

Aalto University
School of Science
Master's Programme in Security and Cloud Computing

Jayshree Rathi

Mapping the Attack Surface of Telecommunication Networks from the Public Internet

Master's Thesis
Espoo, July 30, 2023

Supervisors: Prof. Tuomas Aura, Aalto University
Prof. Markus Hidell, KTH Royal Institute of Technology

Advisor: Siddharth Rao M.Sc. (Tech.), Nokia Bell Labs

Copyright © 2023 Jayshree Rathi

This work is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Author:	Jayshree Rathi	
Title:	Mapping the Attack Surface of Telecommunication Networks from the Public Internet	
Date:	July 30, 2023	Pages: 58
Major:	Security and Cloud Computing	Code: SCI3113
Supervisors:	Prof. Tuomas Aura Prof. Markus Hidell	
Advisor:	Siddharth Rao M.Sc. (Tech.), Nokia Bell Labs	
<p>The telecommunications sector is increasingly connected to the Internet, resulting in an expanded attack surface accessible from the public Internet. This has increased the availability of information such as IP addresses, open ports, and other network details that anyone from the Internet can access. As a result, potential entry points for attackers have increased, making it essential to map the attack surface of telecommunication networks from the public Internet. While previous research has explored various tools and techniques for mapping the attack surface of the Internet of Things (IoT) and Industrial Control Systems (ICS), such techniques have not yet been extended to the telecommunications domain.</p> <p>This thesis aims to comprehensively map the attack surface of telecommunications operators from the public Internet. To achieve this, we conducted a thorough literature review and proposed a methodology for mapping the attack surface explicitly designed for the telecommunications sector. First, we devised a research workflow that outlines the steps involved in the methodology. Second, we developed a Python-based tool to automate the workflow. We used the tool for a particular mobile network operator. It successfully gathered DNS records, IP addresses, exposed ports, services, Autonomous System Numbers (ASN), server versions, and potential vulnerabilities. The collected data provides valuable insights into the network infrastructure of the operator, aiding in the understanding of potential security risks.</p>		
Keywords:	attack surface mapping, telecommunication, passive reconnaissance, mobile network operator, public internet	
Language:	English	

Utfört av:	Jayshree Rathi		
Arbetets namn:	Kartläggning av Attackytan på Telekommunikationsnätverk från det Offentliga Internet		
Datum:	Juli 30, 2023	Sidantal:	58
Huvudämne:	Säkerhet och Cloud Computing	Kod:	SCI3113
Övervakare:	Prof. Tuomas Aura Prof. Markus Hidell		
Handledare:	Siddharth Rao M.Sc. (Tekn.), Nokia Bell Labs		
<p>Telekommunikationssektorn blir allt mer kopplad till Internet, detta resulterar i en större attackyta som är tillgänglig från det offentliga Internet. Detta har gett en ökad tillgänglighet av information som till exempel IP adresser, öppna portar, och annan nätverksinformation som vem som helst kan få åtkomst till via Internet. På grund av detta, har potentiella ingångar för attacker ökat, detta gör det avgörande att kartlägga attackytan för telekommunikationsnätverk från det offentliga Internet. Medan tidigare forskning har undersökt olika verktyg och tekniker för att kartlägga attackytor för the Internet of Things (IoT) och Industrial Control Systems (ICS), så har sådana tekniker ännu inte sträckt sig till telekommunikationsdomänen.</p> <p>Denna avhandling har som mål att utförligt kartlägga attackytan för telekommunikationsoperatörer från det offentliga Internet. För att uppnå detta, har vi utfört en grundlig litteraturgranskning och föreslagit en metodologi för kartläggning av attackytor specifikt designat för telekommunikationssektorn. Först konstruerade vi ett forskningsarbetsflöde som beskriver stegen involverade i metodologin. Sedan konstruerade vi ett Python-baserat verktyg för att automatisera arbetsflödet. Vi använde verktyget för en särskild mobilnätverksoperatör. Den samlade framgångsrikt in DNS uppgifter, IP adresser, exponerade portar, tjänster, Autonomous System Numbers (ASN), versioner av servrar, och potentiella sårbarheter. Den insamlade informationen ger värdefulla insikter i nätverksinfrastrukturen hos operatören, vilket hjälper till att förstå potentiella säkerhetsrisker</p>			
Nyckelord:	kartläggning av attackytan, telekommunikation, passiv rekonnaisance, mobilnätoperatör, offentligt internet		
Språk:	Engelska		

Acknowledgements

I would like to thank my supervisor **Prof. Tuomas Aura** from Aalto University, and **Prof. Markus Hidell** from KTH University for their feedback, guidance, and comments during this thesis project.

I would like to express my sincere gratitude to my advisor **Siddharth Rao**, for his continuous support and mentorship throughout this thesis project. I am incredibly grateful for the weekly meetings, brainstorming sessions, constant feedback on thesis drafts, and motivation throughout to keep me going. I would also like to thank my manager **Yoan Miche** and all my teammates at Nokia Bell Labs for their constant support and encouragement.

I would like to thank all my friends across the world for their constant check-ups and motivation, and **SECCLO 2021** batch for the most enjoyable two years.

A special thank you to my family, my grandparents, parents, and siblings, for their unconditional love. Big thank you for always believing in me.

Finally, my biggest thank you to **Sanyog** for all the late night and early morning coffee and for always being my rock during the most challenging times.

Espoo, July 30, 2023

Jayshree Rathi

With the support of the
Erasmus+ Programme
of the European Union



Abbreviations and Acronyms

API Application Programming Interface

AS Autonomous System

ASN Autonomous System Number

CNAME Canonical Name

CPE Common Platform Enumeration

CRM Customer Relationship Management

CVE Common Vulnerabilities and Exposures

CVSS Common Vulnerability Scoring System

DDoS Distributed Denial-of-Service

DNS Domain Name System

ENUM E.164 Number Mapping

ERP Enterprise Resource Planning

FQDN Fully Qualified Domain Name

GGSN Gateway GPRS Support Node

GRX GPRS Roaming eXchange

GSM Global System for Mobile Communications

GSMA Global System for Mobile communications Association

ICMP Internet Control Message Protocol

ICS Industrial Control Systems

IDE Integrated Development Environment
IDS Intrusion Detection Systems
IoT Internet of Things
IP Internet Protocol
IPS Intrusion Prevention Systems
IPX Internet Packet Exchange
ISDN Integrated Services Digital Network
ISP Internet Service Provider
LTE Long Term Evolution
MCC Mobile Country Code
MMSC Multimedia Messaging Center
MNC Mobile Network Code
MNO Mobile Network Operator
MX Mail Exchanger
NS Name Server
NVD National Vulnerability Database
OSINT Open Source Intelligence
SCADA Supervisory Control and Data Acquisition
SGSN Serving GPRS Support Node
SPF Sender Policy Framework
SSD Solid-State Drive
SSH Secure Shell
TLS Transport Layer Security

Contents

Abbreviations and Acronyms	6
List of Tables	10
List of Figures	11
1 Introduction	12
1.1 Problem Statement	13
1.2 Goals	13
1.3 Research Questions	13
1.4 Sustainability and Ethics	14
1.5 Research Methodology	14
1.6 Delimitations	14
1.7 Structure of the Thesis	15
2 Background	16
2.1 Attack Surface Mapping	16
2.2 Assets	17
2.2.1 Generic Enterprise Assets	17
2.2.2 Network Assets	18
2.3 Reconnaissance	20
2.3.1 Active Reconnaissance	20
2.3.2 Passive Reconnaissance	22
2.3.2.1 Search Engine Supported by Internet Scanning	23
2.3.2.2 Subdomain Enumeration	24
2.3.2.3 GSMA Documents	25
2.3.2.4 National Vulnerability Database	25
2.3.2.5 Other Tools	26
2.4 Literature Review	27
2.4.1 Mapping Devices on the Internet using Shodan	27
2.4.2 Using Shodan for Vulnerability Assessments	28

2.4.3	Aggregate Attack Surface Mapping	28
2.4.4	Asset Discovery Framework	29
3	Methodology	30
3.1	Research Workflow	30
3.2	Tool Implementation	33
3.2.1	DNS Reconnaissance Module	34
3.2.2	Banner Grabbing Module	35
3.2.3	Banner Analysis Module	36
3.3	Tool Usage	38
4	Evaluation and Results	43
4.1	Subdomains	43
4.2	DNS Records	44
4.3	Statistical Summary	44
5	Discussion	49
5.1	Limitations	49
5.2	Future Directions	50
6	Conclusions	53

List of Tables

2.1	Common DNS records	19
2.2	Network scanning tools	21
2.3	Vulnerability scanning tools	22
2.4	Ports scanned by Shodan	24
2.5	Subdomain enumeration tools	25
2.6	OSINT tools	26
3.1	Meta-data used from Shodan	32
3.2	Meta-data used from Censys	33
3.3	Banner and server information	39
4.1	Sample of gathered subdomains	44
4.2	Sample of DNS CNAME records gathered	45
4.3	Statistical summary of CVE count	46
4.4	Gathered Vulnerability details for Apache/2.4.54	48
5.1	List of telecommunications-specific ports and protocols	51

List of Figures

2.1	A sample banner grabbed from Shodan	27
3.1	Research workflow used for attack surface mapping.	31
3.2	Graphical representation of the tool’s architecture showcasing different modules, their interdependencies, and output data.	34
3.3	DNS record gathered from MXToolBox	36
3.4	DNS records stored for a telecommunications operator after processing	36
3.5	A banner grabbed from Shodan	37
3.6	A banner grabbed from Censys	37
3.7	Aggregate banner	41
3.8	Snapshot of the tool	42
4.1	Command for gathering subdomains	43
4.2	Command for gathering DNS records	44
4.3	Services distribution	46
4.4	Server types distribution	47

Chapter 1

Introduction

The telecommunications sector has become an indispensable part of modern society, connecting people across the globe and facilitating the flow of information. The purpose of telecommunication networks is to enable reliable, fast, and robust communication between individuals, businesses, and institutions on a global scale while ensuring the security and integrity of both the communication itself and the underlying network infrastructure. However, the criticality and importance of the telecommunications infrastructure have also made it an attractive target for cybercriminals [1].

Traditionally, telecommunication networks were designed to operate in closed networks that cannot be reached by the public Internet. Security of these telecommunication networks progressed in an enclosed setting, where protocol standardization often involved close collaboration among technology companies. The primary approach to network security was to limit access to a network of trusted associates rather than using robust cryptographic building blocks. However, in recent decades, telecommunication networks have adopted the use of IP-based protocols, which have enabled these networks to be connected to the Internet in both control and data layers. In addition, telecommunications operators are transitioning from operating all aspects of their networks in-house to outsourcing network functions to services and infrastructure provided by commercial vendors. Companies like Deutsche Telekom and Globe Telecom have migrated all their IT workloads to the public cloud, resulting in cost savings, efficiency improvements, and increased capacity [2]. In addition, telecommunications operators are building edge applications with the help of the public cloud, an example being SK Telecom's use of AWS's Wavelength.

This exposure of telecommunication networks to the public Internet increases their security footprints and makes them susceptible to misconfiguration. *Security footprints* are the public information available about a network or an organization that can be gathered by anyone. This publicly accessible information might in-

clude data about IP addresses, internal network architecture, open ports, version of the software and operating system used, exposed services, and DNS records. This opens up potential points of entry for attackers and increases the overall attack surface of telecommunication networks. However, the security perspective employed in the telecommunications sector has not kept pace with these changes and, in many respects, continues to adhere to outdated security models. Consequently, it is crucial to develop a fresh perspective on the mapping of attack surfaces within the telecommunications sector, with a focus on detecting the exposure of security footprints from the public Internet before potential attackers discover them.

1.1 Problem Statement

The necessity of mapping the attack surface of telecommunications operators has grown due to its potential exposure to the public Internet. While various tools and techniques have been employed to map the attack surface from the public Internet in the context of Industrial Control Systems (ICS) and Internet of Things (IoT), such research has not been extended to the telecommunications domain. This is mainly attributed to the initial design and the belief that telecommunication networks operate within a closed environment. However, this belief may no longer hold, necessitating the use of these techniques to identify the extent of a telecommunications operator's exposure to the Internet. This thesis aims to bridge this gap by employing multiple attack surface mapping techniques and tools to comprehensively map the network attack surface of a telecommunications operator from the public Internet.

1.2 Goals

The main goals of this thesis are:

- To establish a comprehensive methodology for mapping the attack surface of a telecommunications operator.
- To develop an automated tool for aggregating data related to the network attack surface of a telecommunications operator.

1.3 Research Questions

This thesis seeks to address the following research questions:

- Have attack surface mapping techniques been applied in the domain of telecommunications operators? Can existing tools and methods be adapted to assess the attack surface of telecommunications operators?
- What network assets are visible and accessible to the public Internet in the context of telecommunications operators?

1.4 Sustainability and Ethics

The primary focus of this research is to enhance the long-term security of telecommunications operators and their users, contributing to the sustainability of secure digital communications. From an ethical standpoint, active scanning methodologies were deliberately omitted throughout this thesis. No probes were sent directly, ensuring compliance with ethical guidelines and avoiding violations. It is important to emphasize that the proposed methodology and the data collected solely rely on publicly available sources on the Internet. Furthermore, no exploits of vulnerabilities were conducted during this study. This ethical approach ensures that the research maintains its integrity and credibility, prioritizing the security and confidentiality of telecommunications systems.

1.5 Research Methodology

Our thesis employed a qualitative approach to map the attack surface of the telecommunications operator. This involved conducting an extensive literature review and investigating various attack surface mapping tools and techniques used in this field. Through our qualitative research methods, we recognized the significance of leveraging a combination of multiple tools and search engines to ensure a comprehensive understanding of the attack surface. Building upon the findings from our qualitative research, we devised a research workflow and designed the architecture of our tool.

1.6 Delimitations

We have established certain delimitations to maintain a clear and well-defined scope for our research. First, our focus is specifically on the network attack surface within the telecommunications operator, which refers to the potential points of entry within the network infrastructure of the telecommunications operator. Additionally, it is important to note that our research is based solely on passive

techniques. We rely on information that is publicly accessible through the Internet and do not engage in any active scanning or probing activities. This suggests that our analysis is limited to the data and insights available through passive means, such as analyzing publicly available information and using existing tools and databases.

1.7 Structure of the Thesis

The remainder of the thesis is structured as follows. Chapter 2 provides the necessary background for the thesis. It also outlines the literature review. Chapter 3 presents the proposed methodology and steps taken to conduct our research. Additionally, it details the Python-based tool developed for this research. Chapter 4 presents and evaluates the results gathered from the tool. Chapter 5 discusses the limitations of our work and suggests ways for future directions. Finally, Chapter 6 concludes the thesis, summarizing the key findings.

Chapter 2

Background

In this chapter, we provide background information on attack surface mapping and discuss various tools and techniques used for it. In addition, we discuss enterprise assets, with a specific focus on network assets. Finally, we conduct a comprehensive review of past literature related to attack surface mapping on Internet-connected information systems using passive techniques.

2.1 Attack Surface Mapping

Attack surface mapping is a method used to identify and analyze a system's various components, interfaces, and dependencies. A system's attack surface is the set of ways it might be susceptible to an attack [3]. It encompasses entry points, exit points, channels, and untrusted data items that attackers can exploit. Mapping the attack surface helps assess the overall security status by determining the extent to which the system is susceptible to potential attacks. This includes identifying external points, such as network interfaces and APIs accessible from outside the organization's network, as well as internal components like databases and servers.

Attackers exploit the attack surface by identifying vulnerabilities within the system's resources and using them to gain unauthorized access and manipulate system behavior. By understanding the attack surface, attackers can focus on weakly protected entry points and insecure communication channels. On the other hand, defenders use the concept of attack surface mapping to enhance system security. They assess and analyze the attack surface to identify potential vulnerabilities and prioritize security efforts accordingly. By reducing the attack surface, the defenders aim to minimize the number of potential entry points and channels attackers can exploit. During software development, attack surface measurements are used to guide testing efforts and ensure that security patches are applied in a manner that does not expand the attack surface. Additionally, software consumers consider

attack surface measurements when selecting software configurations to minimize exposure to potential risks [3].

The initial step in the mapping of attack surface mapping involves asset identification. Many organizations struggle to maintain an accurate inventory of their assets, which can give attackers an advantage. Therefore, identifying and analyzing assets is crucial in understanding vulnerabilities and potential targets for attackers. The initial phase of collecting asset-related data is known as reconnaissance [4]. The reconnaissance phase also involves gathering information on vulnerabilities in the target network. The information gathered during the reconnaissance phase helps an attacker choose the most suitable tools to carry out a successful attack on the target while enabling the organization to implement proactive security measures and safeguard their infrastructure against potential vulnerabilities.

2.2 Assets

Before an organization can secure itself, it is imperative to understand the assets that make up its IT infrastructure. Assets are valuable items that require protection. Standardization frameworks such as ISO/IEC and NIST framework define assets as “any entity or item that is of value to an organization” [5, 6]. The NIST framework considers assets as both tangible including physical items like hardware and network devices and intangible, consisting of software, functions, data, and humans. The ISO/IEC definition expands to incorporate soft assets such as employee-related information (e.g., job roles and contact details), as well as business and financial data.

2.2.1 Generic Enterprise Assets

All the assets a business possesses that are used in its daily operations to generate revenue and value constitute its enterprise assets [7]. These assets are essential to the operation and continuity of the business and include tangible and intangible assets such as hardware, software, networks, and information assets.

- Hardware assets refer to the physical components of an organization’s IT infrastructure that support various processes within an organization. These assets include data processing equipment, such as computers and servers, and fixed equipment, such as server racks. In addition, hardware assets encompass data and electronic mediums employed for data storage and transmission, such as hard drives, and Solid-State Drive (SSD).

- Software assets encompass the software and applications used in an organization's operational activities. These assets can be grouped based on their purpose. This includes the organization's operating system and its version, business applications like Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) software, and database systems used for data storage and retrieval. Additionally, development tools encompassing Integrated Development Environment (IDE), compilers, debuggers, and other related components, as well as security software and administration software, contribute to the overall range of software assets used by the organization.
- Network assets are the physical and logical components that comprise a network infrastructure. These assets include physical devices like routers, firewalls, switches, wireless access points, transceivers, LAN adapters, Integrated Services Digital Network (ISDN) terminal adapters, and dongles, as well as network identifiers and network services that are part of an organization's infrastructure.
- Information assets refer to the valuable data owned or controlled by the organization that can be used to support its mission, objectives, and operations. An organization's information assets can be personal information that includes data about employees or customers collected by the organization or strategic information that contributes to the organization's strategic planning and decision-making.

This thesis primarily focuses on identifying logical components of a network, i.e., network services and identifiers of an organization that are accessible from the public Internet.

2.2.2 Network Assets

Network assets encompass various network identifiers, including IP addresses, Fully Qualified Domain Name (FQDN), subdomains, and Domain Name System (DNS) records. These identifiers provide information about the organization's network, which adversaries can later use during targeting [8].

The DNS plays a critical role in the network infrastructure, which not only translates IP addresses into user-friendly domain names but is also an essential factor in establishing trust for Internet service access [9]. DNS record, which is represented in a tuple $\langle name, TTL, class, type, data \rangle$ are stored on DNS servers and allows for configuration and control of other information about the domain name. They provide important information into the categories and geographical

distribution of servers. Some common types of DNS records are presented in the Table 2.1.

DNS Record	Description
A	Points to the IP address of a host
MX	Points to the mail server of a domain
NS	Points to the name server of a domain
CNAME	Allows creating aliases or canonical names for a host
SDA	Indicates authority for a domain
SRV	Specifies service records
PTR	Maps an IP address to a hostname
RP	Identifies the responsible person for a domain
HINFO	Provides domain information, such as CPU type and OS
TXT	Stores unstructured text records

Table 2.1: Common DNS records.

DNS records gathered publicly reveal information that can be used in different ways to target an organization. Collecting DNS information can help determine key hosts in the network and map the network infrastructure of an organization. The data field of Canonical Name (CNAME), Mail Exchanger (MX), and Name Server (NS) records indicate domains that could expire [9]. A potential attacker can re-register these domains and use them for malicious purposes. DNS records, including MX, TXT, and Sender Policy Framework (SPF) records, can potentially indicate the use of third-party cloud and SaaS providers like G Suite, Salesforce, or Office 365 [8]. Furthermore, DNS is now used for email spoofing mitigation through SPF, as well as authentication of Secure Shell (SSH) and Transport Layer Security (TLS) key fingerprints (TXT records) [10]. This information exposure can lead to dedicated phishing campaigns, SPF spoofing, and DNS hijacking attacks.

Moreover, numerous websites integrate third-party services through their DNS servers by adding an A or CNAME record and asserting ownership of the subdomain on the third-party service account. Nonetheless, the subdomain becomes vulnerable if the domain owner abandons it without removing references to the domain name from their information systems. Some third-party services do not verify these claims, allowing an adversary to claim ownership of an abandoned subdomain fraudulently. This enables them to control the subdomain's content, which could lead to a subdomain takeover attack. Additionally, retaining DNS resolution history can be advantageous, as an asset might still exist at a previous IP address even if the associated domain no longer directs to it [11]. Thus, DNS is a valuable asset to an organization.

An Autonomous System Number (ASN) is a unique identifier assigned to an

Autonomous System (AS), signifying the network to which a device belongs and the routing policies for that network [12]. AS denotes a collection of one or more IP prefixes administered by one or more network operators, who maintain a single, well-defined routing policy. While an ASN serves as a crucial identifier for a network and its routing policies, it does not directly identify a specific device on that network.

The allocation of IP addresses for an organization represents an additional valuable asset. Public IP addresses are typically assigned to organizations as a block or a range of consecutive addresses. Moreover, IP addresses can unveil critical organizational details, including the geographical location of servers, the Internet Service Provider (ISP) employed by the organization, and the scale of the organization [13]. Gaining knowledge of physical locations provides insight into hosting essential infrastructure and resources. This type of information gathering renders organizations susceptible to targeted attacks, such as identifying vulnerabilities within systems associated with those IP addresses or launching direct attacks like Distributed Denial-of-Service (DDoS) attacks. Additionally, it exposes organizations to potential risks regarding physical security threats and targeted social engineering attacks.

2.3 Reconnaissance

Reconnaissance is the process of gathering information about assets before the attacks [4]. Reconnaissance can be technological, i.e., using several available tools, or non-technological, i.e., through social engineering. This thesis mainly focuses on the technological method of conducting reconnaissance. Technologically oriented reconnaissance involves two distinct approaches - passive and active.

2.3.1 Active Reconnaissance

Active reconnaissance involves launching probes against the target system and examining the system's response to determine its properties. A probe refers to a form of network communication directed towards a specific system, and the resulting response from the system is analyzed to determine the features and properties of the target system. Target systems may employ mechanisms to analyze incoming communications and identify potential reconnaissance activities. If a probe is detected, systems may choose not to respond or take appropriate defensive measures [4].

Network scanning is one of the methods of conducting active reconnaissance. It is the process of scanning the whole network to identify active hosts on a network, exposed ports and services, the OS of a host, and network topology, and discover

their vulnerabilities. This information is collected by employing several techniques, which include:

- Checking for active hosts on a network by sending Internet Control Message Protocol (ICMP) ping requests to valid Internet Protocol (IP) addresses in user provided IPs range.
- Scanning for open ports for each target host by sending probe packets to each port to be checked.
- Scanning to identify services and their versions by carefully crafting probes against the host and analyzing the collected data. This also involves OS identification via fingerprinting or banner grabbing.

Tools	Description
Nmap	It is a network scanning tool that allows users to discover domains, services, and operating systems on a network using various scanning techniques, such as TCP SYN scan, UDP scan, and ICMP scan. [14]
Zmap	Zmap is a high-speed Internet-wide scanning tool designed to quickly scan the entire IPv4 address space, making it appropriate for large-scale data collection and research. [15]
Masscan	Masscan is a high-speed port scanner that can scan all 65535 ports of an IP address in a matter of minutes, making it ideal for identifying open ports on a large number of hosts. [16]

Table 2.2: Network scanning tools

There are some widely used automated network scanning tools presented in Table 2.2. Recent advancements in high-speed Internet-wide scanning, particularly with the introduction of tools such as Zmap and Masscan, have significantly reduced the time required to scan the entire IPv4 space, bringing it down to just a few minutes [17]. This rapid scanning capability has introduced new opportunities for empirically driven security research. Conversely, it has also provided attackers with the ability to conduct large-scale scans of vulnerable hosts within mere hours of a vulnerability disclosure.

In addition to employing network scanning tools for vulnerability disclosure, there are various widely used vulnerability scanning tools. The purpose of vulnerability scanning is to identify, detect, and evaluate security vulnerabilities in computer systems. Its main goal is to uncover known and unknown security risks

Tools	Description
OpenVAS	Open-source vulnerability scanner for comprehensive assessments of computer systems and networks. [18]
Nikto	Open-source web server vulnerability scanner specializing in identifying security issues in web servers. [19]
Nessus	Widely used commercial vulnerability scanner with an extensive vulnerability database for detecting and assessing security vulnerabilities in various systems. [20]

Table 2.3: Vulnerability scanning tools

within a network or computer system. Typically, vulnerability scanning involves using automated tools to examine a system for known vulnerabilities systematically. This scan actively searches for identified vulnerabilities, such as misconfigurations, out-of-date software, missing patches, or any other vulnerabilities attackers can exploit. A list of some widely used vulnerability scanning tools is provided in Table 2.3.

Despite abundant active scanning tools, conducting such scans without explicit permission can be considered unethical and illegal [21]. For ethical reasons, we rely on passive reconnaissance to gather data and do not perform active scanning ourselves.

2.3.2 Passive Reconnaissance

Passive reconnaissance comprises collecting information from third-party sources such as Internet information databases, search engines, and communication lines outside the organizational perimeter without interacting with the target system, making it undetectable by the owning organization. Since passive reconnaissance does not involve any interaction with the target system, it remains undetectable by the organization that owns the system. The main goal of the reconnaissance phase is to map a target in the real world (such as a company, government, or other organization) to its corresponding representation in the cyber world [22]. There exists a variety of methods and techniques employed during the reconnaissance phase based on the diversity of the information required.

One of the widely used frameworks for passive reconnaissance is the Open Source Intelligence (OSINT) Framework. The OSINT framework is a collection of tools, techniques, and resources to collect information from publicly accessible sources [23, 24]. The current interconnected world generates an extensive volume

of data, a significant portion of which is publicly accessible, allowing any Internet user to access it from any location and at any time. In this regard, OSINT is a form of intelligence that benefits from the open nature of the Internet by accumulating, analyzing, and correlating data from the entire cyberspace to generate valuable insights [25]. By leveraging OSINT, researchers and security professionals gain the ability to discover and analyze data that can aid in understanding and assessing potential vulnerabilities, threats, and targets in a network. The OSINT Framework comprises multiple categories, including search engines, social networks, and domain and IP address analysis.

2.3.2.1 Search Engine Supported by Internet Scanning

As we do not conduct active network scanning ourselves due to ethical implications, we leverage the capabilities of specialized search engines that are specifically designed to democratize the process of Internet-wide scanning. They conduct comprehensive Internet scans and present the data in an accessible format.

- **Shodan** Shodan is a specialized search engine that enables users to locate embedded devices and computer systems, such as routers and servers, using a variety of filters and criteria [26]. Shodan scans the Internet to discover accessible devices and services, collecting data such as IP addresses, ports, and service banners. This information is stored in a database accessible through a web interface and an Application Programming Interface (API). Shodan conducts the scan by gathering and analyzing service banner metadata. A service banner is a message or set of information sent by a network service when a client establishes a connection with it. Typically, the service banner includes the service's name, version number, and possibly additional information, such as the server's operating system or the current date and time. Shodan is an important tool, as it can help determine whether there are known vulnerabilities based on the software and firmware levels identified. Generally, Shodan gathers and analyzes metadata information from the service ports in Table 2.4.
- **Censys** is a public search engine supported by data obtained through continuous Internet-wide scans. It allows users to perform full-text searches on protocol banners and query various derived fields [27]. Censys regularly scans the public IPv4 address space, covering significant ports and protocols. It validates the collected data and executes application-layer handshakes using a framework for pluggable scanners. Censys performs daily scans on the top 137 ports, along with an additional 1440 ports in cloud environments, while refreshing all known services within a 24-hour timeframe. The search

Service	Ports
File Transfer Protocol (FTP)	21
Secured Shell (SSH)	22
Telnet (TELNET)	23
Simple Network Management Protocol (SNMP)	161
Unsecured Web (HTTP)	80
Secured Web (HTTPS)	443
Internet Message Access Protocol (IMAP)	143
Simple Mail Transfer Protocol (SMTP)	25
Session Initiation Protocol (SIP)	5060
Real-Time Streaming Protocol (RTSP)	554

Table 2.4: Ports scanned by Shodan

capabilities of Censys include full-text searches, regular expressions, and numeric ranges, which can be combined using Boolean logic. The search engine maintains up-to-date records of publicly accessible IPv4 hosts, Alexa Top 1 million websites, and known X.509 certificates, enabling users to conduct queries against these datasets.

Censys is a valuable tool for organizations to assess their external-facing attack surface. Given the challenge of monitoring network-connected devices and the potential for inadvertent exposure of private devices and services, Censys facilitates queries based on network blocks and Autonomous Systems (ASes). This functionality enables organizations to easily extract all relevant data collected by Censys about their publicly accessible services [27].

2.3.2.2 Subdomain Enumeration

Subdomain enumeration involves identifying a comprehensive list of subdomains associated with a target domain, which allows for a more thorough reconnaissance. While the primary domain is often highly secure and adheres to strict security protocols, it cannot be assumed that all other subdomains will do the same. By enumerating subdomains, attackers can gain valuable insights into the underlying architecture, uncover hidden user interfaces and admin panels, and potentially exploit low-hanging vulnerabilities. Less frequently visited and unknown domain names create a blind spot that attackers can exploit. In addition, developers create “test” or “shortcut” subdomains that they can later forget to clean up [28]. This process of subdomain enumeration can help gather data and details on subdomains that should have been decommissioned. Table 2.5 presents the subdomain enumerations tools used in this thesis.

Tools	Description
Sublist3r	Sublist3r adopts an extensive strategy for subdomain enumeration, making use of various search engines like Google, Yahoo, Bing, and Ask, along with additional sources such as Netcraft, Virustotal, and ReverseDNS to augment the process. [29]
DNSDumpster	DNSdumpster.com is a free domain research web application that uses DNS lookup and crawling methods to discover hosts and gather extensive information related to a domain. [30]

Table 2.5: Subdomain enumeration tools

2.3.2.3 GSMA Documents

The IR.21 roaming database [31] serves as a crucial source of information for adversaries, providing standardized details about the network infrastructure and interconnection for each operator. Administered by Global System for Mobile communications Association (GSMA), the global IR.21 database grants access to Global System for Mobile Communications (GSM), 3G, and Long Term Evolution (LTE) operators worldwide. It offers a comprehensive overview of the operator’s external interfaces and parameters visible to other operators. These details encompass the IP addresses of various network nodes, such as GGSN, SGSN, MMSC, AAA servers, and DNS Servers, which connect to the GPRS roaming exchange. Operators use this information to configure their firewalls and border gateways accordingly.

The IR.67 document [32] covers DNS and E.164 Number Mapping (ENUM), providing information on general and service-specific server configuration, domain name formats, and usage procedures. It also addresses the GRX/IPX network, focusing on DNS/ENUM servers connected to this backbone network.

2.3.2.4 National Vulnerability Database

The National Vulnerability Database (NVD) is a repository for vulnerability management data based on established standards. It enables “automation of vulnerability management, security measurement, and compliance” [33].

The NVD consists of Common Vulnerabilities and Exposures (CVE) entries, which constitute a publicly accessible database of reported cybersecurity vulnerabilities, where vulnerabilities are added and updated regularly. Each CVE entry is assigned a unique identifier, such as CVE-2021-1023. Furthermore, each entry is accompanied by a vulnerability score in the Common Vulnerability Scoring

System (CVSS) format. The CVSS scoring system is used to assess vulnerability severity and prioritize vulnerability remediation. The CVSS scores span from 0 to 10, wherein 0 indicates the lowest severity, and 10 denotes the highest severity level assigned to a particular CVE entry. Additionally, CVE entries are associated with specific Common Platform Enumeration (CPE) names. CPE is the structured naming scheme for identifying various applications, operating systems, and hardware devices in an enterprise’s computing assets [34]. These CPE names offer important details, including information about the software vendor, name, version, language, and edition. The NVD uses the CVE and the CPE systems to provide detailed vulnerability data. For example, when a new vulnerability is added to the NVD, it is given a CVE identifier, and the systems or software that it affects are identified using CPE.

2.3.2.5 Other Tools

In addition to the tools and techniques described, there exists a diverse array of Open Source Intelligence (OSINT) tools, each serving unique purposes and gathering various data types, including, but not limited to, network data presented in the Table 2.6.

Tools	Description
Spiderfoot	A reconnaissance tool that retrieves a wide range of intelligence from over 100 public data sources (OSINT), including IP addresses, domain names, email addresses, and names, among other elements [35].
Recondog	Recondog gathers information about its scan targets by using external databases and locally driven searches. [36]
Maltego	Maltego extracts data from open sources and presents it in a graphical format. It allows the discovery of data relationships and links, including those that are hidden or not obvious. [37]

Table 2.6: OSINT tools

Moreover, dedicated tools, such as Wappalyzer, specifically designed to detect the technologies used on websites, provide insights into a target’s software stack. In this thesis, however, we specifically leverage a technique known as banner grabbing. Banner grabbing is the technique to determine the operating system running on the target system [38]. This technique enables the determination of the protocol,

application, version, hostname, device type, and OS type of the services under analysis. We rely on the banners grabbed by Censys and Shodan and conduct analysis to grab only relevant information from the banner.

```
HTTP/1.1 200 OK
Server: nginx 1.18.0
Date: Tue, 28 Mar 2023 02:11:42 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: Express
Content-Security-Policy: default-src 'self' https://*.elisa.fi
Cache-Control: public
Strict-Transport-Security: max-age=15724800; includeSubDomains
```

Figure 2.1: A sample banner grabbed from Shodan

From 2.1, we can gather valuable information, for example, the version of the server (nginx 1.18.0) and the service (HTTP/1.1). With this type and version of an application, it is possible to scan for known and exploitable vulnerabilities in that version by using the NVD and the CVE system mentioned in section 2.3.2.4. With the tech stack and version identified from the banner, the NVD and the CVE database can identify and mitigate potential security risks associated with those technologies.

2.4 Literature Review

Several studies have been conducted over the years to map devices on the Internet, with a subset of these using Shodan as a tool for this purpose.

2.4.1 Mapping Devices on the Internet using Shodan

Project SHINE [26] conducted in 2015 employed Shodan to quantify the direct connectivity of Supervisory Control and Data Acquisition (SCADA)/ICS devices to the Internet. The authors found around 500,000 ICS devices connected to the Internet. This study concluded that, among other reasons, the primary factors contributing to device exposure to the public Internet were poorly configured devices and default device settings. To evaluate the indexing functionality of Shodan,

Bodenheim et al. [39] conducted an experiment where they intentionally exposed several industrial systems to the Internet. Their findings indicated that Shodan successfully detected, indexed, and identified the deployed devices. In addition, they proposed that manipulating service banners showed promise as a countermeasure to prevent device identification through Shodan search queries. In [40], the authors employed honeypot technology to analyze the Shodan search engine. The results revealed a significant increase in Internet scans targeting ICS honeypots after they were indexed by Shodan, suggesting that the visibility and attractiveness of these honeypots were amplified after their visibility in Shodan.

2.4.2 Using Shodan for Vulnerability Assessments

In addition to using Shodan to map devices on the Internet, various studies have explored the use of search engines, particularly Shodan, for vulnerability assessments of ICS [41, 42] and IoT [43, 44] devices. The authors of [41] combined Shodan with the NIST NVD and CVE-Details to evaluate vulnerabilities in SCADA systems. Meanwhile, [42] compared Shodan and Google-based vulnerability assessments for IT systems with contact-based assessments using Nessus. The Shodan-based assessment centered on scanning Shodan banners to extract vulnerabilities reported for each device based on CVE or NVD. Furthermore, [43] conducted remote security assessments of publicly accessible IoT devices using an expanded data set derived from initial public Shodan queries, while [44] focused on discovering vulnerabilities in SCADA devices, web cameras, and health networks by employing the Shodan API and testing default passwords against identified SCADA devices to analyze responses and identify vulnerable systems.

Some recent research also focuses on leveraging Shodan to obtain more information. Genge Enachescu in [45] developed an automated passive vulnerability assessment tool, ShoVAT. This tool leverages Shodan's indexing capabilities to discover services, reconstruct CPE information, and extract vulnerabilities from the NVD. Furthermore, another study [46] introduced a tool named ShoBeVODSDT, a non-intrusive tool that used Shodan and Binary Edge to detect vulnerabilities in open data sources. In alignment with these previous studies, our thesis incorporates the NIST NVD and CVE system in conjunction with search engine methodologies to assess vulnerabilities.

2.4.3 Aggregate Attack Surface Mapping

There has also been research on using multiple search engines to map the devices on the Internet [47, 48]. In the study conducted by [48], historical data collected from multiple search engine tools, such as Shodan and Censys, was analyzed. The findings indicated that relying solely on Shodan could not characterize network

exposures effectively. Therefore, the authors recommended the aggregation of data from multiple sources instead of a single search engine. In [47] the authors evaluate several search engines, including Shodan, BinaryEdge, and Censys, to identify the aggregate attack surface of ICS. They find significant variation in the exposures discovered by each tool, but when the attack surface is merged, they recognize distinctive contributions from each tool. The results conclude combining multiple tools provides a more comprehensive view of the attack surface. Building upon these research findings, our thesis uses multiple search engines to create an aggregated picture of the attack surface.

2.4.4 Asset Discovery Framework

In addition, it is worth noting that a significant portion of existing research in the field concentrates on using search engines to map assets present on the Internet. To further contribute to this knowledge, Mathew et al. in [11] provides an overview of asset discovery techniques and propose a framework to systematize the asset discovery techniques. The authors suggest the pivotal role of DNS in asset discovery, particularly when identifying network services. They emphasize the usefulness of DNS in uncovering IPv6 address space, thereby suggesting that asset discovery techniques pertaining to DNS should be of focus. In this thesis, we use DNS and DNS-based tools along with search engine data to provide a comprehensive discovery of network services from the Internet.

Chapter 3

Methodology

To address the research questions, we initially conducted a comprehensive literature review, which revealed a research gap in attack surface mapping for telecommunications operators. To address this gap, we conducted a study involving the analysis of network assets, reconnaissance tools, and techniques related to attack surface mapping. Subsequently, we present a methodology specifically designed to address this challenge within the telecommunications sector. Firstly, we devised a research workflow that outlines the steps of our proposed methodology. Following that, we developed a Python-based tool to automate the implementation of our methodology. This tool serves as a prototype, showcasing the practical application of our proposed approach. In this chapter, we will discuss this in detail.

3.1 Research Workflow

In this section, we present the research workflow of the adopted methodology, which uses existing tools and techniques and applies them to map the attack surface of a telecommunications operator. The research workflow, depicted in Figure 3.1, outlines the step-by-step process proposed to map the attack surface of a telecommunications operator.

Step 1: MNO as an Input: The first step of this methodology involves providing the name of the MNO as an input. Subsequent steps involve collecting and analyzing further information based on the provided MNO name.

Step 2: IR.67 Database: As mentioned in section 2.3.2.3, the IR.67 database provides DNS configuration guidelines for an operator. For instance, the domain name guideline for the Evolved Packet Data Gateway (ePDG) entity in the Evolved Packet Core (EPC) component of the telecommunications infrastructure is presented below. The <<MNC>> and <<MCC>> variables represent the Mobile Network Code (MNC) and the Mobile Country Code (MCC) of a specific Service

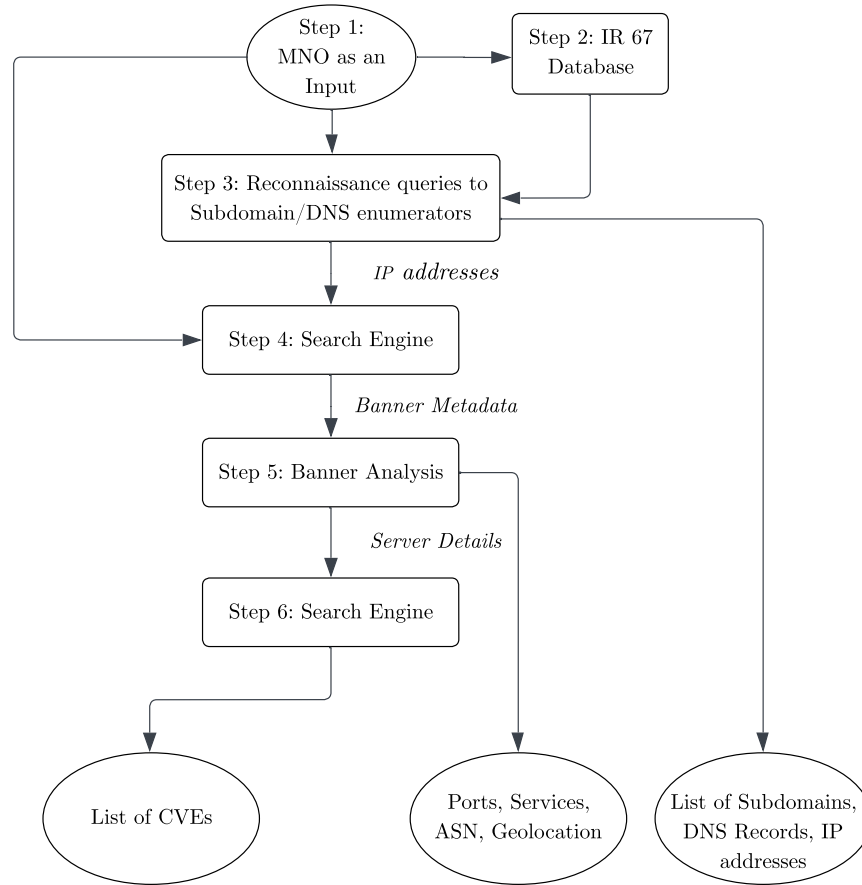


Figure 3.1: Research workflow used for attack surface mapping.

Provider.

```
epdg.epc.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org
```

This constructed domain name, comprising the MNC and MCC of an operator, along with the MNO name, is passed as an input to the subsequent steps.

Step 3: Reconnaissance Queries to Subdomain/DNS Enumerators: Subdomain enumerator tools are employed to obtain a list of subdomains, host-names, and IP addresses associated with an MNO. DNS records (MX, NS, A, TXT, CNAME) are also collected for these subdomains. This step allows us to map the network architecture of a mobile network operator and narrow down the hosts for further analysis. In this thesis, we used the tools Sublist3r [29], DNSdumpster

[30], and MxToolBox [49] for the aforementioned purpose.

Step 4: Search Engine: In this step, we focus on gathering banner metadata from search engines for all IP addresses associated with an MNO. To accomplish this, we use two search engines: Censys [27], and Shodan [50]. These search engines provide a wealth of information for each IP address but have distinct structures for their banner metadata. To ensure that we extract relevant and valuable information from the acquired banner metadata, it is essential to identify and isolate the key details. In light of this, we have summarized and described the selected metadata from Shodan and Censys in Table 3.1 and 3.2, respectively, which reflect such considerations. It is worth noting that the banners obtained from each search engine may differ due to variations in scanning protocols, scan time, and scanning frequency. Consequently, there might be slight variations in the banner results between the two search engines. We aim to merge these results while ensuring we retain unique records for each IP address.

Meta-data	Description
asn	Autonomous system number
data	Contains all data related to the device and its published services (main banner)
ip	The IP address of the device
port	The port number on which the service is operating
location	Geographic location of the device
os	The operating system that powers the device
opts	Supplemental data not formalized in the main banner
opts.vuln	Vulnerabilities found on the service and corresponding countermeasures
isp	ISP responsible for the IP space

Table 3.1: Meta-data used from Shodan

Step 5: Banner Analysis: After an initial grab of banners from the search engine tools, we conduct further analysis by merging the banners and consolidating the information for each field associated with a specific IP address. This consolidation creates a dataset consisting of unique records for each IP address. The next step in banner analysis involves extracting additional details related to the technologies used, such as the server and its version, from the banners. This includes details such as the server software used and its specific version number. This step also involves generating statistical data based on the gathered information.

Furthermore, during this stage, we perform the classification of IP addresses

Meta-data	Description
autonomous system	An object containing fields related to the autonomous system that the host is reachable at
ip	The IP address of the host
location	An object containing fields related to the host's geographical location
services	An array containing preview information of every service on the matching host
services.port	The port number that the service is running on
services.service_name	The name of the service
services.banner	The response message from a service indicating its version, software information, and protocols used
services.operating_system	The operating system associated with the service

Table 3.2: Meta-data used from Censys

and hosts based on the service and port details acquired from the banners. In the context of a mobile network operator, it is important to consider that an IP address can either be assigned to a private customer or be part of the telecommunications infrastructure, and it may also be publicly exposed on the network. We also store those IP addresses that have no services exposed. This indicates that these IPs do not have any services or ports exposed that are scanned by search engines. This list of IPs can be used for the active scanning of ports later.

Step 6: NVD and CVE Lookup The information related to the server and its version number gathered in step 5 is further used to gather potential associated vulnerabilities. To facilitate this process, we leverage the NIST National Vulnerability Database (NVD) mentioned in section 2.3.2.4. Using the server information obtained from the banners, we perform a lookup within the NVD to identify any known CVEs related to the specific server software and its version. This step allows us to identify potential weaknesses in the server's configuration.

3.2 Tool Implementation

We have designed and implemented a Python-based command line tool to streamline the research workflow outlined in Figure 3.1. This tool leverages a combination of passive reconnaissance tools while automating the network asset discovery process. It also generates statistical summaries based on the collected banners, enabling valuable insights and aiding in the assessment of the telecommunications operator's overall security.

The developed tool follows a modular architecture and consists of three main modules: DNS Reconnaissance Module, Banner Grabbing Module, and Banner Analysis Module. The modular architecture followed by the tool is presented in Figure 3.2.

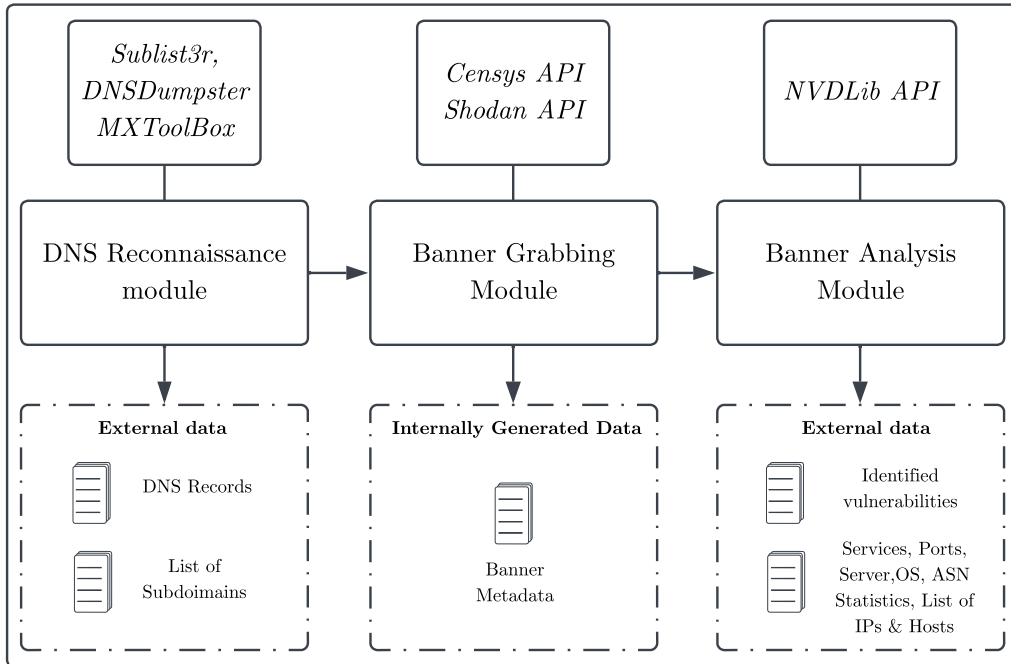


Figure 3.2: Graphical representation of the tool’s architecture showcasing different modules, their interdependencies, and output data.

3.2.1 DNS Reconnaissance Module

The first step in the implementation is gathering DNS, domain, and subdomain-related data, which is gathered using the DNS Reconnaissance Module. In this module, we automate steps 1, 2, and 3 of the research workflow in Figure 3.1. The first input to this module is the name of the Mobile Network Operator (MNO). After receiving the input, the MNO name is then appended to various top-level domains such as *.com*, *.org*, and *.fi*, which are then used as input for the rest of the analysis. For an operator named *MNO*, the domains *MNO.com*, *MNO.fi*, and *MNO.org* are used to get subdomains and host records from Sublist3r and DNSdumpstr.

To use the IR.67 database, we maintain our local database of MNC and MCC for every operator in a JSON file. When given an operator name as an input, we

look up the MNC and MCC details of our *codes.json* file and then construct the domain names by replacing the respective codes in the format. For instance, for the operator *MNO*, the MCC and MNC codes are 043 and 422, respectively, and the ePDG domain name in EPC would appear as shown below.

```
epdg.epc.mnc422.mcc042.pub.3gppnetwork.org
```

Furthermore, we use MXToolBox to obtain the DNS records for the domains and subdomains gathered from the previous step. A typical DNS record obtained from MXToolBox is shown in Figure 3.3, which is then further processed to store in a more structured format as presented in Figure 3.4. By default, results from *MXtoolbox* are stored in a *.txt* file, which we subsequently process to store in a more structured manner within a *.csv* file. All the DNS records are exported and stored in a file named *mno_dns.csv*.

3.2.2 Banner Grabbing Module

In this module, we automate Step 4 of the workflow in Figure 3.1. The banner-grabbing module performs reconnaissance using Shodan and Censys APIs. It involves retrieving banner information and other details such as ASN, location, operating system, and vulnerabilities from the search engine APIs. As mentioned before, since both Shodan and Censys may provide different banners, we collect the banner data separately from Shodan and Censys and store them under two different categories: *banner_shodan* and *banner_censys*. This allows us to have a complete set of banners from both search engines. To ensure the collected banner information is unique and aggregated, we combine the banners obtained from both search engines. By doing so, we eliminate duplicate entries and create a consolidated collection of banners that provides a more comprehensive view of the target system or infrastructure. Figure 3.5 and 3.6 illustrate the service banner grabbed by Shodan and Censys, respectively, for the same IP address. We aggregate this banner by comparing and collecting the unique banner information from both banners in the *banner_aggregated* field. Figure 3.7 presents the banner metadata we store for the given IP Address in a JSON file.

```

asp.telecom.fi
A
DNS Record Published: DNS Record found
{"Type": "A", "Domain Name": "asp.telecom.fi", "IP Address":
  ↪ "217.152.49.12", "TTL": "60 min", "Asn": "[]", "IsIPv6": "
  ↪ False"}
-----
NS
DNS Open Recursive Name Server: No Open Recursive Name Server
  ↪ Detected
{"Type": "NS", "Domain Name": "ns-fi.telecom.net", "IP Address":
  ↪ "193.229.0.49", "TTL": "60 min", "Asn": "[]", "IsIPv6": "
  ↪ False"}
-----
MX
{"Pref": "0", "Hostname": "pop.kolumbus.fi", "IP Address":
  ↪ "193.229.0.46", "TTL": "60 min", "Asn": "[]", "IsIPv6": "
  ↪ False"}
-----
CNAME
DNS Record Published: DNS Record not found
-----
TXT
DNS Record Published: DNS Record not found
-----

```

Figure 3.3: DNS record gathered from MXToolBox

```

Subdomain,A,NS,MX,CNAME,TXT
asp.telecom.fi,217.152.49.12,ns-fi.telecom.net,pop.kolumbus.fi,,

```

Figure 3.4: DNS records stored for a telecommunications operator after processing

3.2.3 Banner Analysis Module

This module focuses on automating steps 5 and 6 of the workflow presented in Figure 3.1. The banner analysis module is important in processing the collected banners and extracting relevant server-related data. For example, consider the banner shown in Figure 3.7, where we further process it and collect the server

```

"banner_shodan": [
  {
    "port": 443,
    "banner": "HTTP/1.1 404 Not Found\r\nServer: Apache/2.4.54 (
      ↪ Unix) OpenSSL/1.1.1n\r\nContent-Length: 196\r\nContent
      ↪ -Type: text/html; charset=iso-8859-1\r\nSet-Cookie:
      ↪ BIGipServer~TCMLAB~labtcm.telecom.fi- path=/; Httponly
      ↪ ; Secure\r\n\r\n"
  }
]

```

Figure 3.5: A banner grabbed from Shodan

```

"banner_censys": [
  {
    "service_name": "HTTP",
    "banner": "HTTP/1.1 404 Not Found\r\nServer: Apache/2.4.54 (
      ↪ Unix) OpenSSL/1.1.1n\r\nContent-Length: 196\r\nContent
      ↪ -Type: text/html; charset=iso-8859-1\r\nSet-Cookie:
      ↪ BIGipServer~TCMLAB~labtcm.telecom.fi-; path=/;
      ↪ Httponly; Secure\r\n",
    "port": 443
  },
  {
    "service_name": "HTTP",
    "banner": "HTTP/1.1 401 Unauthorized\r\nServer: Apache
      ↪ /2.4.54 (Unix) OpenSSL/1.1.1n\r\nWWW-Authenticate:
      ↪ Basic realm=\"CT\"\r\nContent-Length: 381\r\nContent-
      ↪ Type: text/html; charset=iso-8859-1\r\nSet-Cookie:
      ↪ BIGipServer~TCMLAB~labtcm.telecom.fi-; path=/;
      ↪ Httponly; Secure\r\n",
    "port": 8445
  }
]

```

Figure 3.6: A banner grabbed from Censys

data Apache/2.4.29. It is important to note that the structure of the captured banners varies depending on the associated service. Therefore, we adopt a system-

atic approach by processing each service banner differently and extracting only the relevant server details from the grabbed banner. Table 3.2.3 presents an overview of the different banner structures encountered and the corresponding server information obtained from the analysis process.

In addition to capturing server details, the module incorporates a vulnerability analysis component. Leveraging the capabilities of the `nvdlib` API [51], we search for CVE information based on the extracted server details from the banner and the CPE details collected from search engines. By combining the CVE information retrieved through the `nvdlib` API with the vulnerability details obtained from the `opts` field in Shodan, we generate an output file containing potential vulnerabilities along with its severity present within the telecommunications operator network infrastructure.

3.3 Tool Usage

Figure 3.8 represents the snapshot of the command line tool developed in Python. Before using the tool, users can set up their API keys for Shodan, Censys, NVD and, MXToolBox using the `apikeysh.sh` file. The tool runs with a mandatory argument `mno`, corresponding to the name of the Mobile Network Operator (MNO) for which the tool is intended to run. For a more targeted analysis, the tool offers optional flags.

- The `ip` and `host` flags retrieve all IP addresses and hosts associated with the specified MNO operator, respectively. The resulting lists are saved in separate CSV files named `mno-ip.csv` and `mno-host.csv`.
- The `sub` flag retrieves a list of subdomains for the given MNO operator, which is then saved to a CSV file named `mno-subdomains.csv`. Users can also provide a list of keywords that can be used to filter the subdomains.
- The `dns` flag retrieves MX, NS, CNAME, A, and TXT records for the specified MNO operator. The resulting records are saved to a CSV file named `mno-dns.csv` file. Users can further specify an additional argument representing the specific type of DNS record to search for.
- The `stat` flag provides a statistical summary for services, ports, vulnerabilities, and server details related to the specified MNO operator. The results are saved to an Excel file named `mno-stats.xlsx`.
- The `c` flag stores and provides a list of IPs with no services or ports that are reachable.

Banner	Server
<pre>{ "port": 80, "banner": "HTTP/1.1 200OK\r\ ↪ nDate: Mon, 26Jun 202311: ↪ 13:36GMT\r\nServer: ↪ Apache/2.4.29 (Ubuntu)\r\ ↪ nLast-Modified: Tue, 19 ↪ Nov 201911:27:31GMT\r\ ↪ nETag: \"0-597b15994d8c0 ↪ \"\r\nAccept-Ranges: ↪ bytes\r\nContent-Length: ↪ 0\r\nContent-Type: text/ ↪ html\r\n\r\n" }</pre>	<pre>{ "server": "Apache/2.4.29 (Ubuntu ↪)" }</pre>
<pre>{ "port": 21, "banner": "220 ProFTPD Server (↪ ProFTPD)\r\n530 Login ↪ incorrect.\r\n214...\n ↪ TVFS\n UTF8\n211 End\r\n" }</pre>	<pre>{ "server": "ProFTPD Server (↪ ProFTPD)" }</pre>
<pre>{ "port": 22, "banner": "SSH-2.0-OpenSSH_7.4\ ↪ nKey type: ssh-rsa\nKey: ↪ " }</pre>	<pre>{ "server": "SSH-2.0-OpenSSH_7.4" }</pre>

Table 3.3: Banner and server information

- The `port` and `service` flags allow users to search for a specific port or service and then return a list of IPs that have the specified port or service exposed.

This additional tool feature allows users to search for a port or service they are interested in.

- The `server` flag allows the user to search for a particular server name such as `Apache` and provides a list of all the versions associated with the server, along with the list of IP addresses for each version.
- The `cve` flag enables the user to search for CVEs associated with a given server version. The results include CPE, CVE Id, severity, and description of the CVE. The results are saved to an Excel file named `mno_cve_tech.xlsx`


```

{"ip_address": "",
  "isp": "",
  "asn_": {},
  "location": {}
  opts": {
    "vulns": [],
    "heartbleed": "2023/06/20 19:06:09 62.71.106.231:443"
  },
  "operating_system": [
    {
      "cpe": "cpe:2.3:a:f5:big\\-ip_local_traffic_manager
        ↪ :*:~*:~*:~*:~*:~*:~*:~*",
      "vendor": "F5",
      "product": "BIG-IP LTM",
      "other": {
        "family": "BIG-IP"
      }
    }
  ],
  "banner_aggregate": [
    {
      "service_name": "HTTP",
      "banner": "HTTP/1.1 404 Not Found\r\nServer: Apache/2.4.54 (
        ↪ Unix) OpenSSL/1.1.1n\r\nContent-Length: 196\r\nContent
        ↪ -Type: text/html; charset=iso-8859-1\r\nSet-Cookie:
        ↪ BIGipServer~TCMLAB~labtcm.telecom.fi-; path=/;
        ↪ Httponly; Secure\r\n",
      "port": 443
    },
    {
      "service_name": "HTTP",
      "banner": "HTTP/1.1 401 Unauthorized\r\nServer: Apache
        ↪ /2.4.54 (Unix) OpenSSL/1.1.1n\r\nWWW-Authenticate:
        ↪ Basic realm=\"CT\"\r\nContent-Length: 381\r\nContent-
        ↪ Type: text/html; charset=iso-8859-1\r\nSet-Cookie:
        ↪ BIGipServer~TCMLAB~labtcm.telecom.fi-; path=/;
        ↪ Httponly; Secure\r\n",
      "port": 8445
    }
  ]
},

```

Figure 3.7: Aggregate banner

```

hp@DESKTOP-MKRIEM6:~/thesis/thesis code$ python3 telco_attack_surface.py -h
usage: telco_attack_surface.py [-h] [-stat [STAT ...]] [-host [HOST]] [-c [CLASSIFY]]
                               [-sub [SUBDOMAIN ...]] [-ip [IPS]] [-service [SERVICES]]
                               [-p [PORTS]] [-server [SERVER]] [-cve [CVES]] [-dns [RECORDS ...]]
                               MNO

positional arguments:
  MNO                  The Mobile network operator to search for

options:
  -h, --help          show this help message and exit
  -stat [STAT ...]    Returns statistics for Services, Ports, Server and Vulnerability
  -host [HOST], --host [HOST]
                      List of hosts
  -c [CLASSIFY], --classify [CLASSIFY]
                      List of classified hosts with no services exposed
  -sub [SUBDOMAIN ...], --subdomain [SUBDOMAIN ...]
                      List of subdomains; Optional string to filter the subdomains.
  -ip [IPS], --ips [IPS]
                      List of Ip addresses
  -service [SERVICES], --services [SERVICES]
                      Search for hosts related to a specific service e.g., HTTP, SSH
  -p [PORTS], --ports [PORTS]
                      Search for hosts related to a specific port e.g., 80, 443
  -server [SERVER], --server [SERVER]
                      Search for a specific server e.g., Apache
  -cve [CVES], --cves [CVES]
                      Search for CVEs associated with a specific server
  -dns [RECORDS ...], --Records [RECORDS ...]

```

Figure 3.8: Snapshot of the tool

Chapter 4

Evaluation and Results

In order to evaluate the tool, we ran it for one of the major operators in Finland. To maintain anonymity, we will refer to this operator as “MNO” throughout this discussion. Furthermore, our analysis focused exclusively on the results obtained from websites with the *.fi* top-level domain (TLD). We used several commands to run the tool and gather results.

4.1 Subdomains

The tool returned 113 subdomains for the operator. This list of subdomains also revealed some test-related subdomains. By default, while running the tool with the `-sub` flag, it returns all the subdomains it can gather for the given operator. In addition, the subdomains can be filtered by passing additional arguments.

We ran the tool for the operator `mno` and filtered out the list with the keywords `tst`, `test`, `sandbox`, `beta`, `qa` using the command in Figure 4.1. Table 4.1 presents a subset of such subdomains gathered. Test subdomains often have less stringent security measures than production systems. The presence of such subdomains doesn’t necessarily mean that they are vulnerable endpoints. However, such subdomains could be an attractive target for attackers.

```
telco_attack_surface.py mno -sub tst,test,sandbox,beta,qa
```

Figure 4.1: Command for gathering subdomains

apichattst.mno.fi
testsurfmanager.mno.fi
broker-api-sandbox.mno.fi
developer-api-sandbox.mno.fi
developer-test.mno.fi
dnstest.mno.fi
hosting-test.mno.fi
intratest.mno.fi
mobiiilimaksutest.mno.fi
kirja-qa-validator.mno.fi

Table 4.1: Sample of gathered subdomains

4.2 DNS Records

The results for the DNS records for the operator further revealed more details about the operator. By default, the tool presents the DNS results for all the subdomains it has gathered. An additional flag can be sent to gather specific DNS records. We ran the tool with the command in Figure 4.2, and a sample of observed results is presented in Table 4.2. The CNAME record for the subdomain `ar.mno.fi` indicates that the domain is using `cloudfront.net`, a content delivery network (CDN) provided by Amazon Web Services (AWS). Similarly, the CNAME record for `designsystem.mno.fi` suggests that the MNO Design System can be hosted or deployed using the Netlify platform. These findings shed light on the infrastructure and services employed by the operator, offering valuable information for further analysis and assessment.

```
telco_attack_surface.py mno -dns CNAME
```

Figure 4.2: Command for gathering DNS records

4.3 Statistical Summary

We used the `ip` flag in the tool to collect the IP addresses associated with the specified MNO operator. The tool returned 501 distinct IP addresses and their corresponding hostnames for the given operator and `.fi` TLD (*mno.fi*). Using the `-c` flag, we classified the IP addresses based on the exposed services, that is, the network services or applications that are accessible and available on the IP

Subdomain	CNAME	Description
ar.mno.fi	d3h70a8zndek6p.cloudfront.net	Uses Amazon CloudFront CDN
vipraportointi.mno.fi	soneravip.meridix.se	Uses Meridix platform for reporting purposes
viprec.mno.fi	mnolive.icallsuite.com	Uses icallsuite platform
yhteiso.mno.fi	mnofinal-fi-community.insided.com	Uses the inSided platform
designsystem.mno.fi	romantic-hermann-3321ec.netlify.com	Hosted or deployed using Netlify

Table 4.2: Sample of DNS CNAME records gathered

addresses associated with the specified MNO operator from the grabbed banners. Our analysis revealed that out of the collected IP addresses, 401 did not have any exposed services. This implies that either these IPs are not reachable from the public Internet or that search engines do not scan the services running on them.

Additionally, we employed the `-stat` flag within the tool to generate a statistical summary encompassing exposed ports, services, server details grabbed from the banner, and vulnerabilities associated with the servers for the given MNO operator. Users can gain insight into the overall network map and topology by analyzing the summary, enhancing their understanding of the MNO operator's network architecture. It also enables the identification of any potential irregularities or anomalies within the network.

Firstly, the summary provides an overview of exposed ports and services, which helps to determine the accessibility and availability of specific network services. Figure 4.3 depicts the distribution of services, revealing that HTTP and HTTPS (port 80, 443, 8443) were the most prevalent. Next, we visualized the summary of the server details grabbed from the banner. Analyzing these data helped us to understand the technologies and configurations used by the MNO operator. As illustrated in Figure 4.4, Apache emerged as the widely adopted server for web services for the given MNO operator.

Furthermore, the statistical summary also incorporates vulnerability information associated with the server version in use. Table 4.3 offers a comprehensive overview of vulnerabilities and their severity levels. The severity classification of each vulnerability is determined based on the CVSS score, as elaborated in Section 2.3.2.4. The severity levels are categorized as CRITICAL, HIGH, and MEDIUM, with CRITICAL denoting the highest level of severity. A higher severity level indicates a greater potential risk and the likelihood of more severe consequences.

We further analyze the vulnerability of the particular server version in detail using the `cve` flag in the tool. Table 4.4 presents vulnerability details extracted from the tool's most used server version for the given MNO operator, Apache/2.4.54. The CVEs describe various vulnerabilities in the version of Apache HTTP Server. The vulnerabilities involve memory read/write issues, inconsistent interpretation

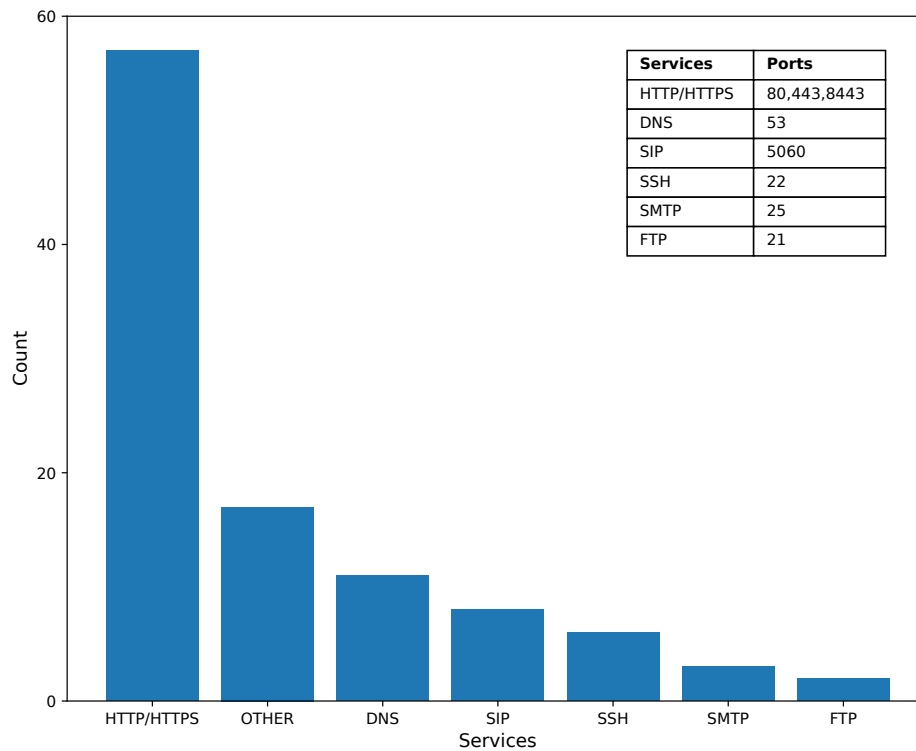


Figure 4.3: Services distribution

Server	Server Version	Server Distribution (% of total servers)	CVE Count		
			Severity CRITICAL	Severity HIGH	Severity MEDIUM
Apache	Apache/2.4.29	10.34%	13	15	13
	Apache/2.4.54	12.07%	2	2	1
	Apache/2.4.6	6.90%	13	13	13
	Apache/2.4.37	8.62%	13	19	14
Subtotal		37.93%	41	49	41
Nginx	nginx/1.18.0	5.17%	0	4	0
	nginx/1.20.1	1.72%	0	3	0
	nginx/1.25.1	3.45%	0	2	0
Subtotal		10.34%	0	9	0
OpenSSH	SSH-2.0-OpenSSH.8.2p1	6.90%	0	4	2
	SSH-2.0-OpenSSH.7.6p1	3.45%	0	2	7
Subtotal		10.34%	0	6	9

Table 4.3: Statistical summary of CVE count

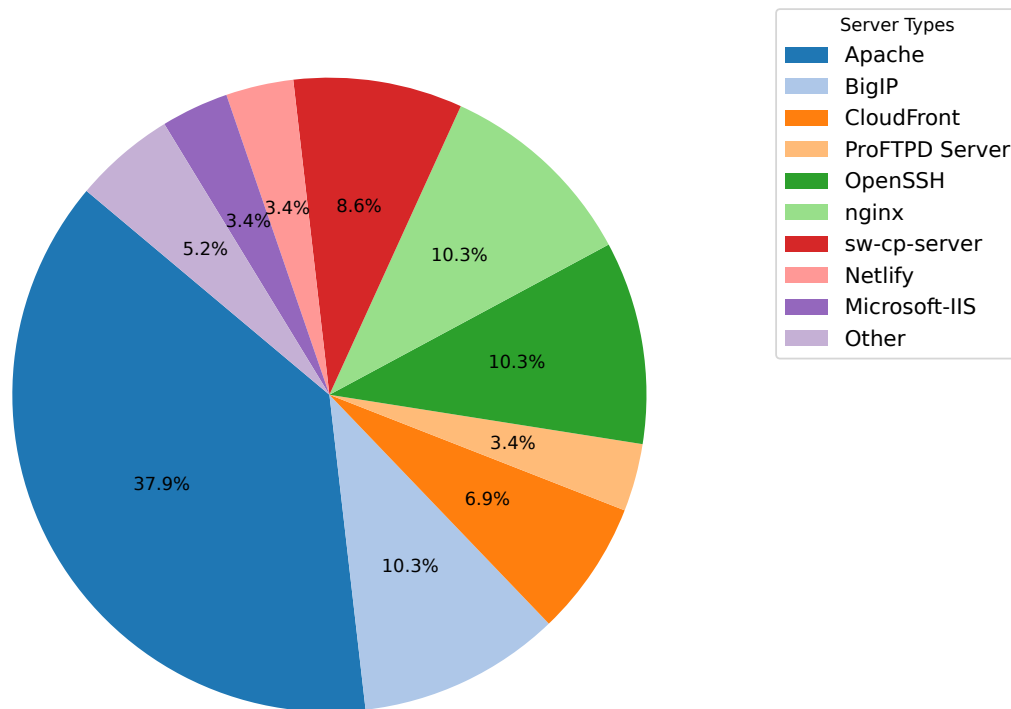


Figure 4.4: Server types distribution

of HTTP requests, header truncation, and HTTP Request Smuggling attacks. However, it is important to acknowledge that the presence of CVEs does not automatically indicate that they are exploitable, as our tool does not evaluate any defense mechanisms in place. The exploitation of these vulnerabilities also depends on other factors, such as specific server configurations, implemented security measures, and applied security patches.

Nonetheless, it remains imperative to pay attention to the existence of such CVEs within the purview of the public Internet. While their exploitability depends on different factors, being aware of these vulnerabilities enables organizations to assess risks, take necessary measures to reduce them, and actively manage vulnerabilities to improve system security.

CPE	CVE ID	Criticality	Description
cpe:2.3:a:apache: :http_server:2.4.54 :*.~*.~*.~*.~*.~*	CVE-2023-25690	CRITICAL - 9.8	A vulnerability in Apache HTTP Server allows an HTTP Request Smuggling attack due to certain mod_proxy configurations.
	CVE-2022-36760	CRITICAL - 9.0	Inconsistent interpretation of HTTP requests vulnerability in Apache HTTP Server's mod_proxy_ajp allows an attacker to smuggle requests to the forwarded AJP server.
	CVE-2006-20001	HIGH - 7.5	A carefully crafted If: request header in Apache HTTP Server 2.4.54 and earlier can cause a memory read or write, potentially leading to a process crash.
	CVE-2023-27522	HIGH - 7.5	Apache HTTP Server via mod_proxy_uwsgi is vulnerable to HTTP Response Smuggling.
	CVE-2022-37436	MEDIUM - 5.3	Prior to Apache HTTP Server 2.4.55, a malicious backend can truncate response headers, causing some headers to be incorporated into the response body, potentially bypassing security purposes

Table 4.4: Gathered Vulnerability details for Apache/2.4.54

Chapter 5

Discussion

In this thesis, we developed a research workflow to map the attack surface of telecommunication networks using passive techniques. We further automated the workflow and developed a tool for this purpose. Anyone with access to the Internet can use this tool to map the attack surface of the telecommunications operator. Telecommunications operators can use this tool to conduct audits and gain an external perspective on their network's security. Additionally, this tool can be leveraged by cybersecurity enthusiasts and researchers to gain a better understanding of telecommunications operators' network architecture.

Due to the use of passive techniques, the tool possesses inherent limitations regarding the results it can gather. Nevertheless, as the tool is built with passive reconnaissance tools and follows a modular architecture, the tool is flexible and can be extended, leaving room for future enhancements and research in the field.

5.1 Limitations

During the implementation of this thesis, several technical limitations were encountered and are worth discussing. While developing our tool, we leveraged several passive reconnaissance tools, including MxToolBox, Censys, Shodan, and Sublist3r. Primarily, limitations were observed with the APIs of Censys and MxToolBox. Censys, a data collection tool, limits free accounts to 250 API requests per month, and MxToolBox, another data collection tool, limits free accounts to 68 requests per day. To overcome this limitation, we extended the data collection over weeks and days until we gathered enough data that allowed further analysis and evaluation. Nonetheless, users of the tool have the option of purchasing the paid version of these APIs and configuring their API keys in the *apikey.sh* file for Shodan, Censys, and MxToolBox, allowing for more comprehensive results.

Furthermore, it is essential to acknowledge the limitations of the employed

methodology and the obtained results. The focus of this thesis was on the use of existing passive tools and techniques to map the attack surface of telecommunications operators. Thus, the gathered results predominantly pertain to generic network assets of mobile network operators and may not be specific to telecommunications infrastructure. This limitation arises because existing tools primarily target generic network assets and do not extensively scan for telecommunications-specific ports and protocols. Active scanning methodology can be used to scan for telecommunications-specific ports and protocols to collect more information regarding telecommunications infrastructure.

In addition, there is a limitation associated with the list of Common Vulnerabilities and Exposures (CVEs) compiled from our tool, as it does not guarantee their exploitable nature. The exploitability and severity of these vulnerabilities depend on specific server configurations and the defense mechanisms in place. Passive techniques cannot assess the existence of defense mechanisms, such as firewall policies set up by the operators. Although we retrieve the list of server versions and their associated CVEs from our tool, we lack information regarding the server configurations, which typically necessitates direct interaction with the server, i.e., active reconnaissance techniques. Therefore, while the results provide insights into potential risks and assist in risk assessment, further analysis using active techniques is necessary to evaluate the exploitability of vulnerabilities.

Another limitation encountered throughout this thesis is the scarcity of publicly available information regarding the configuration and deployment of telecommunications infrastructure. For instance, databases administered by GSMA, such as the IR.21 database, which contains standardized details about network infrastructure and interconnection for each operator, are usually limited to authorized individuals within the telecommunications industry. This limitation restricts the scope of queries formulated using search engines. Therefore, it is crucial to foster collaboration and information exchange among industry stakeholders and researchers to achieve a more comprehensive understanding of the telecommunications network infrastructure.

5.2 Future Directions

To further enhance the capabilities and effectiveness of our tool for mapping the attack surface of telecommunication operators, we propose several extensions and improvements. These future directions aim to overcome the limitations identified in our research and provide a more comprehensive assessment of telecommunication network security.

Active scanning methodology: In our research study thus far, our investigation of the telecommunication operator has relied only on the use of publicly

accessible resources, and the methodology adopted is passive reconnaissance, and we relied on the banners grabbed from Shodan and Censys. This thesis work does not include active scanning for ports. Active scanning can be employed using several network and port scanning tools mentioned in Section 2.3.1 to scan for telecommunications-specific ports and protocols presented in Table 5.2.

Protocols	Port
Radius	1812
Stream Control Transmission Protocol (SCTP)	9899
Diameter	3868
Signaling System 7 (SS7)	2904
GPRS Tunneling Protocol for Control Plane (GTP-C)	2123
GPRS Tunneling Protocol for User Plane(GTP-U)	2152

Table 5.1: List of telecommunications-specific ports and protocols

To address this, we can incorporate such as Masscan into our proposed tool to search for these specific ports and protocols. The Masscan command can be used as shown below. The target list file will be generated by our tool, containing all the IP addresses associated with the specific telecommunications operator.

```
masscan -p <port> --protocol <protocol> -iL <target_list_file>
```

However, it is important to note that existing tools such as Massscan may not have built-in support for telecommunications-specific ports and protocols. To overcome this limitation, the use of Scapy can be explored, as it allows for the generation of custom packets and enables tailored active scanning for telecommunication networks.

Integration of Additional Tools: While the present thesis concentrates on a specific set of tools for assessing network assets, this analysis can be expanded by integrating additional tools that cover a broader range of assets. This expansion can be achieved by integrating other tools that target areas beyond network assets, such as Spiderfoot and Maltego, as discussed in Section 2.3.2.5

Evaluation of Defense Mechanisms: The existing methodology relies solely on passive techniques, which does not allow us to explicitly evaluate the defense mechanisms used by telecommunications operators. To address this limitation, we propose an extension of the tool to assess potential defense mechanisms, including firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS). Through the use of tools like Nmap, we can verify the existence of firewalls and other protective measures that can impact the attack surface [52]. Furthermore, the Metasploit framework can be leveraged to develop, test, and

execute exploit code, enabling a comprehensive examination of security vulnerabilities [53]. Therefore, the incorporation of active scanning techniques using tools like Nmap and Metasploit can aid in identifying and evaluating potential defense mechanisms.

HCI-Based Methods: As previously mentioned in the limitations, the lack of insider information regarding the deployment of telecommunications infrastructure presents challenges to the methodology. To overcome this limitation, conducting interviews with network engineers responsible for these systems is proposed as a potential area for improvement. Engaging with industry experts would enable the collection of valuable insider knowledge and insights into the deployment and configuration of telecommunications infrastructure. This approach complements the technical analysis and facilitates a more comprehensive understanding of the attack surface.

Chapter 6

Conclusions

In this thesis, we proposed a methodology to map the network attack surface of telecommunications operators using passive techniques. We provided an overview of network assets and passive reconnaissance tools. Additionally, we conducted a thorough literature review, revealing a research gap in mapping attack surfaces for telecommunications operators. To address this gap, we developed a research workflow to streamline our proposed methodology and developed a Python-based command line tool to automate the process. To evaluate the tool, we ran it for a mobile network operator and examined the results. The tool effectively gathered information about network assets, such as subdomains, DNS records, IP addresses, and server version details. It also identified potential vulnerabilities in the network infrastructure of mobile network operators. These results helped us identify test subdomains, third-party services from DNS records, exposed services and ports, and the server and operating system versions used by the operator. Additionally, the results highlighted potential vulnerabilities associated with the used servers.

It is important to note that our tool did not evaluate the defense mechanisms in place, and we did not conduct any exploits. Therefore, the exploitable nature of these vulnerabilities cannot be guaranteed. Furthermore, our tool primarily focused on generic network assets rather than being specific to telecommunications infrastructure. While it offered valuable insights into enterprise network assets, future research could explore a more granular view of telecommunications-specific infrastructure.

Our developed tool is flexible and allows for easy extension and adaptation, making it a versatile asset for future enhancements and research in the field. We anticipate our work will inspire further advancements in telecommunication network security and open doors for academic research in a traditionally closed industry.

Bibliography

- [1] CYWARE, “Telecom sector comes under attack as new APT groups emerge.” CYWARE Labs, Oct 2021. <https://cyware.com/news/telecom-sector-comes-under-attack-as-new-apt-groups-emerge-a2814fb9> Accessed 13.07.2023.
- [2] TelcoDR, “Telcos are moving to public cloud,” 2021. <https://www.telcodr.com/15-examples-of-how-telcos-are-moving-to-the-public-cloud/> Accessed 13.07.2023.
- [3] S. M. Bellovin, “Attack Surfaces,” *IEEE Security & Privacy*, vol. 14, no. 3, pp. 88–88. IEEE, 2016.
- [4] W. Mazurczyk and L. Caviglione, “Cyber reconnaissance techniques,” *Commun. ACM*, vol. 64, no. 3, pp. 86–95. Association for Computing Machinery, Feb 2021.
- [5] International Organization for Standardization and International Electrotechnical Commission, “ISO/IEC 19770-1: Information Technology - Software Asset Management - Part 1: Processes.” <https://www.iso.org/obp/ui/#iso:std:iso-iec:19770:-1:ed-3:v1:en>. Accessed 13.07.2023.
- [6] National Institute of Standards and Technology, “Asset.” <https://csrc.nist.gov/glossary/term/asset>. Accessed 13.07.2023.
- [7] S. Hazra, “Development of an IT asset management tool for enterprise information system,” *Asian Journal for Convergence in Technology (AJCT) ISSN -2350-1146*, vol. 4, Apr. 2018.
- [8] J. Li and Microsoft Threat Intelligence Center (MSTIC), “Gather Victim Network Information: DNS.” MITRE ATT&CK, Oct 2022. <https://attack.mitre.org/techniques/T1590/002/> Accessed 13.07.2023.
- [9] D. Liu, S. Hao, and H. Wang, “All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records,” in *Proceedings of*

- the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16*, p. 1414–1425, Association for Computing Machinery, 2016.
- [10] O. v. der Toorn, R. van Rijswijk-Deij, T. Fiebig, M. Lindorfer, and A. Sperotto, “TXTing 101: Finding Security Issues in the Long Tail of DNS TXT Records,” in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 544–549, 2020.
- [11] M. Vermeer, J. West, A. Cuevas, S. Niu, N. Christin, M. Van Eeten, T. Fiebig, C. Ganan, and T. Moore, “SoK: A Framework for Asset Discovery: Systematizing Advances in Network Measurements for Protecting Organizations,” in *2021 IEEE European Symposium on Security and Privacy (EuroSP)*, pp. 440–456, IEEE, 2021.
- [12] “Autonomous System Numbers.” <https://www.arin.net/resources/guide/asn/>. Accessed 13.07.2023.
- [13] MITRE ATT&CK, “Gather victim network information: IP addresses.” <https://attack.mitre.org/techniques/T1590/005/>, 2021. Accessed 13.07.2023.
- [14] “Nmap: the Network Mapper.” <https://nmap.org/>. Accessed 13.07.2023.
- [15] R. D. Graham, “ZMap: The Internet Scanner.” <https://github.com/zmap/zmap>, 2023. Release- 3.0.0, Accessed 13.07.2023.
- [16] R. D. Graham, “MASSCAN: Mass IP port scanner.” <https://github.com/robertdavidgraham/masscan>, 2023. Release- 1.3.2, Accessed 13.07.2023.
- [17] Z. Durumeric, M. Bailey, and J. A. Halderman, “An internet-wide view of internet-wide scanning,” in *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 65–78, 2014.
- [18] Greenbone, “OpenVAS.” <https://github.com/greenbone/openvas-scanner>, 2023. Release - v22.7.1, Accessed 13.07.2023.
- [19] Greenbone, “Nikto.” <https://github.com/sullo/nikto>, 2023. Accessed 13.07.2023.
- [20] “Tenable - Nessus.” <https://www.tenable.com/products/nessus>. Accessed 13.07.2023.
- [21] S. Jamieson, “Ethics and legality of port scanning.” GIAC (Global Information Assurance Certification) Paper, 2001. Accessed 13.07.2023.

- [22] J. Faircloth, *Penetration Tester's Open Source Toolkit*. 3 ed., 7 2011. Accessed: 2023-05-31.
- [23] B. Akhgar, P. S. Bayerl, and F. Sampson, *Open Source Intelligence Investigation from strategy to implementation*. Springer-Verlag, 2016.
- [24] "OSINT framework." <https://osintframework.com/>. Accessed 13.07.2023.
- [25] J. Pastor-Galindo, P. Nespoli, F. Gomez Marmol, and G. Martinez Perez, "The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends," *IEEE Access*, vol. 8, pp. 10282–10304. IEEE, 2020.
- [26] R. Radvanosky and J. Brodsky, "Project Shine (SHodan INtelligence Extrac-tion) Findings Report," in *10th SANS ICS Security Summit*, (Orlando, FL), 2 2015. Accessed: 2023-05-31.
- [27] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A Search Engine Backed by Internet-Wide Scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pp. 542–553, Association for Computing Machinery, 2015.
- [28] V. Troia, *Hunting Cyber Criminals*. Wiley, 2 2020.
- [29] A. Aboul-Ela, "Sublist3r." <https://github.com/aboul31a/Sublist3r>, 2020. Release 1.1, Accessed 13.07.2023.
- [30] Nmmapper, "Dnsdumpster." <https://github.com/nmmapper/dnsdumpster>, Feb 2023. Accessed 13.07.2023.
- [31] "IR.21 GSM Association Roaming Database, Structure and Updating Proce-dures." <https://www.gsma.com/security/resources/ir-21-gsm-association-roaming-database-structure-and-updating-procedures-v12-0>. Accessed 13.07.2023.
- [32] "IR.67 - DNS/ENUM Guidelines for Service Providers GRX/IPX Providers 4.1." <https://www.gsma.com/iot/wp-content/uploads/2012/03/ir6741.pdf>. Accessed 13.07.2023.
- [33] "National Vulnerability Database (NVD)." <https://nvd.nist.gov>. Ac-cessed 13.07.2023.
- [34] B. Cheikes, D. Waltermire, and K. Scarfone, "Common Platform Enumera-tion: Naming Specification Version 2.3." NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, Aug 2011.

- [35] S. Micallef, “SpiderFoot.” <https://github.com/s0md3v/ReconDog>, 2023. Release - v4.0, Accessed 13.07.2023.
- [36] S. Sangwan, “ReconDog.” <https://github.com/s0md3v/ReconDog>, 2019. Release 2.0, Accessed 13.07.2023.
- [37] “Maltego.” <https://www.maltego.com/downloads/>. Accessed 13.07.2023.
- [38] “Banner grabbing.” <https://www.geeksforgeeks.org/what-is-banner-grabbing/>. Accessed 13.07.2023.
- [39] R. Bodenheimer, J. Butts, S. Dunlap, and B. Mullins, “Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices,” *International Journal of Critical Infrastructure Protection*, vol. 7. Elsevier BV, 2014.
- [40] Y. Chen, X. Lian, D. Yu, S. Lv, S. Hao, and Y. Ma, “Exploring Shodan from the Perspective of Industrial Control Systems,” *IEEE Access*, vol. 8, pp. 75359–75369. IEEE, 2020.
- [41] I. Alsmadi, Z. Dwekat, R. Cantu, and B. Al-Ahmad, “Vulnerability assessment of industrial systems using Shodan,” *Cluster Computing*, vol. 25, pp. 1563–1573, 6. Springer International Publishing, 2022.
- [42] K. Simon, “Vulnerability Analysis Using Google and Shodan,” in *Cryptography and Network Security* (S. Foresti and G. Persiano, eds.), (Cham, Switzerland), pp. 725–730, Springer International Publishing, 2016.
- [43] A. Albataineh and I. Alsmadi, “IoT and the Risk of Internet Exposure: Risk Assessment Using Shodan Queries,” in *2019 IEEE 20th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoW-MoM)*, pp. 1–5, IEEE, 2019.
- [44] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker, and H. Chen, “Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT),” in *2014 IEEE Joint Intelligence and Security Informatics Conference*, pp. 232–235, Springer, 2014.
- [45] B. Genge and C. Enăchescu, “ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services,” vol. 9, pp. 2696–2714, John Wiley and Sons Inc., 10 2016.
- [46] “ShoBeVODSDT: Shodan and Binary Edge based vulnerable open data sources detection tool or what Internet of Things Search Engines know about you,” pp. 38–45, IEEE, 2021.

- [47] T. Ashley, S. N. G. Gourisetti, N. B. Brown, and C. A. Bonebrake, “Aggregate attack surface management for network discovery of operational technology,” *Computers & Security*, vol. 123, p. 102939. Elsevier Advanced Technology Publications, 2022.
- [48] G. Guo, J. Zhuge, M. Yang, G. Zhou, and Y. Wu, “A Survey of Industrial Control System Devices on the Internet,” in *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, pp. 197–202, IEEE, 2018.
- [49] “Mxtoolbox.” <http://mxtoolbox.com/default.aspx>. Accessed 13.07.2023.
- [50] “Shodan.” <https://www.shodan.io/>. Accessed 13.07.2023.
- [51] Vehemont, “nvdlib.” <https://github.com/vehemont/nvdlib>, 2023. Accessed 13.07.2023.
- [52] M. Wolfgang, “Host Discovery with nmap,” vol. 1, p. 16, 2002. <https://nmap.org/docs/discovery.pdf> Accessed 13.07.2023.
- [53] “The Metasploit Framework.” <https://www.metasploit.com/>. Repository accessible at <https://github.com/rapid7/metasploit-framework> Accessed 13.07.2023.