unknown - reports indicate the impacts of vulnerability, but the root cause has not been discovered

Chapter 3

Related Work

3.1 Cyberthreats to Civil Society

A paper about the cybersecurity of high-risk users by John Scott-Railton from Citizen Lab [3], provides a basic understanding of the nature of cybersecurity threats targeted at civil society. The study shows that members of civil society often use unmanaged devices and extensively use online platforms and social networks. Their security protection is usually only behavioral precautions and widely available antivirus software. Later in the paper, the author gives an overview of the common actors behind the cyberattack, including cyber-militias and nation-sponsored actors. The author ends with a plea to the providers of software to make default settings more secure and thus automatically improving the cybersecurity of ordinary users.

A more recent work, Cybersecurity for Civil Society [21], explores security and privacy concerns of Civil Society Organizations. Authors discovered that the guidance often offered by cybersecurity professionals is not always applicable to the environment of civil society. Organizations are usually under-funded and have limited capacity for strategic planning. They propose improving the cybersecurity literacy of the members of civil society as well as improving the security patterns used during software development.

Although the past studies offer a good overview of the reason causing inadequate cybersecurity of civil society members, they do not offer interpretations of the risks created by the data leaks resulting from the existing software vulnerabilities. Our challenge is to understand how the cybersecurity threats are connected to leaks of private information and how the leaked data affect civil society members.

3.2 Leaks of Private Data

In the study ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic [5], authors present an automated way of discovering personally identifiable information (PII) leaks in network traffic. Their software uses machine learning to recognize patterns in network traffic signifying PII leaks. In the study, they verified the effectiveness of the detection by conducting a study with 92 participants, which revealed their solution is more accurate and reveals a broader range of PII leaks than other existing solutions.

Another research group has performed a large scale study of Android applications using their system ANDRUBIS [22]. The system combines static with dynamic analysis of the applications. The group created a dataset of more than one million Android applications, 40% of which were malicious. Using the generated dataset, the researchers observed trends in malware behavior between the years 2010 and 2014. They identified typical patterns between malware and its Command & Control servers and, for example, produced statistics on the difference in the network communication between malware and goodware.

Nevertheless, past studies do not focus on the risks which data leaks and software vulnerabilities impose on exposed users. Our challenge is to understand how data leaks impact activists, human rights defenders, and other civil society members.

3.3 Standardization of Cybersecurity Threat Rating

While there is no universally accepted standard for classifying mobile data leaks specifically, there is a respected industry standard for rating the severity of software vulnerabilities. One of the standards is CVSS [20]. The system provides a way to capture the main characteristics of a vulnerability and to produce a numerical score reflecting its severity. The standard is aimed at companies, to aid them during prioritization of addressing the discovered vulnerabilities. We described CVSS in detail in the Section 2.3.

However, our challenge is to provide a standardized risk score method for the user. The current metrics describe the implications for software and companies; therefore, we are aiming to describe risks to the human user caused by a software vulnerability.

Chapter 4

Methodology

We aimed to assess the state of security of mobile devices from the perspective of an attacker who has access only to network traffic of the target device. An adversary could capture the traffic himself, or for example, order the target's internet service provider to monitor and capture all incoming and outgoing traffic from the device. For such analysis, we needed to obtain many network traffic captures from a diverse set of devices. We could not generate the traffic ourselves because it would not necessarily reflect the state of the real-world devices.

We build on the data obtained from Cvilisphere project's EmergencyVPN service. The data includes completed reports sent back to the users, as well as the metadata from the network captures. Metadata contains information about network flows (time, size, duration, target), DNS requests, and HTTP traffic. For this research we obtained 65 PDF reports (ranging from 3 to 20 pages each), and 65 files containing network flows of the respective capture (thousands of flows each).

To ease future analysis, we designed and created a structured database, which we can easily query. We set on using a MySQL database[1]. MySQL offers an extensive suite of tools for quick database deployment. We created a model (Figure 4.1), which we then forward engineered to the actual database structure. To extract and compile useful information from the captured traffic, we used a combination of manual and automatic data analysis processed that will be described in the next sub sections.

4.1 Manual Data Analysis

Members of Civilsphere created a report for each assessed device, which informs the user about threats found within the network communication. Each finding contains information about the severity, name, summary, and lastly, a detailed description of the threat. In the first phase, we reviewed every entry from each report and associated the threats with the devices in the database. From all the reports, we created a unique list of threats found.

Since we aimed to analyze the nature of the threats, we could not rely solely on the threats text description found in the reports. Therefore, we proceeded to the next phase, where we enriched the data. First, we identified the device model from the network flows and unencrypted traffic. Second,

[1]MySQL is a popular open-source relational database management system. More information is available at `https://www.mysql.com/`

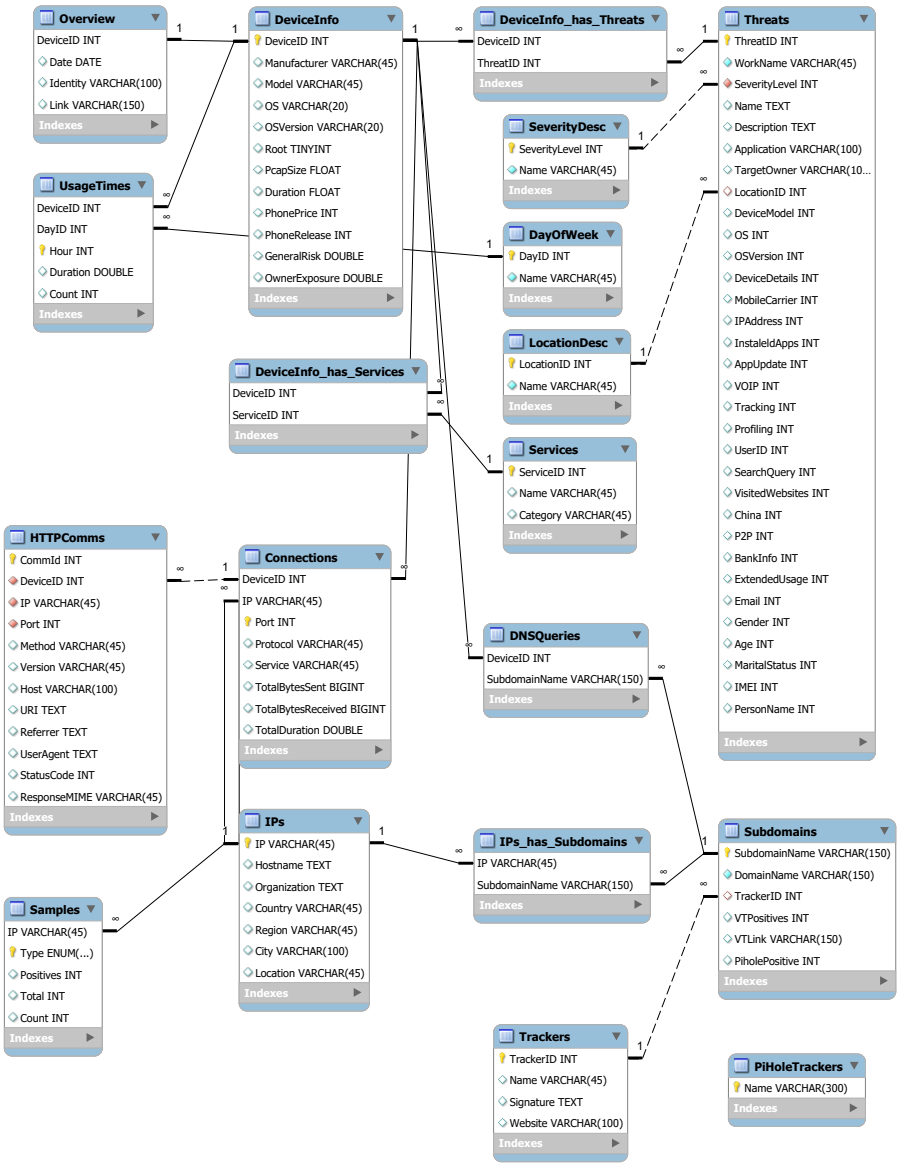


Figure 4.1: Model of the MySQL database used for storing the dataset.

we used this information to look up the device market price and launch date manually. Third, whenever possible, we identified the application that caused the security issue. Fourth, we looked up the developer of the application.

After collecting all details of the reported threats, we analyzed the data to see what types of information leaked. When we identified these types, we analyzed the data again and for every threat we annotated if it was a data leak, which type, and the level of the data leak. The classified data leak types served as a basis for our standardized threat table, which we created after analyzing the impacts of each leak type on the user's privacy. The methodology of the standardized threat table creation is described in more detail in Section 7.2.1.

At the end of the first phase of manual analysis, our database contained a list of devices, their details, and all reported threats. Each device was associated with every threat it contained. Threats were rated according to our threat table.

4.2 Automated Data Analysis

The initial manual analysis provided interesting data, however there was still a considerable amount of available information in the packet captures themselves. The reports offer a high-level overview of discovered threats. We had to extract details of connections and DNS requests via automated means to understand any underlying problems and relationships.

The Emergency VPN server processes every network capture by Zeek (formerly Bro)[2], which is a network security monitoring tool. Zeek produces several files. We focused on the following three, as they provide data most relevant to our study:

- `conn.log` - TCP/UDP/ICMP connections
  - Time, destination IP, port, protocol, duration, packet size...
- `dns.log` - DNS activity
  - Server IP, domain name, resolved IPs...

[2]Zeek is an open source network security monitoring tool. More information is available at `https://zeek.org/`

- `http.log` - HTTP traffic
  - URL, User-Agent, response code...

A more detailed description of the files mentioned above is presented later in Section 5.1.1. With these files in mind, we developed a set of Python scripts for automated processing. Our goal was to extract potentially useful information from the files, structure it so it matches our database model, and lastly, output it in CSV. We settled on this file format because we not only wanted to have the data in the MySQL database but also in a large spreadsheet.

The main script looks up all the necessary files on the Emergency VPN server. In the next steps, scripts extract data from Zeek logs and, in some cases, create summarize them. We then store everything in the database. After gathering the data, we augment them. The scripts look up details about IP addresses and research their connection to known malware or malicious sites. Section 5.1.2 provides a depper look into the data processing.

Every major step in the script is standalone. If needed, we can run only DNS extraction or, for example, VirusTotal analysis. The scripts are available in the thesis repository. [6]

Chapter 5

Datasets

In this section we describe the process of turning the raw input data from Emergency VPN, packet captures and PDF reports, into a full dataset containing all the features needed for further analysis in a format suitable for fast data processing.

5.1 Network Captures

To understand the state of the security of the mobile devices of the civil society, we require a substantially large set of samples. Therefore, it is important to gain access to network traffic of the civil society members, which we can then analyze and attempt to discover threats and risks. The PDF reports created by Civilsphere are based on the analysis of network data. The PDF reports and the network data used to create them are considered Raw Data.

5.1.1 Raw Data

The Civilsphere Project temporarily stores the network captures from the Emergency VPN service. While pcaps are only stored for short periods of time, netflow files generated by Zeek are kept longer for re-analysis. Zeek generates, among others, the following files:

- *conn.log* - It contains details of TCP/UDP/ICMP connections like the timestamp, IPs, and packet sizes. The complete description is presented in Table 5.1.
- *dns.log* - It contains all DNS requests along with the responses. The complete description is presented in Table 5.2.
- *http.log* - Provides details of HTTP communication. The log contains full URLs as well as selected headers. The complete description is presented in Table 5.3.

In addition to the network flows, Civilsphere provided reports created for clients who requested the Emergency VPN service. Each report contains details of vulnerabilities found during a single session and has three main parts: Summary, Packet Capture Information, and Detailed Findings. We present a sample report in Figure 5.1. The summary contains a short overview of the

Attribute	Description	Example
ts	UNIX timestamp of the first packet.	1550691501.573953
uid	Unique identifier of the connection.	CLldPr38pSrjjAI4sa
id	An object containing the connection's 4-tuple of endpoint addresses/ports.	(object [23])
proto	Transport layer protocol.	tcp
service	Application layer protocol.	http
duration	Duration of the connection, excluding the final ACK.	0.02
orig_bytes	Number of payload bytes from the originator.	42
resp_bytes	Number of payload bytes from the responder.	42
conn_state	State of the connection.	RSTO
local_orig	Whether the connection originated locally.	F
local_resp	Whether the connection is responded to locally.	F
missed_bytes	Number of bytes missed in content gaps, i.e., packet loss.	0
history	State history of the connections.	ShADadR
orig_pkts	Number of packets from the originator.	10
orig_ip_bytes	Number of packets from the originator.	10
resp_pkts	Number of packets from the responder.	42
resp_ip_bytes	Number of packets from the responder.	42
tunnel_parents	UID of encapsulating connections, only when tunneled.	(empty)

Table 5.1: Description of fields in conn.log generated by Zeek. Adapted from [24]

Attribute	Description	Example
ts	UNIX timestamp of the first packet.	1550691501.573953
uid	Unique identifier of the connection.	CLldPr38pSrjjAI4sa
id	An object containing the connection's 4-tuple of endpoint addresses/ports.	(object [23])
proto	Transport layer protocol.	tcp
trans_id	ID assigned by the program that generated the DNS query.	5537
rtt	Round trip time for query and response.	0.001029
query	Requested domain name.	google.com
qclass	Class of the query	1
qclass_name	Name of the query class.	C_INTERNET
qtype	Type of the query.	1
qtype_name	Name of of the query type.	A
rcode	Response code.	0
rcode_name	Name of the response code.	NOERROR
AA	Is the answer from the authoritative server?	F
TC	Was the message truncated?	F
RD	Does the client desires recursive service?	T
RA	Does the server support recursive queries?	T
Z	Reserved	0
answers	The set of resource descriptions in the query answer.	-
TTLs	Caching interval of the answers	113.00
rejected	Was the query rejected by the server?	F

Table 5.2: Description of fields in dns.log generated by Zeek. Adapted from [24]

result of the analysis and informs the user about the discovered vulnerabilities. When relevant, Civilsphere provides recommendations to the user on how to improve their security posture. Each vulnerability consists of its severity, name, and a short description. The details of the vulnerability are expanded in the Detailed Findings section, where the user is provided with specific data leaks and information relevant to the finding.

Attribute	Description	Example
ts	UNIX timestamp of the first packet.	1550691501.573953
uid	Unique identifier of the connection.	CLldPr38pSrjjAI4sa
id	An object containing the connection's 4-tuple of endpoint addresses/ports.	(object [23])
trans_depth	Pipelined depth of the transaction.	1
method	HTTP request method.	GET
host	Value of the HOST header.	google.com
uri	URI used in the request.	/?q=puf
referrer	Value of the REFERER header.	google.com
version	HTTP version.	1.1
user_agent	Value of the User-Agent header	iPhone6,1/12.1.4
origin	Value of the Origin header	google.com
request_body_len	Uncompressed content size transferred from the client.	69
response_body_len	Uncompressed content size transferred from the server.	42
status_code	Status code returned by the server.	404
status_msg	Status message returned by the server.	NOT FOUND
info_code	Last seen 1xx informational reply code returned by the server.	100
indo_msg	Last seen 1xx informational reply message returned by the server.	CONTINUE
tags	Indicators of various attributes related to the transaction.	-
username	Username if basic-auth is performed for the request.	Poro
password	Password if basic-auth is performed for the request.	abc123
capture_password	Was the password captured?	T
proxied	Headers that may indicate the request was proxied.	-
range_requests	Can the request assume 206 partial content in response?	F
orig_fuids	File unique IDs from the client.	FROfaf01y3Fu9x19e
orig_filenames	Filenames from the client.	file.xml
orig_mime_types	Mime types from the client.	application/xml
resp_fuids	File unique IDs from the server.	FROdof01y3Fu9x19e
resp_filenames	Filenames from the server.	file.xml
resp_mime_types	File unique IDs from the server.	application/ocsp-response

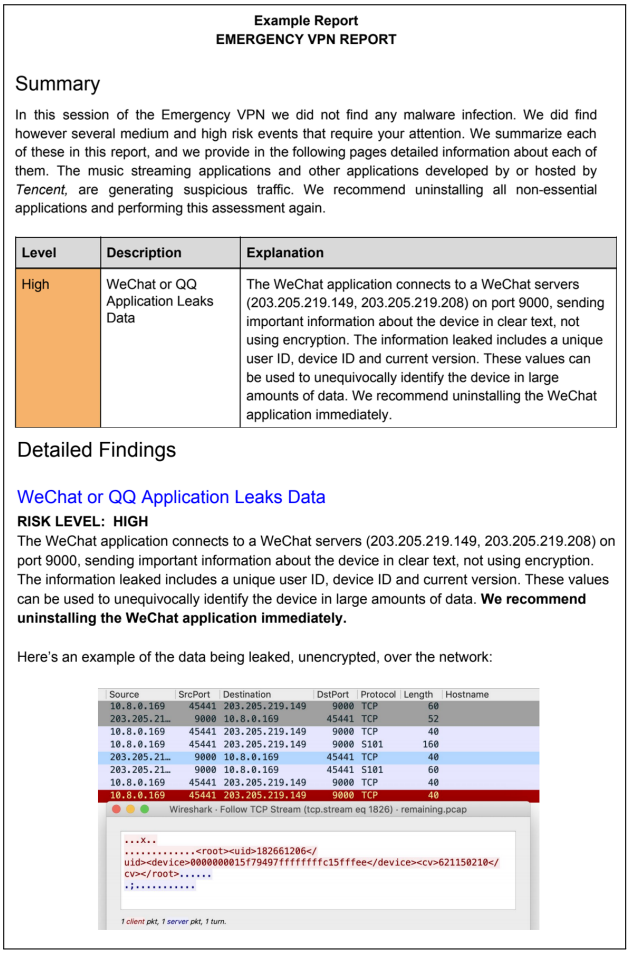
Table 5.3: Description of fields in http.log generated by Zeek. Adapted from [24]

5.1.2 Dataset creation

Since the information in the raw data is vast and split up among many files, we needed to create the dataset ourselves. We set on using a MySQL database to structure and compile the data. First, we devised a database model. We designed the database tables in a way that allows us to query any feature combination we might need quickly. Second, we processed Zeek netflow files and imported them to our database. For easy data extraction from Zeek files, we created, a broad set of Python scripts that can parse the files into our desired format. The scripts also split the data into CSV files, which are in 1:1 relationship to the database tables. Then we were able to import the resulting CSV files into the MySQL database as well as to an Excel spreadsheet. Having the data in the spreadsheet allowed us to look up and compare simple data quickly, without the need to create SQL queries.

The main script uses a module called `bro-log-parser`[1], which transforms the Zeek files into a Python dictionary. Having the data in the form of a dictionary allows us quick and easy access to all fields present in the files. Each network capture (profile) resides in its separate folder. The script reads the names of the profiles from a CSV file and then determines the full path

[1]The bro-log-parser Python module was created by Fabian Weisshaar and is available at `https://github.com/elnappo/bro-log-parser` under MIT license.



Example Report
EMERGENCY VPN REPORT

Summary

In this session of the Emergency VPN we did not find any malware infection. We did find however several medium and high risk events that require your attention. We summarize each of these in this report, and we provide in the following pages detailed information about each of them. The music streaming applications and other applications developed by or hosted by *Tencent,* are generating suspicious traffic. We recommend uninstalling all non-essential applications and performing this assessment again.

Level	Description	Explanation
High	WeChat or QQ Application Leaks Data	The WeChat application connects to a WeChat servers (203.205.219.149, 203.205.219.208) on port 9000, sending important information about the device in clear text, not using encryption. The information leaked includes a unique user ID, device ID and current version. These values can be used to unequivocally identify the device in large amounts of data. We recommend uninstalling the WeChat application immediately.

Detailed Findings

WeChat or QQ Application Leaks Data

RISK LEVEL: HIGH

The WeChat application connects to a WeChat servers (203.205.219.149, 203.205.219.208) on port 9000, sending important information about the device in clear text, not using encryption. The information leaked includes a unique user ID, device ID and current version. These values can be used to unequivocally identify the device in large amounts of data. **We recommend uninstalling the WeChat application immediately.**

Here's an example of the data being leaked, unencrypted, over the network:

Source	SrcPort	Destination	DstPort	Protocol	Length	Hostname
10.8.0.169	45441	203.205.219.149	9000	TCP	60	
203.205.21…	9000	10.8.0.169	45441	TCP	52	
10.8.0.169	45441	203.205.219.149	9000	TCP	40	
10.8.0.169	45441	203.205.219.149	9000	S101	160	
203.205.21…	9000	10.8.0.169	45441	TCP	40	
203.205.21…	9000	10.8.0.169	45441	S101	60	
10.8.0.169	45441	203.205.219.149	9000	TCP	40	
10.8.0.169	45441	203.205.219.149	9000	TCP	40	

Wireshark · Follow TCP Stream (tcp.stream eq 1826) · remaining.pcap

...x..
.....<root><uid>182661206</
uid><device>0000000015f79497ffffffffc15fffee</device><cv>621150210</
cv></root>.....
.;.....

1 client pkt, 1 server pkt, 1 turn.

Figure 5.1: Sample of an Emergency VPN report generated by human analysts at the Civilsphere Project.

where they are stored. We log all missing profiles for future reference. After that, we can start extracting data from the files.

Concessions had to be made when preparing the data. The conn.log contains all network flows, one by one, which amounts to millions of entries. Storing and subsequently querying such big data is not practical, fast or necessary. Therefore, we decided to merge the connections with the same Destination IP - Destination port pair. There is no need to track the source IP and port because the logs contain data from only one device. Therefore, the source IP is always the same, and the source port is random and not relevant for this analysis. For the merged connections, we summarized the total number of sent and received bytes. We also summed up the total duration of the connections. We made an exception to the rule for HTTP requests and responses, for each of which we stored interesting details available in the http.log file.

After compiling the raw data, the next step was to augment it. To understand the used devices more, we looked up the launch year and launch price of every known device model in the database. We also used a service provided by ipinfo.io, which allows extending specific details about IP addresses. Thanks to the service, we were able to look up owners and locations of each server associated with an IP address. Furthermore, we extensively expanded the data using VirusTotal API[2]. VirusTotal is a service for analyzing suspicious files, domains, and IP addresses. It uses a large number of antivirus programs to do the analysis. We submitted every accessed domain to the VirusTotal, obtaining the number of antivirus engines that found the domain malicious. We also compared the domains against a list of advertisements and trackers, described in the Section 5.3.2, to determine which domains belong to known advertisement servers or trackers. We used the VirusTotal service for IP addresses as well. For IPs, VirusTotal provides more information, compared to domains. We store the following:

- Positive downloaded samples - number of files downloaded from this IP with at least one positive antivirus result
- Negative downloaded samples - number of files downloaded from this IP with no positive antivirus results
- Positive communication samples - number of files communicating with the IP address with at least one positive antivirus result
- Negative communication samples - number of files communicating with the IP address with no positive antivirus results
- Positive referrers - URLs hosted on this IP address with at least one positive antivirus result
- Negative referrers - URLs hosted on this IP address with no positive antivirus result

A significant portion of the dataset is created from the Emergency VPN reports themselves. We analyzed every available report. We compiled all findings into a threat database and assigned them to their device. The elimination of duplicates allowed us to compare the vulnerabilities among the devices better. We rated each finding according to our standardized threat table. We also store the descriptions and details of the findings.

[2]VirusTotal is a free service aggregating output of different antivirus products, file and website characterization tools, website scanning engines and datasets, and user contributions. The service is available at `https://www.virustotal.com/`

5.1.3 Dataset Statistics

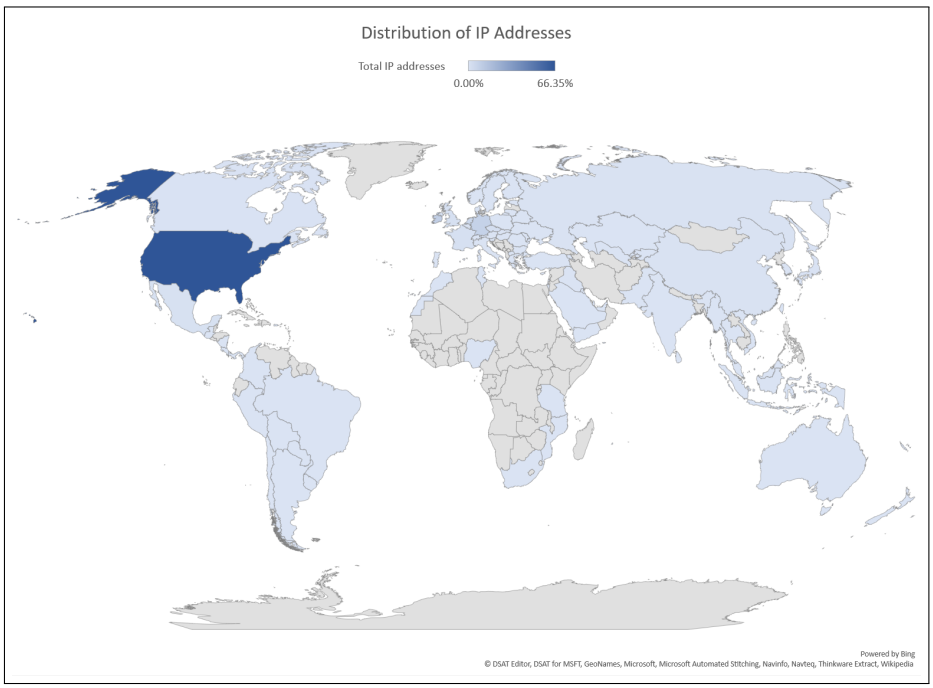


Figure 5.2: Distribution of locations of destination IP addresses accessed by devices in the dataset.

We based our dataset on Emergency VPN network captures from 65 devices. The users do not share their location with Civilsphere, but from the data leaks, we know some of them used the EVPN in Europe, Central America, and South America. It is important to note that the EVPN server is in the Czech Republic and hence Content Delivery Networks serve users the content local to the Czech Republic and not local to the user's location. We excluded the communication between client and EVPN server from the analysis.

The data contains 66.4 GB of traffic, with a median of 0.6 GB of data per device. The total capture time is 3,550.3 hours; therefore, each device contributed 54.6 hours of traffic on average. We also measured network connections. For purpouse of this thesis, we measure the network connection as a sum of network flows with the same triplet of IP, port, and device ID. The devices handled 48,419 network connections in total. The minimum number of connections per device is 55, and the maximum is 4,134.

The devices accessed 10,454 unique domains and 22,238 IP addresses. The vast majority of servers associated with the accessed IP addresses are based in the United States of America, which hosts 66% of the servers for a total of 14,731. Ireland follows the USA with 11%. The third most common server location is Germany, with a 5% share. The geodistribution of IP addresses is

displayed in Figure 5.2.

The most commonly used operating system in the dataset is Android with 39 devices, followed by 25 iOS devices. We could not determine the operating system for one device. Furthermore, most of the devices were manufactured by Apple (25), followed by Samsung (13), and Motorola (11). The full list of devices along with their details is included in Table A.1. According to our data, users have newer phones; the average age of the phones is two years. As we can see in Figure 5.3, many users either have cheap phones in a $200 range or high-end ones costing over $700. The average launch price of the smartphones is $410. We only track the launch price of the phones, the actual amount paid by civil society members can be different.

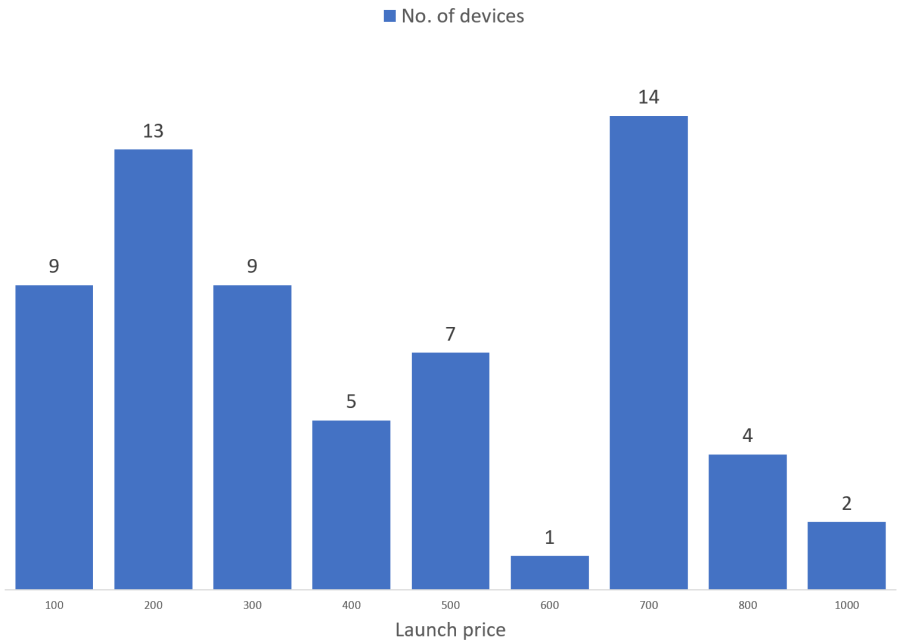


Figure 5.3: Launch price of phones in the dataset in USD. The price is rounded to hundreds.

We compiled and ranked the threats according to our standardized threat table described in Section 7. The reporters discovered 119 unique vulnerabilities. While there were several devices with zero vulnerabilities, the maximum number of reported threats was 14.

ID	IPs	Duration (h)	Received (MB)	Sent (MB)
2	943	73.6	2,652	552
3	1,197	77.6	1,290	249
4	360	56.0	315	31
5	876	81.6	1,326	70
6	896	73.8	1,387	852

Table 5.4 continued from previous page

ID	IPs	Duration (h)	Received (MB)	Sent (MB)
7	942	95.2	1,472	407
8	656	72.0	595	141
9	848	80.5	569	101
10	1,260	49.4	1,216	82
11	1,476	55.2	824	345
12	1,540	85.0	2,845	96
13	962	75.3	1,083	317
14	684	78.7	563	98
15	840	59.7	633	67
16	940	71.0	910	51
17	234	20.3	213	14
18	2,053	80.1	1,052	387
19	894	30.8	783	116
20	1,101	93.8	728	764
21	954	73.2	2,088	717
22	562	75.6	469	24
23	869	75.2	443	31
24	802	44.7	537	90
25	336	25.8	402	23
26	88	3.7	56	2
27	253	5.1	197	11
28	271	20.1	297	22
29	310	44.6	482	20
30	1,003	84.1	2,776	116
31	140	3.5	36	5
32	752	50.8	3,398	75
33	283	22.3	455	13
34	910	75.9	1,108	134
35	573	36.5	382	57
36	1,157	82.3	2,679	162
37	1,153	76.3	2,132	126
38	328	76.7	161	15
39	577	73.8	2,007	426
40	496	76.9	499	27
41	1,112	57.3	855	106
42	344	48.0	136	11
43	4,134	343.9	3,496	1,417
44	886	55.7	562	53
45	2,727	52.8	3,305	246
46	540	30.5	344	301
47	935	78.3	1,456	254
48	435	0.0	273	40
49	199	6.2	252	17
50	117	1.7	80	4

Table 5.4 continued from previous page

ID	IPs	Duration (h)	Received (MB)	Sent (MB)
51	73	51.9	16	2
52	55	0.4	107	5
53	313	17.2	54	49
54	73	51.9	16	2
55	464	51.0	330	25
56	349	28.1	182	54
57	268	4.8	27	4
58	126	2.0	10	1
59	55	1.7	8	1
60	623	73.0	1,175	120
61	920	66.8	990	94
62	925	51.8	1,162	148
63	92	15.6	90	3
64	352	47.3	230	20
65	1,685	69.9	188	71
66	97	5.6	54	3

Table 5.4: Overview of the profiles in our dataset, along with the total number of IP addresses, total duration of the captured traffic, and total received and sent data in Megabytes.

5.2 Feature Extraction

The network captures and Emergency VPN reports offer many different types of data. We needed to narrow down the broad dataset and extract only features useful for this thesis.

5.2.1 Packet Captures

A large number of available features in the raw data presented us with a challenge. We needed to select the elements of the network traffic relevant to our study. As we can see in Tables 5.1, 5.2, and 5.3, Zeek logs offer us many valuable traffic features already extracted from the raw packet captures.

As we mentioned before, we grouped the network flows according to the IP address and port of the remote server. These two values tell us with

what servers the device communicates. We also preserved the protocol, e.g., TCP, and service, e.g., DNS. Other essential features of a connection are the amount of transferred data and its duration. We decided that the added benefit of having the information for every network flow is not significant enough to outweigh the added complexity of data analysis. Therefore, we grouped the network flows with the same destination IP - destination port pair and summed up the `orig_ip_bytes` field, the `resp_ip_bytes field`, and the duration. The two fields describe the IP-level size of the connection, as seen on the wire. Their values are derived from the `total_length` header field. We chose to track the number of bytes as opposed to the number of packets. Due to the flexible size of a packet, they do not reflect the transferred data accurately. Another option for tracking the transferred data was using the fields `orig_bytes` and `resp_bytes`. They reflect the payload size, and in the case of TCP, they are derived from sequence numbers. [24] During our testing, we discovered a bug in Zeek, making values in these fields unreliable and frequently off by order of magnitudes.

We aimed to perform an analysis of the times when a device was used. We decided that hourly statistics provide enough granularity for the time data. Every device has 168 records in the database. Each entry is identified by a tuple of the day of the week and the hour. We counted the number of flows that occurred in the hour and calculated the total duration of these flows. Such division allows us to see not only how the habits change during the day but also how the day of the week affects the user's schedule.

We believe HTTP traffic deserves special description. Due to the lack of encryption, we could analyze complete URL as well as request and response headers. We decided to preserve the features that can help us learn more about the user. For every HTTP requests and response, we track:

- Source device
- Destination IP
- Destination port
- HTTP Version
- Host header
- Full URI
- Referer header
- User-Agent
- Status code

- Response MIME-type

The availability of unencrypted DNS requests gave us access to all the domains devices requested. Because of the time that passed between the network capture and our analysis, we needed to create a historic DNS resolver. We achieved the task by reading all DNS requests and responses and saving the domain-IP relationship in the database. The extracted data allows us to see where each domain was hosted at the time of the network capture.

5.2.2 Emergency VPN Reports

We attempted to extract as much useful information from the Emergency VPN reports as possible. We decided to process and store all reported findings into our database. The directly extracted features were the name, severity, and short description. For further features, we had to dig deeper into the details of the findings. A detailed description should accompany each of the findings. The description has no standard format and is usually a long section of text. We used the details to learn more about the vulnerability and the device. Our goal was to rate each finding according to our standardized threat table from Section 7. We had to extract every feature described in the threat table to be able to rate the finding accurately.

5.2.3 Overview of all the Extracted Features

We extracted 73 features in total from the reports and the network traffic files. The data describe devices, threats, connection, HTTP communication, and DNS requests. A complete overview of all extracted features is presented in Table 5.5.

5.3 External Datasets

The following text describes datasets, which we did not create, that are used to expand the information about the data collected from network captures.

Device	Threat	Connection	HTTP	IPs	DNS
Date of Analysis	SeverityLevel	IP	IP	IP	Domain
Identity	Name	Port	Port	Hostame	IP
Manufacturer	Description	Protocol	Method	Organization	VT Positives
Model	Application	Service	Version	Country	VT Link
OS	Application Owner	Total Bytes Sent	Host	Region	PiHole Positive
OS Version	Location	Total Bytes Received	URI	City	
Root	Name	Total Duration	Referrer	Location	
Pcap Size	Gender		User-Agent	VT Communicating Samples	
Duration	Age		Status Code	VT Referrers	
Phone Price	Marital Status		Response MIME	VT Downloaded Samples	
Phone Release	Device Model				
Time of Use	OS				
	OS Version				
	Installed Apps				
	Device Details				
	Mobile Carrier				
	IMEI/IMSI				
	VOIP Calls				
	P2P				
	China				
	IP Address				
	E-mail				
	User ID				
	Profiling				
	Search Queries				
	Visited Websites				
	Extended Usage				
	Tracking ID				
	Bank Info				

Table 5.5: Overview of every feature extracted from the data and used later for analysis.

5.3.1 Social Networks

In the data analysis we perform an analysis of social networks. To perform this analysis we needed a list of the most popular and active social networks. We used a list of 65 social networks and services [25].

The list contains the following social networks: Badoo, Baidu Tieba, Buzznet, CafeMom, Care2, Cellufun, Classmates, Delicious, DeviantArt, Douban, Facebook, Flickr, Flixster, Foursquare, Friendster, Funny or Die, Gaia Online, Google+, Instagram, Kiwibox, LINE, LinkedIn, LiveJournal, MeetMe, Meetup, Mixi, MyHeritage, Myspace, Nextdoor, Pinterest, QQ, Quora, QZone, Ravelry, Reddit, Renren, ReverbNation, Sina Weibo, Skype, Skyrock, Snapchat, Snapfish, Spreely, StumbleUpon, Tagged, Taringa, Telegram, The Dots, TikTok, Tout, Tumblr, Twitter, Vero, Viadeo, Viber, Vine, VKontakte, Wayn, We Heart It, WeChat, WhatsApp, Xanga, Xing, YouTube, YY.

5.3.2 Trackers and Ads

Ads and trackers often reduce the privacy of a user. We used a list intended for use in PiHole [26], which is a DNS-based ad blocker. The Reddit user *sjghvr* maintains the list on the site `dbl.oisd.nl`. According to the author, the domains included in the list belong to the following categories: Ads, Mobile Ads, Phishing, Malvertising, Malware, Spyware, Ransomware, CryptoJackin, Telemetry, and Analytics and Tracking. It contains 1,186,632 domains.

Chapter 6

Analysis

6.1 Data Processing

We stored our dataset in a relational database. To communicate and query the data, we wrote a script that can interface with the MySQL database. There are drivers (or connectors) for every major programming language. We decided to create the script in Python 3, as it allows fast prototyping and testing of new ideas. Another reason is our having significant experience with the language.

MySQL uses SQL language for database queries. Therefore, we can use SQL to join data from multiple tables and extract only the data we need for a particular use case. As an example, we present the following SQL query:

```
SELECT DISTINCT
    android.SubdomainName AS AndroidSubdomain
FROM
    (SELECT DISTINCT
        SubdomainName
    FROM
        thesisdata.dnsqueries
    LEFT JOIN deviceinfo
    ON dnsqueries.DeviceID = deviceinfo.DeviceID
    WHERE
        OS = 'Android') AS android
LEFT JOIN
    (SELECT DISTINCT
        SubdomainName
    FROM
        thesisdata.dnsqueries
    LEFT JOIN deviceinfo
    ON dnsqueries.DeviceID = deviceinfo.DeviceID
    WHERE
        OS = 'iOS') AS ios
    ON android.SubdomainName = ios.SubdomainName
WHERE
    ios.SubdomainName IS NULL;
```

Listing 6.1: Example of an SQL query

The query retrieves domains that were accessed only by Android devices. Any domain accessed by at least one iOS device will not be included in the list.

As mentioned above, we used Python to further process the results from SQL queries. We utilized two methods of outputting the data. The first one is text in Markdown format, which we use to display tables and lists. The other method is graphing using a library Plotly.

6.2 Domains

The devices accessed 10,454 unique domains. One question we asked was: What can we determine about a device just from the accessed websites? For this analysis, we used DNS queries made by each device.

VirusTotal analysis of the domains revealed that the vast majority of the visited domains is safe. However, about 4% of them were flagged as positive by one or more antivirus engines. Domain `api.ipify.org` was the most often flagged domain in our dataset. Upon closer inspection, we discovered the service itself is not malicious but is often used by malware to discover what is the public IP of the compromised device.

We discovered it is possible to guess if a device is an Apple iOS or Android device only by knowing the requested domains. There were 22 domains accessed on more than 90% of iOS devices and 5 domains present on all of them. In the case of Android, we discovered only one domain, which crossed the 90% threshold. All android devices accessed the domain android.clients.google.com. We expected the lower number of Android domains due to the diversity of Android ecosystem. The complete lists of unique domains for operating systems as well as brands are available in the thesis repository. [6]

We attempted to determine the manufacturer of an Android device by analysing the requested domains. We expected the source of the DNS queries to be from applications pre-installed on a device by its manufacturer. We present the results for three most common brands in our dataset in Table 6.1 The domains in the table are the most commonly accessed from that brand and are not present on any device of a different brand of device. We found that Xiaomi mobile devices disclose the manufacturer more evidently than other brands, followed by Motorola and Samsung. We did not have enough data to find unique domains that would identify the other brands.

Lastly, we asked a question: How unique is a device? To answer the question, we plotted the fraction of DNS requests unique to a device. In other words, how much of the DNS traffic was not seen in any other device. As

Samsung	Motorola	Xiaomi
urlappcloud.mcafee.com	urlmoto-cds.appspot.com	urlapi.ad.intl.xiaomi.com
urldc.di.atlas.samsung.com	urlargo.svcmot.com	urlglobal.market.xiaomi.com
urlservice.game-mode.net		urlsdkconfig.ad.intl.xiaomi.com
		urltracking.intl.miui.com

Table 6.1: Domains accessed only by Samsung, Motorola, and Xiaomi devices.

we can see in Figure 6.1, the average fraction of DNS requests unique to a device is 30% and fluctuates by a significant amount. The high number of unique requests might be affected by the size of our dataset. This graphs also includes the DNS request for random, non-existent domains, generated by applications such a Google Chrome browser to check for connectivity or DNS interception.

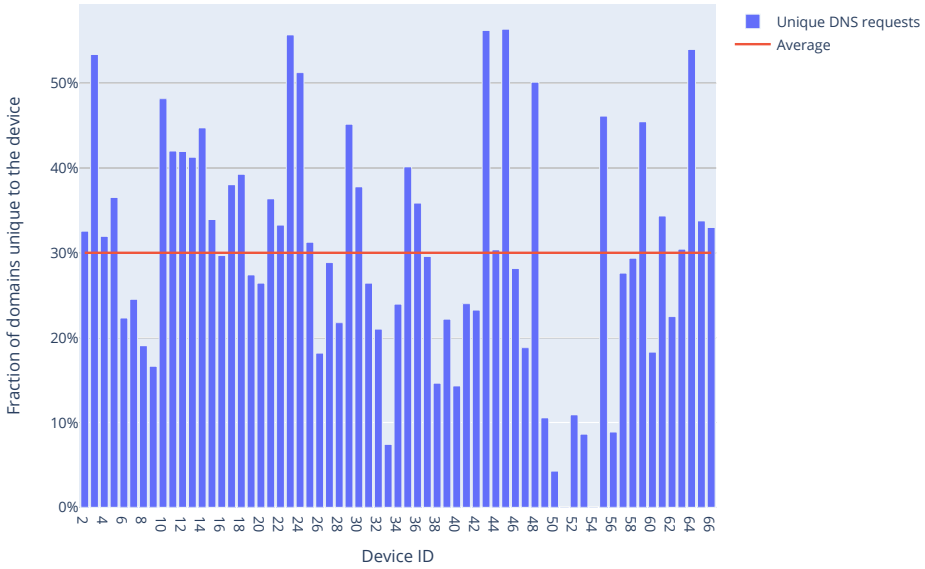


Figure 6.1: Percentage of domains unique to a device in all domains accessed by the device.

6.3 Social Networks

The goal for this thesis is to assess what social networks civil society uses. To determine whether the device accessed a social network, we used the list of 65 social networks described in the Section 5.3.1. We then looked for strings in

DNS requests matching the social networks' names. While we cannot be sure whether a user actively uses a social network or whether he only accessed a page associated with it, we can acknowledge the error exists.

We present the histogram showing the presence of social networks in Figure 6.2. Unsurprisingly, a substantial share of devices uses Facebook, Twitter, WhatsApp, and Youtube. Messaging applications offering end-to-end encryption, including WhatsApp and Line, are also popular among civil society members.

To further expand the analysis of the presence of social networks, we examined the number of social networks on each device in Figure 6.3. We can see that while every device uses about seven social networks on average, the presence varies largely. The two outliers are using 14 different social networks, and two devices use only one.

6.4 HTTP

An attacker with access to its victim network traffic can read and potentially intercept any traffic transported by HTTP. It is widely agreed upon that the websites and services should use encrypted HTTPS to protect its users, instead of plain-text HTTP. We were interested in how much of the traffic used the HTTP protocol. Figure 6.4 shows the fraction of web traffic transported on port 80/TCP, commonly used for HTTP, compared to port 443/TCP, which servers generally use for HTTPS traffic. We can see that almost all devices received more HTTP traffic than sent. The results for some devices may raise significant concerns. For example, device number 26 received less than 10% and sent less than 40% of web traffic using port 443/TCP. Nine devices out of the 65 received more than 30% of web traffic on port 80/TCP.

Users can rarely choose whether they allow their device to communicate with advertisement servers or trackers. Users cannot decide which advertisement servers to use either. For the identification of ads and trackers, we used a list for PiHole, described in the Section 5.3.2. We correlated the DNS queries from the devices with the PiHole list of ads and trackers to see which domains belong to those categories.

We depicted the relationship between the number of ads and the amount of HTTP traffic in Figure 6.5. Each point in the graph corresponds to a device. The Y axis shows what fraction of all DNS queries PiHole identified

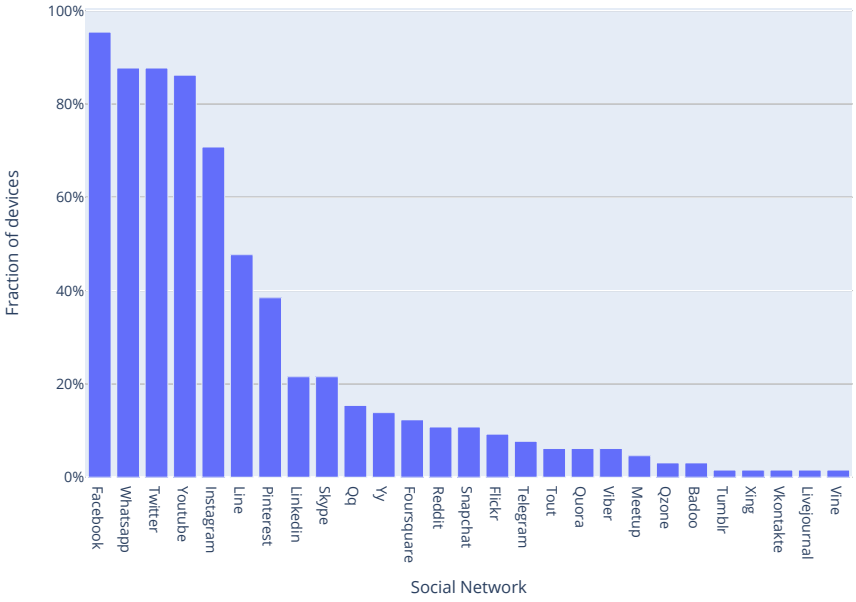


Figure 6.2: Histogram of social networks accessed by devices.

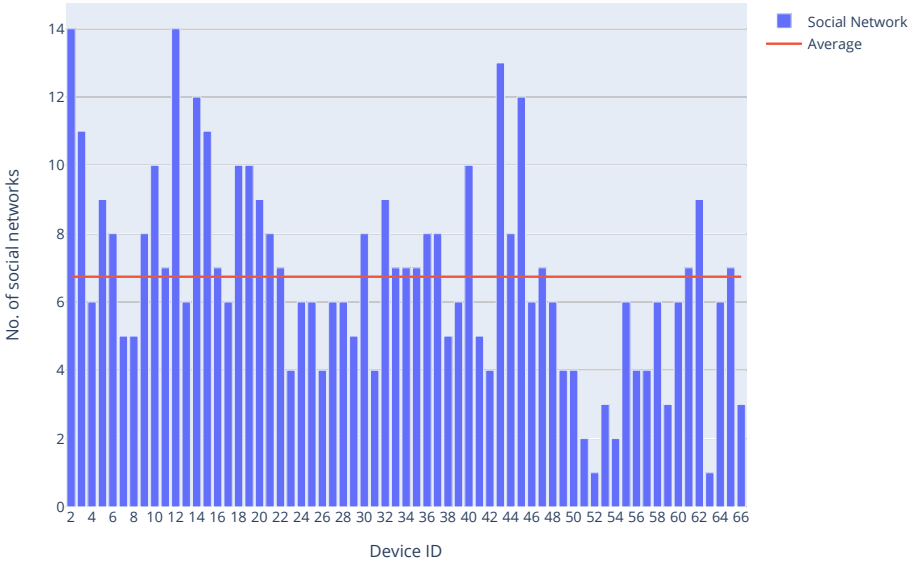


Figure 6.3: Access to social networks on each device.

as ads or trackers. The X axis displays what fraction of web traffic the device transferred on port 80, which is most of the time associated with HTTP traffic. The graph shows only the percentages to normalize the different amounts of data available among device. We can see that the distribution of points is reasonably uniform and that the amount of HTTP traffic does not significantly increase with more ads and trackers.

6.5 Location

Civilsphere does not ask for the location of users who request the network traffic analysis. Our analysis focused on finding out if it was possible to determine the country of origin of the user by looking at the traffic. From the available reports created by Civilsphere we knew the location of three devices, found by analysing data leaks on each traffic capture. We used these three cases to validate our analysis. We do not disclose the IDs of the devices for privacy reasons.

Firstly, we plotted the total duration of network connections in Figure 6.6. The graph displays the fraction of total network traffic duration occurring in each hour for three devices. There is noticeably less traffic between hours 6 and 13, followed by a massive spike in traffic at hour 14 and beyond. It is safe to assume the low traffic region is caused by users sleeping and the spike by users waking up. According to a study of sleeping habits, most of the world's population goes to bed between 12 AM and 1 AM and wakes up between 7 AM and 8 AM [27]. Hour 14 in the graph corresponds to 7 AM, therefore a reasonable assumption is that the user lives in Central Standard Time.

We created a heat map per device activity, showing all the countries associated with IPs that each device communicated with. To illustrate this concept, in Figure 6.7, we present a heat map for device 1. The map shows the difference between the average of the whole dataset and data for the device. We must be aware that the dataset used for creating averages is rather small and most likely is skewed towards countries that request the Emergency VPN services the most. The baseline for this analysis was presented in Figure 5.2 We can see that the device communicated more than average with IPs based in China, Mexico, Canada, and Ireland. In combination with the time chart described earlier, we can exclude China and Ireland as the device's home country as the Time Zone does not match. Therefore, based only on the two graphs, we can deduce the device is likely in North America.

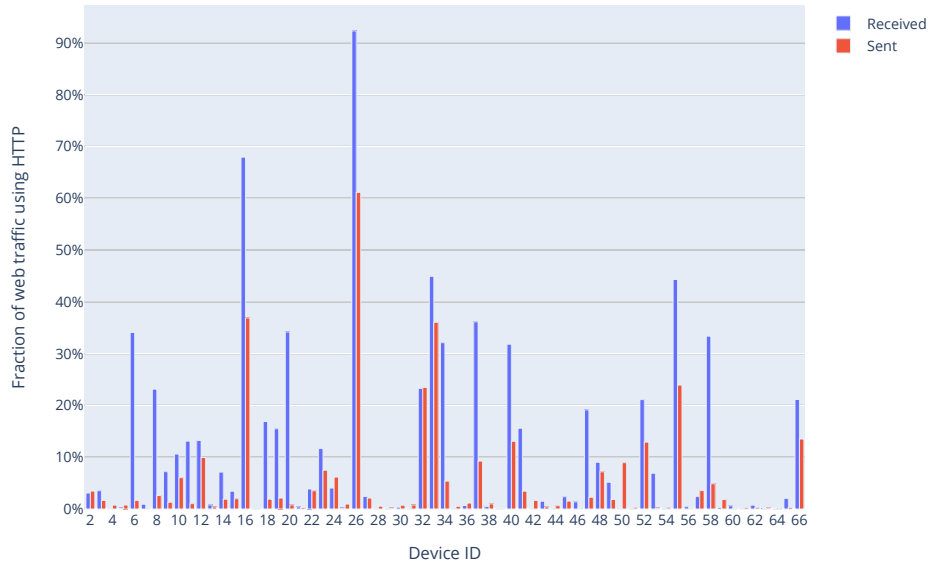


Figure 6.4: Fraction of web traffic transferred on port 80/TCP.

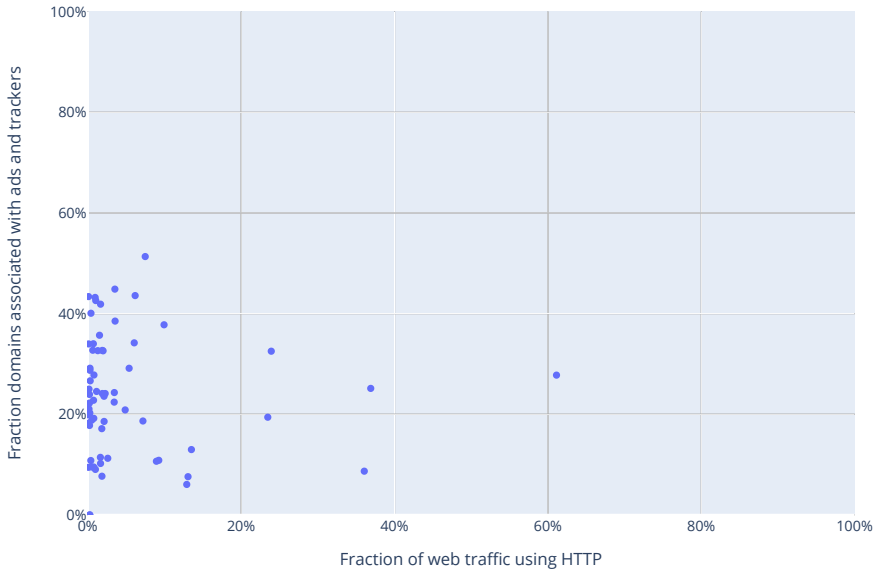


Figure 6.5: Relationship between number of ads and trackers and traffic on port 80/TCP for every device.

We conclude that through the analysis of network flows we could determine with certain accuracy the possible country where the device is located.

6.6 Threats

From the Emergency VPN reports created by Civilsphere analysts, we gathered the reported threats found and classified them according to our standardized threat table from Section 7. In Figure 6.8, we can see that the most common problem were the disclosure of the device model, operating system and its version. Worryingly, profiling, vulnerabilities which disclose user's likes or hobbies, and the vulnerabilities disclosing location were often present.

We also attempted to trace the source of the vulnerabilities. Most commonly, the device information leaked was traced to applications developed by mobile phone manufacturers, likely pre-installed on the device. The worst offender is iOS, which, in all cases, leaked the device model and OS version. The most common vulnerability was in the Spotify application, which uses HTTP to download album covers, therefore revealing the music a user likes.

The offenders leaking device's location were, in most cases, weather applications like AccuWeather. Often using HTTP, the applications included in the unencrypted requests either leak the exact GPS coordinates or the name of the city. We were able to determine the user's city for 15 devices, which is more than 20% of all devices. The exact GPS location was leaked on 13 devices, a 20% of all devices.

The most vulnerable and incomparably worst leaking application was Mercado Libre. The Argentinian application operating in Central and South America provides services for e-commerce and auctions. Its concept is similar to Ebay.com. The application uses HTTP only. It leaks, among other things, extensive device details, installed applications, username, and search history.

The relationship between the number of threats in a device and the number of ads and trackers is presented in Figure 6.9. The graph displays the fraction of all DNS requests to addresses associated with ads or trackers. Using only the fraction allows us to mitigate problems from having a different amount of data for each device. The fraction of ads and trackers increase up to three or four threats, and then it fluctuates around 25%.

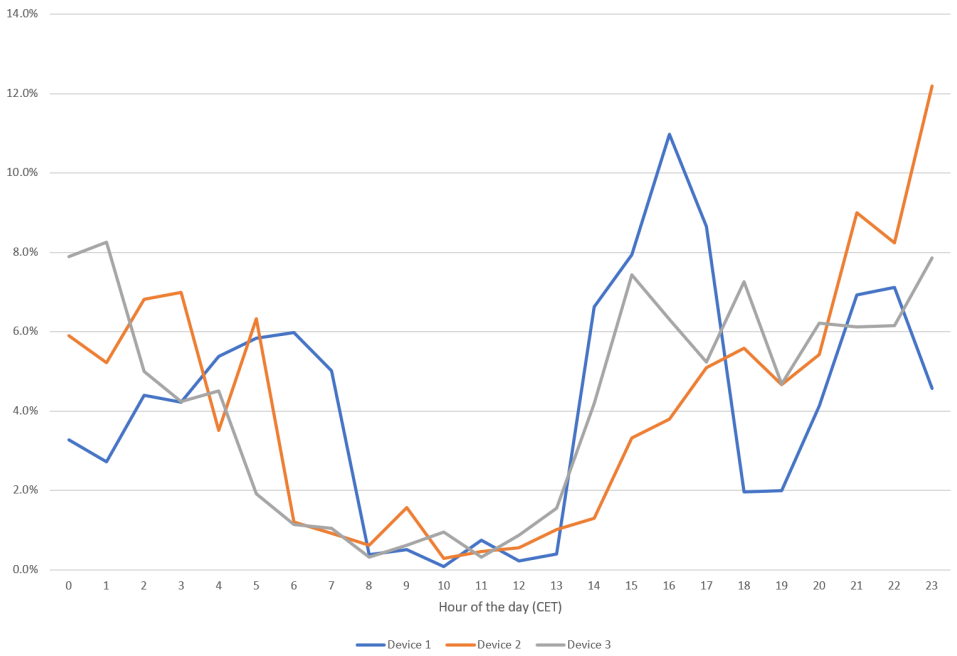


Figure 6.6: The fraction of total network traffic duration spent in hour.

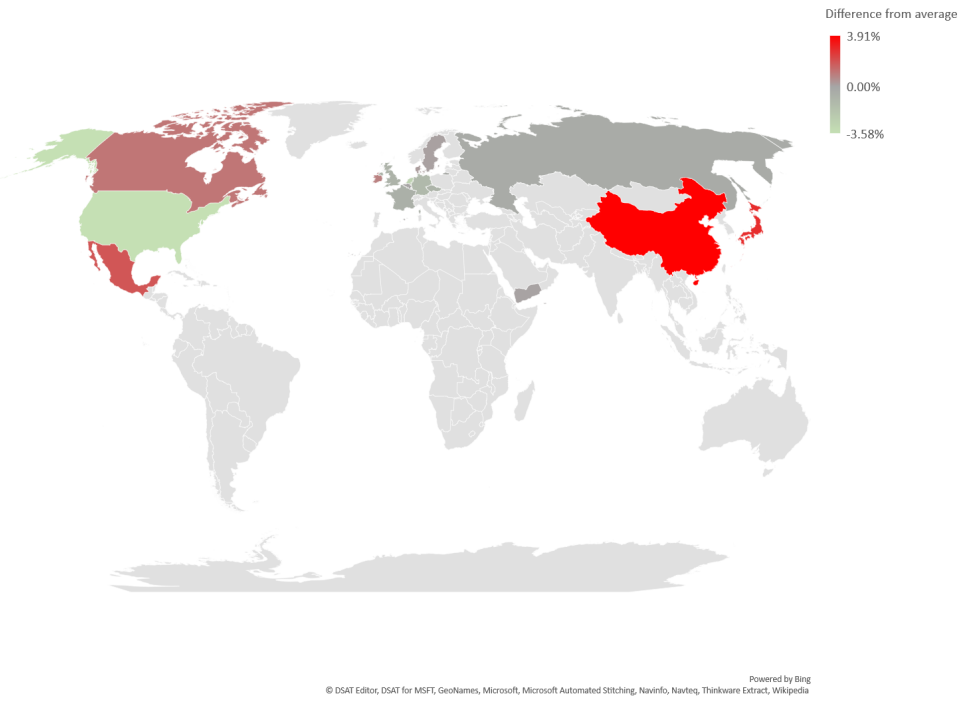


Figure 6.7: Countries associated with abnormal amount of traffic.

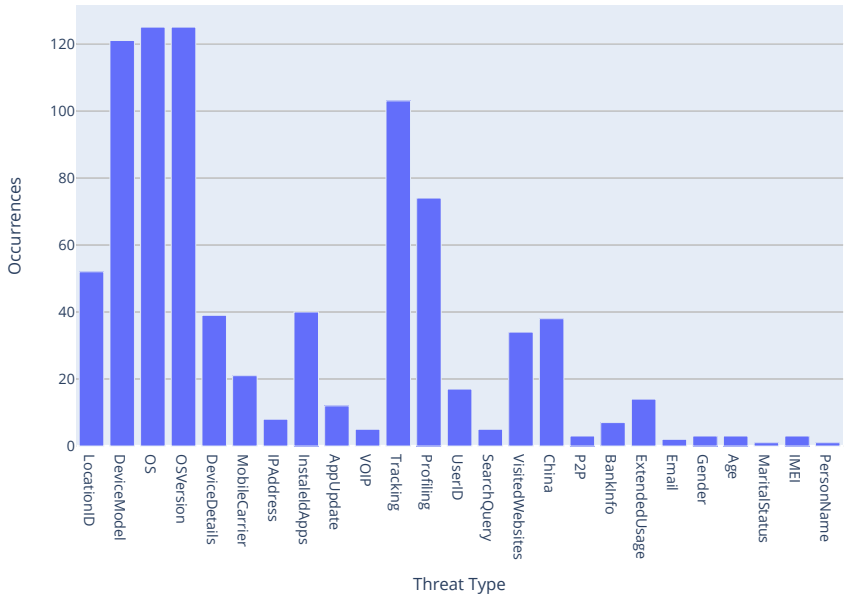


Figure 6.8: Number of reported types of threats.

On average, iOS devices were more secure. We identified 153 vulnerabilities on 39 Android devices, averaging 3.9 threats per device. In comparison, we found 74 vulnerabilities on 25 iOS devices, averaging 3.0 threats per device. The severity of the vulnerabilities was also lower on iOS on average, but the difference is almost negligible.

Defying our expectations, only a few civil society members used applications Signal or Telegram, which offer secure messaging. Services owned by internet giants like Google, Facebook, and Twitter were present on a vast majority of the analyzed devices. The data confirmed our suspicion that weather services often leak the precise location of the user. We discovered it is easy to identify the device's exact model and version of its operating system. Most of the analyzed devices did not contain threats, which would be high risk for the safety of its user. However, we found that a determined attacker with access to its victim traffic can triangulate their geographical location, profile their behavior, and use this information to craft a targeted attack, or harass the victim.

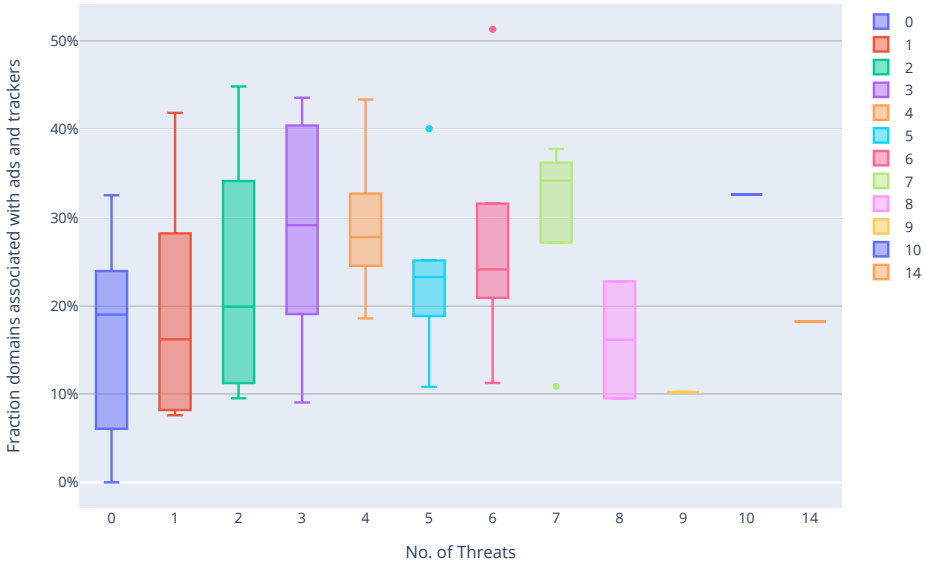


Figure 6.9: Relationship between the number of reported threats for each device and the number of ads and trackers.

Chapter 7

Standardized Threat Table

7.1 Threat Classification

There is a large number of different devices. If we include their software, one could argue that no two actively used devices are precisely the same. This uniqueness presents a challenge for finding and comparing the threats present on the devices. The existing metrics usually focus on the threat as a whole. Our goal is to protect user's privacy on the network; therefore we focus only on privacy related aspects of a threat, which allows us to examine closely only one category of threats to mobile devices - data leaks. Our project introduces the need for a way to compare the threats independently of the device, the operating system, or even the source application.

We developed our standardized threat table in two steps. First, we gathered all individual threats from reports sent to Emergency VPN users. In a report, a threat has an assigned severity (Info, Low, Medium, High, or Critical), name, a short description, and a detailed description, where we can see what data was leaked and the source application, if known.

Second, we observed similarities among the threats, based on the information they leak. We focused on features that affect a user's privacy the most. In the end, we narrowed the list down to the following categories:

- Location - device's position, can be one of the following: country, city, exact GPS coordinates
- Device Model - the exact model and manufacturer of the device, e.g., OnePlus 6T
- OS - operating system, i.e., Android, iOS
- OS Version - version of the operating system
- Device Details - information about configuration or hardware, not including model, e.g., screen size, CPU
- Mobile Carrier - e.g., Telcel, Vodafone
- IP Address - device's IP address was visible
- Installed Apps - information about installed applications, i.e., package name
- VOIP Calls - insecure VOIP communication
- Tracking ID - a unique string usable for tracking

- Profiling - information about user's habits and likes, e.g., podcasts, music, hotels
- User ID - login name or nickname
- Search Queries - searched products or items
- Visited Websites - details of the visited websites, e.g., specific news articles
- P2P - use of peer-to-peer connection
- Bank Information - details about the user's bank, e.g., its name
- Extended Usage - exact details about apps' usage, i.e., when and for how long an app was used
- IMEI/IMSI
  - International Mobile Equipment Identity (IMEI), a number unique to a device
  - International Mobile Subscriber Identity (IMSI), a number unique to a cellular network user
- Name - user's first or last name, or both
- E-mail
- Gender
- Age
- Marital Status

Most of the categories above are boolean type; therefore, we only track whether an application leaked that specific information or not. There are two exceptions to this rule. The first one is the "installed apps" category. Here we track if an application leaked only its presence or the presence of other applications as well. The second exception is the "location" category, as it is possibly the most sensitive information. The "location" category can have one of the five values:

- No Leak
- Country - only a user's state or country leaked, e.g., Czech Republic
- City - either city or city district leaked, e.g., Prague, Prague 6
- Indirect GPS - information that can narrow down a user's location to a small area, e.g., bus stops nearby, map tile

- Direct GPS - a user's exact GPS coordinates at a point in time
- Continuous GPS - application leaks the exact coordinates continuously, e.g., car navigation

Civilsphere will further use the standardized threat table to improve and to normalize the rating of threats among everyone creating the Emergency VPN reports. Our table can also lead to improved effectiveness since reporting common threats will not require the reporter to write the finding from scratch.

7.2 User Exposure and Risk Scores

Merely reporting the vulnerabilities found during an EVPN session might be confusing for less tech-savvy users. We decided to develop comprehensive scores to rate the exposure of user's information and the susceptibility to attacks. User Exposure Score allows us, and most importantly, the end-user, to determine the extent of the leaked information from just one number. Furthermore, User Risk Score expands the exposure data to provide users with information about their susceptibility to certain attacks.

The metrics are focused on the user's privacy. We attempted to view each type of information leak from the perspective of an attacker, who is determined to surveil, blackmail, or physically harass and intimidate its target. An example of the relationship can be a dictatorship hunting a whistleblower who went into hiding after leaking sensitive government documents.

7.2.1 Methodology

We grouped the metrics described in our standardized threat table into groups based on their uniting factors. For example, one group is Personal Information. In this group, we included all leaks which reveal information closely tied to the user. The information includes the user's name, gender, age, e-mail. All categories, their description, and the included metrics can be found in the Table 7.1

Category Name	Category Description	Metric
Location	Measures the extent to which the user's location can be determined from looking at the leaked data. An attacker can use the information to intercept the user in the real world.	Location
Personal Information	Information closely tied to the user's real-world identity and which can be used to describe the user when meeting in person.	Name Gender Age Marital Status
Device Details	Information about the device's hardware and software. An attacker can use the data to better target its attacks.	Device Model OS OS Version Installed Apps Device Details
Cellular Network	Information about the cellular network presence. A powerful actor (government) can use the information to gather more data about the user outside its internet network presence, e.g., call history, friends.	Mobile Carrier IMEI/IMSI
Risky Connections	The category includes connections which can put the user at risk. Connections that leak the persons in contact with the user. Connections routed through an untrusted entity that could capture and track the traffic.	VOIP Calls P2P IP Address
Internet Identity	User's internet persona. Information which the user uses to log in to services or to communicate with other people on the internet.	E-mail User ID
Behavior	Information about user's behavior and likes on and off the internet. The data can be used to determine, for example, music taste, hobbies, news sources, habits.	Profiling Search Queries Visited Websites Extended Usage Tracking
Finance	The leaked data which an attacker can use to track a user's finances or in bank related spear phishing attacks.	Bank Info

Table 7.1: Standardized Threat Table. Categories group multiple metrics that put the user at risk.

7.2.2 User Exposure Score

After categorizing the metrics, we proceed to create a method to calculate the overall user's exposure from all threats. User Exposure Score is meant to be a simple and easy to understand number which quantifies the extent of the leaked information. The final User Exposure Score is derived from the number of categories with at least one positive metric. The possible values of the User Exposure Score are Low, Medium, High, and Critical. Table 7.2 describes the conversion from the number of positive categories to the final score. Furthermore, the score can be expanded with an exposure table showing exactly which information types leaked.

Score	Positives
Low	0-2
Medium	3-4
High	5-6
Critical	7-8

Table 7.2: Conversion between the number of positive categories and the User Exposure Score.

The algorithm for obtaining the final score and the table is:

1. For each vulnerability.
   a. Fill out the Standardized Threat Table 7.1.
   b. For each category.
      (i) Calculate the number of positive metrics within the category.
      (ii) If the number of positives is more than zero, mark the category as positive.
2. For each final category.
   a. Mark the final category as positive if there is at least one positive category of its type in any threat.
3. Count the number of positive final categories.
4. Mark the positive final categories in the exposure table.

7.2.3 User Risk Score

Our next goal was to determine how the threats found make a user susceptible to the most common attacks. We decided on examining three groups of attacks:

- Social Engineering - techniques that use psychological manipulation to trick users into making security mistakes or giving away sensitive information.
  - Spear phishing
  - Scareware
  - Baiting
- Attacking - attacks which are targeted on user's device or account
  - Malware
  - Direct device hacks
  - Brute forcing credentials
- Surveillance - tracking of the user's behavior or movement online or even in the real world

After specifying the attack groups, we assigned a weight to each category. The sum of weight equals to 100. Impact of a threat can be different for every particular attack group. For example, a user's location does not have a great impact on the device attacks but a very large impact on the surveillance. The presented weight are temporary and based solely on the authors' experience in the cybersecurity field. We plan to re-evaluate the weights in the future after pooling more expert opinions. We present all temporary weights in the Table 7.3.

The value of a category is the portion of all possible leaked information. For example, the category Risky Connections contains three boolean metrics. We sum up all positive ones and divide the result by the total number of possible values, e.g., if there are two positives in four metrics, the category value is 0.5. To calculate the overall score, we use only the maximum category value across all identified vulnerabilities. We calculate the User Risk Score for an attack group as a linear combination:

$$S_{group} = W_L * V_L + W_P * V_P + W_D * V_D + W_C * V_C + W_R * V_R + W_I * V_I + W_B * V_B + W_F * V_F \quad (7.1)$$

Category	Social Engineering	Attacking	Surveillance
Location	10	3	40
Personal Information	20	9	15
Device Details	8	40	2
Cellular Network	10	5	10
Risky Connections	2	10	2
Internet Identity	10	20	8
Behavior	20	5	15
Finance	20	8	8

Table 7.3: Weights of the metrics in the attack groups.

where S_{group} is the User Risk Score for an attack group, W_X is the weight of the category X, V_X is the value of category X (X is the first letter of the category name).

The final User Risk Score is the maximum score among the attack groups:

$$S = max(S_{Social}, S_{Attacking}, S_{Surveillance}) \tag{7.2}$$

7.3 Example

This section shows how the User Exposure Score and the User Risk Score are calculated. We chose the device with ID 10 to serve as an example. The Emergency VPN report revealed seven vulnerabilities with severity ranging from low to high. The most significant contributor to data leaks was an application from the popular dating website OkCupid. The reported vulnerabilities and the full extent of leaked information are available in Table 7.4.

We start by calculating the User Exposure Score. For that, we need to

Severity	Description	Leaked data
High	Unencrypted requests to Google Maps	Live GPS location, profiling
High	Leakage of GPS location information by OkCupid app	Location, Device model, OS version, Carrier, Tracking, Profiling, Gender, Age, Marital status
High	Alibaba app P2P network and security concerns	P2P
Medium	Information leaked by Samsung services	Device model, OS version, Installed applications
Medium	Dangerous phishing sites visited	
Medium	WunderGround leaking location	City
Low	Suspicious non-encrypted connection to WhatsApp messaging server	

Table 7.4: Reported vulnerabilities for device 10

know what categories of data leaked. As we mentioned before, we store the vulnerabilities and the data they leak in a MySQL database. Therefore, we can write an SQL query that returns the leak status for each category. The query code is presented in Appendix A.2. As we can see in Table 7.5, device 10 leaks every category except Internet Identity and Finance, hence the final User Exposure Score is High.

The User Risk Score builds on the previous metric. First, we need to calculate how many data leak types leaked in each category and divide the result by the total possible value of that category. Now we use an SQL query from Appendix A.2. The category-wise results are again in Table 7.5. Then we use (7.1) to calculate the risk score for all three attack groups. Lastly, (7.2) signifies the final User Risk Score is the maximum score among the

Category	Leaked	Value
Location	YES	5/5
Personal Information	YES	3/4
Device Details	YES	4/5
Cellular Network	YES	1/2
Risky Connections	YES	1/3
Internet Identity	NO	0/2
Behavior	YES	2/5
Finance	NO	0/1
User Exposure Score	**High**	

Table 7.5: User Exposure Score and leaked data categories for device 10

attack groups. Therefore, the final User Risk Score for device 10 is 65/100. Results for the individual attack groups are in Table 7.6.

Attack Group	Score
Social Engineering	45/100
Attacking	50/100
Surveillance	65/100
User Risk Score	**65/100**

Table 7.6: User Risk Score and the individual attack group scores for device 10

Chapter 8

Conclusion

This thesis aimed to understand the current state of data privacy and security of mobile devices owned by the civil society members and standardize the classification of privacy-related mobile cybersecurity threats.

We used network captures and security reports produced from the Emergency VPN service provided by the Civilsphere Project at Czech Technical University. We extracted the relevant data from raw packet captures and compiled a list of reported threats from the security reports. Because the raw data cannot tell us the whole story, we augmented them with more details about each device and annotated the reported threats according to the standardized threat table we developed.

We learned that 60% of civil society members use Android phones with the rest using iOS devices. Users tend to have newer phones; an average owner has a phone released in the past two years. The average launch price of their smartphones is $410.

In all but one case, we succeeded in determining the exact model of the device — the same phones leaked versions of their respective operating systems as well. Our statistics indicate Android smartphones contain more vulnerabilities than iOS ones. Civilsphere researchers reported, on average, one more finding per device on Android than iOS. While the number of threats is different, the average severity level across all reports was similar.

We discovered it is possible to deduce the brand and OS of a device from DNS requests only. Every brand and manufacturer extends the clean version OS with its applications and analytics that communicate back with their servers. We found that operating systems have particularly specific patterns in network traffic. iOS devices heavily interact with iCloud and other services not available outside of the Apple ecosystem. In the case of Android, every device periodically performs connectivity checks. The smartphones keep querying a DNS server for random, non-existent domains. We created a list of domains specific for each of the operating systems and published the results in the Section A.1. iOS requests significantly more domains unique to its platform. We suspect this to be caused by the fragmented Android market.

A surprisingly high number of devices transferred a significant portion of data over HTTP. A third of the phones received more than 10% of web traffic through unencrypted HTTP connections. In one device, the unencrypted portion of the web traffic reached more than 90% for incoming and 60% for outgoing traffic. In combination with the number of reported threats, we discovered the increasing amount of HTTP traffic increases the number of reported vulnerabilities. The result confirms our expectations as the

researchers cannot analyze the content of encrypted network communication. We theorized the increase of HTTP traffic could be directly proportional to the number of ads and trackers. But, our statistics did not reveal any significant dependency.

Our analysis shows a worryingly high number of devices leak their location. We were able to determine at least a city-level position for nearly a quarter of devices and the exact GPS coordinates for more than a fifth. With enough data, it is possible to deduce at least a region or a country of the device. The use of regional services can also help determine with certain precision the general location of users. For example, a search engine `seznam.cz` is most likely used by Czech nationals.

Leaks of personally identifiable information are rare but still present. In two cases, a popular dating website OkCupid used an unencrypted connection to transfer user details. The data leaked included age, gender, and sexual preference of the user. In one instance, an application leaked the user's first and last name.

We developed a method for rating and describing privacy-related vulnerabilities in mobile devices. We analyzed the most commonly leaked kinds of sensitive information and created a set of metrics focusing on each type. We then combined the metrics into high-level groups based on their unifying characteristics. Thanks to these groups, we can present our Emergency VPNs clients with a comprehensive and standardized overview of the leaked data. To provide the most relevant results, we defined three areas of frequent cyber attacks — social engineering, attacking, and surveillance. We assigned weights to the parameters according to their importance in an attack group and used them to calculate the User Risk Score.

The Civilsphere project is planning on implementing the results of this thesis in future projects. Statistics and learning from our work will help to inform and train current and new specialized analysts. The standardized method for rating the reported vulnerabilities will ease the burden on researchers creating the reports from Emergency VPN captures, and provides a step forward in automating certain parts of the analysis and risk assessment. It can also result in normalized security levels across the assessed devices. The risk rating is flexible and can be tweaked in the future as new threats and security issues appear. We now have a better understanding of the types of data leaks and what an adversary can learn from its victim's network traffic.

Database models and scripts used in this thesis are published at [6].

Bibliography

1. ADAMS, D. *The Hitchhiker's Guide to the Galaxy*. Random House Publishing Group, 2007. Hitchhiker's Guide to the Galaxy. ISBN 9780307417138. Available also from: https://books.google.com/books?id=j24GMN0OtS8C.
2. O'DEA, S. *Smartphone users worldwide 2016-2021* [https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/]. 2020. Accessed on 2020-04-20.
3. SCOTT-RAILTON, John. Security for the high-risk user: separate and unequal. *IEEE Security & Privacy*. 2016, vol. 14, no. 2, pp. 79–87.
4. MARTIN, Kirsten; SHILTON, Katie. Putting mobile application privacy in context: An empirical study of user privacy expectations for mobile devices. *The Information Society*. 2016, vol. 32, no. 3, pp. 200–216.
5. REN, Jingjing; RAO, Ashwin; LINDORFER, Martina; LEGOUT, Arnaud; CHOFFNES, David. Recon: Revealing and controlling pii leaks in mobile network traffic. In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 2016, pp. 361–374.
6. ČECH, Jakub. *GitHub - MobileSecurityAnalysis* [https://github.com/stratosphereips/MobileSecurityAnalysis]. 2020.
7. CENTRE FOR CIVIL SOCIETY, LONDON SCHOOL OF ECONOMICS. *Report on Activities* [http://eprints.lse.ac.uk/29398/1/CCSReport05_06.pdf]. 2006. Accessed on 2020-04-20.
8. COLLINS ENGLISH DICTIONARY. *Civil society definition and meaning* [https://www.collinsdictionary.com/dictionary/english/civil-society]. 2020. Accessed on 2020-04-20.

9. CAROTHERS, Thomas; BARNDT, William. Civil society. *Foreign policy*. 1999, pp. 18–29.
10. GOUGH, J. W. *The Social Contract*. Oxford: Clarendon Press, 1936.
11. HOBBES, T. *Leviathan*. Andrew Ckooke at the Green Dragon, 1651. Available also from: `https://books.google.cz/books?id=3VddAAAAcAAJ`.
12. EUROPEAN UNION AGENCY FOR FUNDAMENTAL RIGHTS. *Challenges facing civil society organisations working on human rights in the EU* [`https://fra.europa.eu/sites/default/files/fra_uploads/fra-2018-challenges-facing-civil-society_en.pdf`]. 2018. Accessed on 2020-04-20.
13. FRONT LINE DEFENDERS. *Front Line Defenders Global Analysis* [`https://www.frontlinedefenders.org/sites/default/files/global_analysis_2019_web.pdf`]. 2020. Accessed on 2020-04-20.
14. BRITISH BROADCASTING CORPORATION. *Russia to label individuals as 'foreign agents' under new law* [`https://fra.europa.eu/sites/default/files/fra_uploads/fra-2018-challenges-facing-civil-society_en.pdf`]. 2019. Accessed on 2020-04-20.
15. INTERNATIONAL TELECOMMUNICATION UNION. *Facts and figures 2019* [`https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2019.pdf`]. 2019. Accessed on 2020-04-20.
16. GAMBA, Julien; RASHED, Mohammed; RAZAGHPANAH, Abbas; TAPIADOR, Juan; VALLINA-RODRIGUEZ, Narseo. An Analysis of Pre-installed Android Software. *arXiv preprint arXiv:1905.02713*. 2019.
17. GREENE, Tristan. *Comcast continues to inject its own code into websites you visit* [`https://thenextweb.com/insights/2017/12/11/comcast-continues-to-inject-its-own-code-into-websites-you-visit/`]. 2017. Accessed on 2020-04-20.
18. CHEN, Caleb. *Comcast still uses MITM javascript injection to serve unwanted ads and messages* [`https://www.privateinternetaccess.com/blog/comcast-still-uses-mitm-javascript-injection-serve-unwanted-ads-messages/`]. 2016. Accessed on 2020-04-20.
19. MEYER, David. *Nokia: Yes, we decrypt your HTTPS data, but don't worry about it* [`https://gigaom.com/2013/01/10/nokia-yes-we-decrypt-your-https-data-but-dont-worry-about-it/`]. 2013. Accessed on 2020-04-20.
20. FIRST.ORG, INC. *Common Vulnerability Scoring System version 3.1: Specification Document* [`https://www.first.org/cvss/specification-document`]. 2019. Accessed on 2020-04-20.
21. JAGALUR, Pavan K; LEVIN, Peter L; BRITTAIN, Kate; DUBINSKY, Matthew; LANDAU-JAGALUR, Kristine; LATHROP, Charlotte. Cybersecurity for Civil Society. In: *2018 IEEE International Symposium on Technology and Society (ISTAS)*. 2018, pp. 102–107.

22. LINDORFER, Martina; NEUGSCHWANDTNER, Matthias; WEICHSELBAUM, Lukas; FRATANTONIO, Yanick; VAN DER VEEN, Victor; PLATZER, Christian. Andrubis–1,000,000 apps later: A view on current Android malware behaviors. In: *2014 third international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*. 2014, pp. 3–17.

23. THE ZEEK PROJECT. *Zeek Manual - $conn_idrecord$* [`https://docs.zeek.org/en/current/scripts/base/init-bare.zeek.html#type-conn_id`]. 2019. Accessed on 2020-04-20.

24. THE ZEEK PROJECT. *Zeek Manual* [`https://docs.zeek.org/en/current/script-reference/log-files.html`]. 2019. Accessed on 2020-04-20.

25. SPENCER, Jamie. *65+ Social Networking Sites You Need to Know About* [`https://makeawebsitehub.com/social-media-sites/`]. 2019. Accessed on 2020-04-20.

26. SJHGVR. *abp.oisd.nl* $||Internet's\#1domainblocklist$ [`https://www.reddit.com/r/oisd_blocklist/comments/gjn972/abpoisdnl_internets_1_domain_blocklist/`]. 2020. Accessed on 2020-04-20.

27. SLEEP CYCLE. *Sleeping habits of the world revealed through Sleep Cycle app* [`https://www.dailymail.co.uk/sciencetech/article-3042230/Sleeping-habits-world-revealed-wakes-grumpy-China-best-quality-shut-eye-South-Africa-wakes-earliest.html`]. 2015. Accessed on 2020-04-20.

Appendix A

Attachments

A.1 Analyzed Devices

ID	Brand	Model	OS	Version	Price (USD)	Release Year
2	Asus	Zenfone 3	Android	8.0	90	2016
3	Xiaomi	Mi A1	Android	8.1	210	2017
4	Motorola	E5 Cruise	Android	8.0	100	2018
5	Asus	ZenFone 3 Max	Android	7.0	180	2016
6	Apple	iPhone 6	iOS	12.1.4	650	2014
7	Apple	iPhone 7	iOS	11.4.1	450	2016
8	Apple	iPhone 5s	iOS	12.1.4	650	2013
9	Apple	iPhone 8	iOS	12.1.2	700	2017
10	Samsung	Galaxy S7	Android	7.0	670	2016
11	OnePlus	6T	Android	9.0	550	2018
12	Apple	iPhone 7	iOS	12.1.4	450	2016
13	Motorola	E5 Plus	Android	8.0	170	2018
14	Xiaomi	Redmi 4X	Android	7.1.2	110	2017
15	Xiaomi	Redmi 6	Android	8.1	100	2018
16	Apple	iPhone 8	iOS	12.3	700	2017
17	Samsung	Galaxy J7 Prime	Android	8.1	260	2016
18	Apple	iPhone 7	iOS	12.1.2	450	2016
19	Apple	iPhone 7	iOS	12.2	450	2016
20	Apple	iPhone 6s Plus	iOS	12.3.1	750	2015
21	Motorola	Z3 Play	Android	9.0	350	2018
22	Xiaomi	Redmi 4X	Android	7.1.2	110	2017
23	Samsung	Galaxy J5	Android	5.1.1	210	2015
24	Samsung	Galaxy Tab S2	Android	7.0	400	2015
25	Huawei	Honor 7 Lite	Android	7.0	240	2016
26	Xiaomi	Redmi 4A	Android	6.0.1	120	2016
27	Huawei	P9 Lite	Android	7.0	270	2017
28	Samsung	Galaxy S7	Android	6.0.1	670	2016
29	Samsung	J2 Pro	Android	7.1.1	130	2018

ID	Brand	Model	OS	Version	Price (USD)	Release Year
30	Apple	iPhone 7	iOS	12.0	450	2016
31	Motorola	G5 Plus	Android	8.1	190	2016
32	Apple	iPhone 8	iOS	12.1	700	2017
33	Apple	iPhone 8	iOS	12.1	700	2017
34	Apple	iPhone 7	iOS	12.1	450	2016
35	Motorola	G5S Plus	Android	8.1	280	2017
36	Motorola	E5	Android	8.0	140	2018
37	Apple	iPhone 8	iOS	13.0	700	2017
38	Huawei	P9 Lite	Android	8.0.0	270	2017
39	Samsung	J5	Android	5.1.1	220	2015
40	Apple	iPhone 6s Plus	iOS	13.1.2	750	2015
41	Apple	iPhone 7	iOS	13.1.3	450	2016
42	LG	G Pro Lite	Android	4.4.2	230	2013
43	Motorola	G7 Power	Android	9	250	2019
44	Samsung	J6+	Android	9	220	2018
45	Samsung	Galaxy S9	Android	9	790	2018
46	Motorola	G7 Power	Android	9	250	2019
47	Apple	iPhone 8	iOS	12.1	700	2017
48	LG	K10	Android	7.0	200	2018
49	Apple	iPhone SE	iOS	13.1.3	400	2016
50	Sony	Xperia XA	Android	7.0	280	2016
51	Motorola	G4 Plus	Android	7.1.2	190	2016
52	Apple	iPhone X	iOS	11.4.1	1000	2017
53	Apple	iPhone 6s	iOS	11.4	650	2015
54	Motorola	G4 Plus	Android	7.1.2	190	2016
55	Xiaomi	Redmi 4A	Android	7.1.2	120	2016
56	Apple	iPhone 6 Plus	iOS	11.4.1	750	2014
57	Samsung	Galaxy Core Prime	Android	5.1.1	210	2014
58	Apple	iPhone SE	iOS	11.4.1	400	2016
59	Unknown	Unknown	Unknown	Unknown	N/A	N/A

ID	Brand	Model	OS	Version	Price (USD)	Release Year
60	Apple	iPhone 5s	iOS	11.2.5	650	2013
61	Apple	iPhone X	iOS	13.1.2	1000	2017
62	Apple	iPhone 8	iOS	13.1.3	700	2017
63	Motorola	G7	Android	9	250	2019
64	Samsung	Galaxy S8	Android	9	720	2017
65	Samsung	A7	Android	9	330	2018
66	Samsung	A10	Android	9	120	2019

Table A.1: Details of all devices in the dataset.

A.2 Code Samples

```
SELECT
    MAX(LocationID) AS 'Location',
    GREATEST(MAX(PersonName),
             MAX(Gender),
             MAX(Age),
             MAX(MaritalStatus)) AS 'Personal Information',
    GREATEST(MAX(DeviceModel),
             MAX(OS),
             MAX(OSVersion),
             MAX(DeviceDetails),
             MAX(InstaleldApps))
             AS 'Device Details',
    GREATEST(MAX(MobileCarrier), MAX(IMEI))
        AS 'Cellular Network',
    GREATEST(MAX(IPAddress), MAX(P2P), MAX(VOIP))
        AS 'Risky Connections',
    GREATEST(MAX(UserID), MAX(Email))
        AS 'Internet Identity',
    GREATEST(MAX(Tracking),
             MAX(Profiling),
             MAX(SearchQuery),
             MAX(VisitedWebsites),
             MAX(ExtendedUsage))
                 AS 'Behavior',
    MAX(BankInfo) AS 'Finance'
FROM
    deviceinfo_has_threats
        NATURAL JOIN
    threats
WHERE
    deviceid = 10;
```

Listing A.1: An SQL query which returns which categories of data leaked

```
SELECT
    MAX(LocationID) AS 'Location',
    (MAX(PersonName) + MAX(Gender) + MAX(Age)
        + MAX(MaritalStatus)) AS 'Personal Information',
    (MAX(DeviceModel) + MAX(OS) + MAX(OSVersion)
        + MAX(DeviceDetails) + MAX(InstaleldApps)) AS 'Device Details',
    (MAX(MobileCarrier) + MAX(IMEI)) AS 'Cellular Network',
    (MAX(IPAddress) + MAX(P2P) + MAX(VOIP)) AS 'Risky Connections',
    (MAX(UserID) + MAX(Email)) AS 'Internet Identity',
    (MAX(Tracking) + MAX(Profiling) + MAX(SearchQuery)
        + MAX(VisitedWebsites) + MAX(ExtendedUsage)) AS 'Behavior',
    MAX(BankInfo) AS 'Finance'
FROM
    deviceinfo_has_threats
        NATURAL JOIN
    threats
WHERE
    deviceid = 10;
```

Listing A.2: An SQL query which returns the total value of each data leak category

Appendix B

Acronyms

API Application Programming Interface. 34

CTU Czech Technical University in Prague. 2

CVSS Common Vulnerability Scoring System. ix, 14, 15

ECA Eastern Europe and Central Asia. 10

EVPN Emergency VPN. 4, 35, 58

FRA European Union Agency for Fundamental Rights. 8

GPS Global Positioning System. 2, 51, 56–58, 67

HRDs Human Rights Defenders. 9

HTTP HyperText Transfer Protocol. 24, 27, 30, 32, 33, 39, 47, 49, 51, 66, 67

HTTPS Hypertext Transfer Protocol Secure. 13, 47

IP Internet Protocol. ix, 26, 33–35, 38–40, 56, 59

ISP Internet Service Provider. 13

ITU International Telecommunication Union. 10, 11

LAN Local Area Network. 15

MENA the Middle East and North Africa. 10

MITM Man-in-the-Middle. 13, 14

OS Operating System. 12, 74–76

PII Personally Identifiable Information. 20

SSH Secure Shell. 15, 16

VOIP Voice over Internet Protocol. 56, 59

VT VirusTotal. 41