NTNU
Norwegian University of
Science and Technology

Improving SS7 Security Using Machine Learning Techniques

Kristoffer Jensen

01-06-2016

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Norwegian University of Science and Technology, 2016

Supervisor 1: Associate Professor André Årnes, NTNU
Supervisor 2: Dr. Do Van Thanh, Telenor Research

Preface

This master's thesis was completed during the spring of 2016 at the Norwegian University of Science and Technology (NTNU), Gjøvik. The thesis was initiated by and completed in cooperation with the Telenor Group. It was completed as part of the Information Security program on the technology track. The intended audience for this thesis is telecommunication operators wishing to secure the operation of their Signaling System No. 7 networks.

01-06-2016

Acknowledgments

I would like to express my sincerest gratitude towards my main supervisor Dr. André Årnes for starting me out on this project and pushing me towards greater goals. A huge thanks to Dr. Do Van Thanh for providing detailed knowledge on the topic and for providing excellent guidance. Thanks to Dr. Hai Thanh Nguyen for great discussions and providing me with the necessary boost and knowledge to achieve the results in this thesis.

Thanks to fellow students Jonas Taby and Simen Steig, who have provided interesting discussions and guidance during the work on the master's thesis. Finally, thanks to my family and my girlfriend who have provided me with the necessary support during these months. This is for you.

K.J.

Abstract

The Signaling System No. 7 is the nervous system of telecommunication networks based on 2G and 3G technologies. Previously confined in a walled garden, SS7 has become more exposed due to increased liberalization of the market in conjunction with the industry switching to IP technology. In the walled garden of trusted operators, security have received minimal attention. SS7 has become more vulnerable in the recent years, with attackers exploiting network communications to track subscribers, intercept calls, perform denial of services, and commit fraud. This master thesis is a part of the effort to reduce the vulnerabilities contained in the old, yet crucial protocols that the telecommunication operators cannot function without. Subscribers, operators, and national governments are dependent on one of societies critical infrastructures, it needs to be adequately protected.

In this thesis, a detailed overview of SS7 threats and vulnerabilities is presented. In an effort to mitigate these attacks, open source technology has been used to simulate network traffic. This generated traffic were used to analyse and detect attacks against SS7 in an effort to propose detection mechanisms. Machine learning, big data, and anomaly detection techniques have been used as tools in order to propose an improved online protection system for SS7 networks. The results achieved in this master's thesis have been submitted in the form of a paper to the International Conference on IT Convergence and Security 2016, Appendix A presents the submitted paper in its current form.

Contents

Acronyms

3GPP The 3rd Generation Partnership Project.

A-NADS Anomaly-Based Network Abuse Detection System.

API Application Program Interface.

CN Core Network.

DoS Denial of Service.

EIR Equipment Identity Register.

FPR False Positive Rate.

GSM Global System for Mobile Communications.

HLR Home Location Register.

IDS Intrusion Detection Systems.

IMEI International Mobile Station Equipment Identity.

IMSI International Mobile Subscriber Identity.

IP Internet Protocol.

LAC Location Area Code.

MAP Mobile Application Part.

MCC Mobile Country Code.

MNC Mobile Network Code.

MS Mobile Station.

MSC Mobile Switching Center.

MSISDN Mobile Station PSTN/ISDN Number.

NIDS Network Intrusion Detection Systems.

PSTN Public Switched Telephone Network.

S-H-ESD Seasonal Hybrid Extreme Studentized Deviate.

SCP Signal Control Point.

SIGTRAN Signaling Transport.

SMS Short Message Service.

SMSC Short Message Service Centre.

SP Signaling Point.

SS7 Signaling System No. 7.

SSP Signal Switching Point.

STP Signal Transfer Point.

UMTS Universal Mobile Telecommunications System.

USSD Unstructured Supplementary Service Data.

VLR Visitor Location Register.

List of Figures

List of Tables

1 Introduction

This master's thesis covers Signaling System No. 7 (SS7), the nervous system of telecommunication networks, with a focus on mobile telecommunication networks. More specifically, the thesis focuses on identifying and understanding the vulnerabilities of SS7 with the goal of proposing an innovative protection solution based on machine learning and data mining techniques.

SS7 is used for signaling in the Core Network (CN) for both the Global System for Mobile Communications (GSM) and the Universal Mobile Telecommunications System (UMTS)[1]. It is used primarily for setting up and tearing down calls, Short Message Service (SMS), and general information exchange in the CN. SS7 is a necessity in today's most used mobile telecommunication technology [1], and it simply cannot function without it. Originally developed in the nineteen-seventies [2], the protocols were created in another era and are starting to show their age.

Previously, SS7 was protected by the walled garden with minimal needs for security. Deregulation and the industries' continuous move to Internet Protocol (IP) technology, makes it easier to become an operator and also gain access to the confined SS7 network. In general, the SS7 network has been labeled as vulnerable and prone to exploitation by researchers and the media. Attackers are able to track phone users on a global scale, intercept calls and SMS messages, deny service to subscribers, and commit fraud. This thesis is part of the attempt to increase the understanding of the current vulnerabilities and attacks against SS7. This is done in an attempt to assist in the mitigation of the attacks on the networks and its subscribers.

Keywords

Telecommunications Security, SS7, Signaling Protocols, Machine Learning

1.1 Problem description and motivation

The Signaling System No. 7 (SS7) is a crucial component that the telecommunication networks cannot function without. In a closed network of trusted operators, it has not been necessary to provide extensive security measures to protect the network and its subscribers. Due to recent deregulation making it easier for anybody to become an operator and with the transition to the Internet Protocol (IP), SS7 has become exposed to attacks which threatens the security and privacy of mobile subscribers, and the integrity of operators' networks. SS7's recent

[1] Commonly known as 2G and 3G respectively.

media attention has labeled the networks as being insecure and easy to exploit [3, 4, 5]. Adversaries are potentially able to track phone users' location on a global scale, perform wire tapping, redirect phone calls and deny service to subscribers. SS7 stands as a pillar to support one of societies' critical infrastructures. In light of recent events and future development, additional measures to secure and protect the networks and users of telecommunication are needed.

As SS7 is the nervous system of the mobile communication networks it needs to be sufficiently protected. From the subscriber's viewpoint, mobile communication should be reliable, secure, and free of risk in daily use. Subscribers should not feel insecure when making calls or sending messages. From the operator's viewpoint, the daily operation should run smoothly without fraudulent incidents, which can incur extra cost. From the government's viewpoint, the mobile network is a critical infrastructure supporting emergency services for the population.

There has been an increasing amount of research done on SS7 security from 2008 and onwards [6, 7, 8, 9, 10]. Researchers and companies are disclosing critical attacks and entry points to the closed SS7 networks that threatens the privacy of subscribers and the integrity of telecommunication operators' network and operations. A continuing trend is the increase of disclosed vulnerabilities, but a lack of protection measures. There is a need to research specific mitigation measures for SS7 attacks, and this thesis is part of that work. In this thesis, an innovative protection methodology using machine learning is proposed. This method can help in the mitigation and detection of the disclosed attacks and vulnerabilities.

1.2 Research approach

A number of research questions was devised to approach the current problems with SS7. These research questions were meant to give insight into mitigation techniques for current SS7 vulnerabilities and attacks. Machine learning and data mining were researched as potential mitigation techniques and its feasibility stands as the main research topic for this master's thesis. The research questions used as the basis of this thesis were:

1. What are the vulnerabilities and weaknesses of SS7?
2. How can attacks on SS7 be prevented?
3. How can attacks on SS7 be detected using machine learning and data mining techniques?

1.3 Brief summary of results

This thesis provides a comprehensive and concise overview of SS7, and its vulnerabilities and threats are identified and explained. To mitigate the vulnerabilities and threats, the use of machine learning techniques was proposed to detect attacks against SS7.

As access to a real SS7 network and real SS7 data was infeasible, due to privacy and ethical concerns, the SS7 Attack Simulator was implemented to simulate a larger SS7 network. In these simulations, attacks and normal traffic was generated to develop a dataset containing SS7 network traffic. This dataset was used to prove the feasibility of using machine learning as a protective countermeasure to attacks in the SS7 network.

In brief, a set of carefully selected features was used as input to a number of machine learning algorithms in an effort to explore machine learnings feasibility. The data and the results indicate that machine learning techniques are a feasible approach to detect attacks in an SS7 network. Furthermore, a complete Anomaly-Based Network Abuse Detection System (A-NADS) was implemented based on these findings. The implemented A-NADS was used in conjunction with the simulator as an example of how an online detection approach can be implemented in a real network.

As a side effect of the results, the SS7 Attack Simulator can serve as part of a security testbed, which can be used to further research and study SS7 vulnerabilities and attacks. All implementations and techniques used in this master's thesis were released under a free license containing code and documentation [11].

Based on the methods used in this thesis, a paper was submitted to the 6th International Conference on IT Convergence and Security 2016 and is currently undergoing review. The submitted version can be viewed in its entirety in Appendix A.

1.4 Structure of the thesis

This thesis starts by setting the scene based on the current problems with SS7 in chapter 2. In this chapter, it is discussed what motivated this thesis and what the current researchers and companies are focusing on in regards to SS7 security. To provide the reader with the necessary fundamental knowledge to further understand the topic, an introduction to SS7 and the Core Network (CN) are provided in chapter 3. As well as giving an introduction to machine learning and the algorithms applied in this thesis.

After the background has been thoroughly explained, a technically detailed overview of the current known publicly disclosed attacks is provided to the reader in chapter 4. In this chapter, it is discussed what capabilities an attacker must have to be able to launch attacks. Furthermore, a detailed explanation of the attacks and how they unfold in the SS7 network is explained.

In chapter 5, a discussion on some approaches an operator may use to detect SS7 attacks is provided. The SS7 Attack Simulator is presented in this chapter, which was used to generate an appropriate dataset in order to test the feasibility of machine learning as an attack detection tool.

Chapter 6 explains the proof of concept using the dataset generated by the SS7 Attack Simulator. In this chapter, the Seasonal Hybrid Extreme Studentized

Deviate (S-H-ESD) algorithm is applied as part of an offline test, and the results using the algorithm is provided. Based on the offline test results, a fully functional online Anomaly-Based Network Abuse Detection System (A-NADS) for SS7 is presented in detail.

An overall discussion is presented in chapter 7. In this chapter, the overall approach of this thesis is presented. Including topics such as what concerns an operator must acknowledge when implementing an A-NADS and deploying it in a real SS7 network. Finally, chapter 8 provides an overview of some future working points based on this thesis and the current state of SS7.

2 Background

An increasing amount of attention has been given to Signaling System No. 7 (SS7) security in the recent years. The SS7 networks are in such a state that they have attracted media attention from newspapers such as Washington Post and Computer Weekly [3, 4, 5]. This chapter will highlight the importance of SS7, the Core Network (CN), and recent developments making attacks on SS7 possible.

2.1 SS7 weaknesses and increased attack surface

One of the first public reveals of SS7's weaknesses started with Tobias Engel's presentation at the Chaos Communication Club in 2008 [6]. In his findings, he demonstrated that with access to the SS7 network, an attacker could locate and track the movements of mobile telephone subscribers down to a regional level.

After this reveal, the presentations and demonstrations on SS7's weaknesses increased in number. Several businesses and research personnel have shown their concern with the current state of the protocols. In 2014, presentations made by Karsten Nohl [9] and again Tobias Engel [8] highlighted the critical state of SS7. Attacks resulting in tracking, fraud, interception of calls and texts, and denial of service were disclosed and demonstrated.

Several companies specializing in telecommunication security have come up with solutions and suggestions to how SS7 security problems should be solved. The people at P1 Security has made an attempt to track the severity of SS7 security issues in one of their projects labeled SS7map [12]. Their goal is to highlight vulnerable operators in countries around the world. Their approach focuses on the amount of privacy information leaked and the size of operators' attack surface based on exposed network elements. A screen dump of their project is presented in Figure 1. In addition, white papers have been published by the SANS institute [13], Positive Technologies [14], and AdaptiveMobile [15] on the current state of SS7 in promotion of the respective companies' products.

The next chapter will explore the technical implementation of telecommunication networks. With an emphasis on the 2G standard Global System for Mobile Communications (GSM) Core Network (CN), which is also used in the 3G standard Universal Mobile Telecommunications System (UMTS). After the fundamentals have been presented, some of the attack procedures as disclosed in [6, 9, 8] will be discussed in the following chapters. First, an historical introduction as to why SS7 security have received more attention recently.

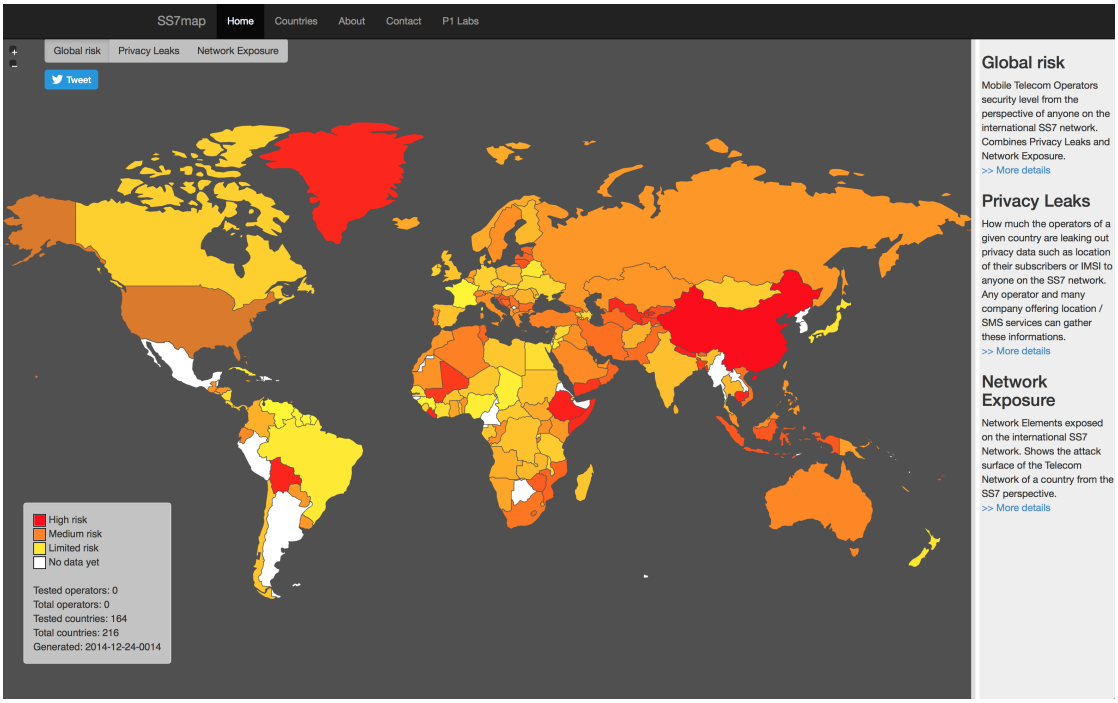


Figure 1: Screen shot of SS7map. An overview of the current state of mobile networks' SS7 security, by P1 Security [12].

2.2 What makes attacks possible?

There are a couple of historical events that makes SS7 more vulnerable and prone to attacks. More specifically, this includes the deregulation of the telecommunication sector and the industries' move to IP.

2.2.1 Deregulation of the telecommunication sector

In the past decades, the telecommunication sector has experienced a deregulation, or a liberalization of their markets. This change happened in both the United States and the European Union, in 1996 and 1998 respectively. The goal of liberalization was to foster competition and support the free market [16].

The deregulation on both continents have given room for smaller players to enter the market, instead of having a small number of larger dominant operators. These smaller companies now has access to the already existing infrastructure put in place by the larger companies [16]. An example of these smaller companies is for example a mobile virtual network operator (MVNO).

Deregulation have resulted in an increased amount of operators, and therefore increased accessibility to the closed SS7 networks. Access can for example be granted to an MVNO in order to provide additional services to subscribers by providing custom applications.

2.2.2 The industry is moving to IP

Traditionally, SS7 does not utilize the Internet Protocol (IP) to assist in transfer of messages and data over the network. But with the introduction of the Signaling Transport (SIGTRAN) protocols [2], and the introduction of the 4G technology LTE [17], the industry and standardization bodies are pushing towards IP. Using IP has several advantages as the technology is more available and also cheaper. It is therefore possible to utilize off the shelf hardware that cut costs for operators. In practice, the industry is merging together two communication arenas, bringing together both the positive and the negative aspects of both technologies. It is therefore uncertain what the security consequences of this merger are, but in combination with deregulation it certainly makes the closed SS7 networks more accessible.

3 Fundamentals

This master's thesis is based on fundamental principles within telecommunication systems, machine learning, and intrusion detection systems. In this chapter, the theoretical basis required to understand the topics in this master's thesis is presented to the reader.

3.1 Signaling System No. 7

Signaling System No. 7 (SS7) is a family of signaling protocols originally used in the Public Switched Telephone Network (PSTN). Standardized by The International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) in 1988, it is used in between the elements in the PSTN to exchange information. Primarily for setting up and tearing down phone calls, but is also used in billing, Short Message Service (SMS), routing and general information exchange between elements in the GSM and UMTS Core Network (CN)[18].

3.1.1 Signaling points and network architecture

Each node in an SS7 network must provide SS7 features which makes the node a Signaling Point (SP) in an SS7 network. Each SP in an SS7 network communicates with other nodes via data links, referred to as signaling links. There are three essential nodes in an SS7 network that are used to transfer signaling. Signal Switching Point (SSP) are the telephone switches of the network. These SPs are located at the end points of the network and perform functions such as originating, terminating or switching calls. Signal Transfer Point (STP) are the packet switches in the SS7 network. They route signaling messages in the network to their destination using specialized routing functions such as congestion control. Signal Control Point (SCP) provides additional information to the STPs to perform advanced call processing. These functions might include number translation in the case of special numbers [2].

To make sure service is provided with maximum uptime, the SS7 network provides several layers of redundancy in the network. A prime example is the fact that STPs and SCPs are usually deployed in pairs. Redundancy is also improved by using several signaling links per node. Figure 2 shows a typical SS7 network with nodes connected by signaling links with SSP nodes A-D, SCP nodes M-P, and STP nodes W-Z.

3.1.2 The SS7 protocol stack

There are several protocols used at different abstraction levels to transfer signaling information in an SS7 network, the SS7 protocol stack is shown in Figure 3.

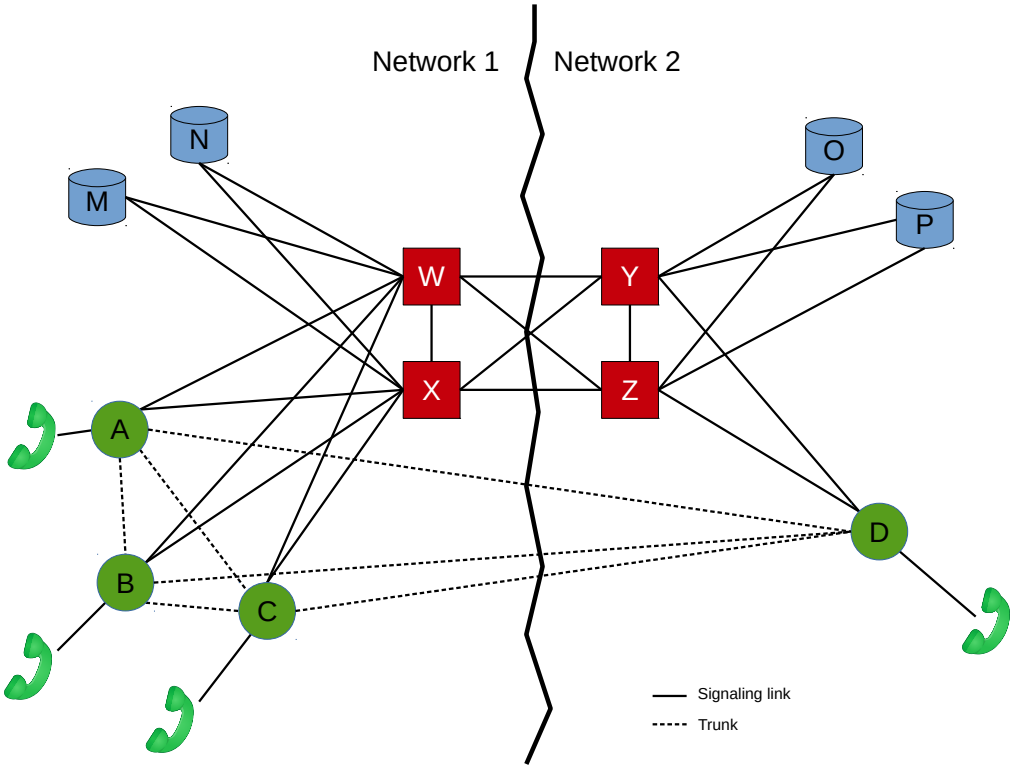


Figure 2: Overview of a typical SS7 network. Inspired by drawings in [19].

The different abstraction levels will be explained as follows starting from the bottom of the stack.

- *Message Transfer Part (MTP) Level 1 - 3:* The MTP is split into three parts where MTP1 is the physical layer, MTP2 is the data link layer, and MTP3 is the network layer. These layers have the main purpose of transferring information between SPs with functionality like reliable information transfer, error correction, and routing [2].
- *ISDN User Part (ISUP) and Telephony User Part (TUP):* The ISUP and TUP provides signaling functionality to initiate, maintain, and terminate calls. Both protocols use MTP to transfer messages [2].
- *Signaling Connection Control Part (SCCP):* The SCCP provides improved routing and transferring of data in the SS7 network. Which is used to interact with databases (SCPs), provide application management functions, and enhanced routing. Enhanced routing (referred to as global title (GT) routing) makes it unnecessary for every STP to handle large routing tables by assigning every SP with a GT that functions as an alias for a physical

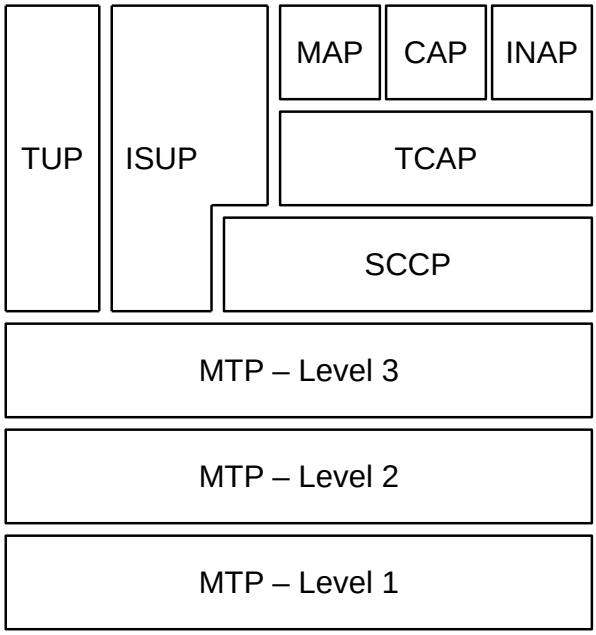


Figure 3: The SS7 protocol stack.

address [2].

- *Transaction Capabilities Application Part (TCAP):* The TCAP makes inter application (subsystem) communication possible in between SPs. It provides the necessary functionality for subsystems to send instructions between one another, which causes applications such as MAP, CAP, and INAP to function [2].
- *Mobile Application Part (MAP):* The MAP provides an application layer that is used by the various nodes in the Core Network (CN) to provide services to mobile subscribers. It provides functionality such as mobility management (roaming), SMS, and subscriber authentication [2]. MAP has been further extended by The 3rd Generation Partnership Project (3GPP) to support 3G networks [20].
- *CAMEL Application Part (CAP):* The CAP makes it possible to extend the services provided by the standard mobile networks, with the use of Customized Applications for Mobile networks Enhanced Logic (CAMEL). The CAP can be used to offer additional services to subscribers when roaming, for example improved telephone number translation [21, 22].

3.1.3 SIGTRAN (Signaling Transport)

SIGTRAN is an addition and compliment to the SS7 protocols developed by the Internet Engineering Task Force (IETF). It makes it possible to transfer SS7 over

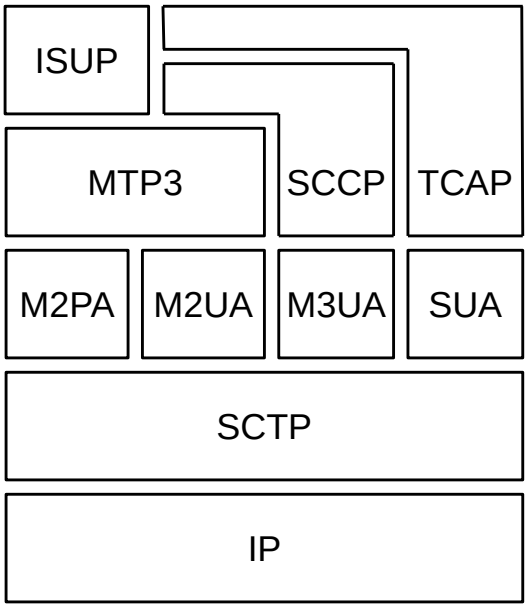


Figure 4: The SIGTRAN protocol stack. Inspired by drawings in [23].

.

IP networks by wrapping the signaling protocols into additional layers. SIGTRAN uses the same upper application layers as the original SS7 stack, but adds additional functionality in the lower layers that makes it possible to transfer the application parts over IP networks. The Stream Control Transmission Protocol (SCTP) is utilized because it has several advantages compared to the more often used TCP and UDP protocols for transporting signaling over IP [24]. SIGTRANs basic functionality involves several nodes in the network that translates packets into the original SS7 protocols and visa versa [25]. To transfer signaling over IP, SIGTRAN uses protocols such as MTP3 User Adaption Layer (M3UA) and MTP2 User Adaption Layer (M2UA) to reliably transfer signaling on IP networks using SCTP [26, 27] The SIGTRAN stack is shown in Figure 4.

3.2 The core network

In both GSM and UMTS mobile telecommunication networks (2G and 3G respectively), the Core Network (CN) provides the functionality and services necessary for serving the mobile subscribers connected to the network. In the CN, there are several networked elements crucial to its operation. These nodes are defined and standardized by The 3rd Generation Partnership Project (3GPP) [28]. Figure 5 shows an overview of the GSM Architecture containing some of the elements defined by 3GPP. When roaming in the network, the Mobile Station (MS) will

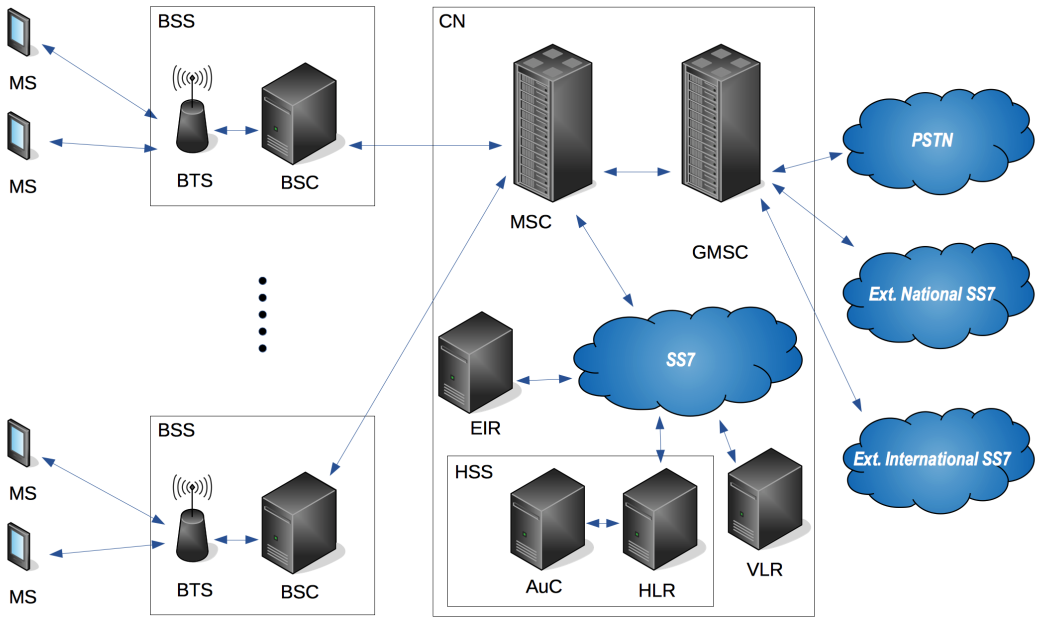


Figure 5: Overview of some of the components used in the 2G/3G infrastructure. Inspired by 3GPP TS 23.002 [28].

be connected to a Base Station Subsystem (BSS) containing a Base Transceiver Station (BTS) and a co-located Base Station Controller (BSC). The BSS handles the connection to and from the MS and further sends and receives information to the CN.

The different elements in the CN handle different tasks to provide services to subscribers. This includes handling location information, billing, authentication and tracking user locations such that calls and SMS can be routed to their equipment.

3.2.1 Mobile switching center

The Mobile Switching Center (MSC) handles the switching and signaling to and from a subscribers' Mobile Station (MS). Its main job is to route calls and SMS, and other functions such as handover operations when a subscriber is changing location during a call. Basically, the MSC is used to translate user-network signaling to network-network signaling. Another component with similar functionality is the Gateway MSC(GMSC), located at the border of the CN, that appropriately routes a call or SMS to an MS located in another network [28].

3.2.2 Home subscriber server

The Home Subscriber Server (HSS) serves as the main database for the operator's subscribers. In support of handling calls and other telephone activities,

the HSS handles subscription information. Located in the subscriber's home network, its main functionality lies in subscriber identification, network access control information, inter-system subscriber location information, and user profile information. There can exist several HSSs based on the number of subscribers served by the operator [28]. The HSS consists of two components:

- The Home Location Register (HLR), which handles roaming information such as where the subscriber is located at all times so that calls and SMS can be routed correctly.
- The Authentication Center (AuC), which helps to authenticate subscribers making an attempt to connect to the network. To assist in the authentication procedure, the AuC stores identity keys for each subscriber used in generating security data for authentication, integrity checks, and enciphered communication.

3.2.3 Visitor location register

The Visitor Location Register (VLR) controls a Mobile Station (MS) roaming in the area covered by an MSC and is usually co-located with the MSC. When a subscriber is roaming to a new MSC area, the VLR handles the registration procedure which includes exchange of information between the VLR and the subscriber's HLR. The VLR will inform the HLR of the subscriber's location, and will in return get information required to handle calls and other services. The VLR handles different elements such as the International Mobile Subscriber Identity (IMSI), used to identify a subscriber in the network, and the Mobile Station PSTN/ISDN Number (MSISDN) which is the subscriber's telephone number [28].

3.2.4 Equipment identity register

The Equipment Identity Register (EIR) contains information on handsets identified by their International Mobile Station Equipment Identity (IMEI). The IMEIs are stored in either a white, gray, or black list which can be used to identify and disallow service to stolen devices [28].

3.2.5 Short message service center

The Short Message Service Centre (SMSC), or more formally the SMS Service Centre (SMS-SC), is an entity that handles routing of SMS messages in the CN. It operates by querying routing information from an HLR, and routes the message to the appropriate SMSC or MSC in order to deliver an SMS to the intended subscriber [29].

3.2.6 Identifiers in the core network

To identify subscribers in the CN, 3GPP has defined a set of identification numbers to differentiate the subscribers and equipment connected to the CN.

- *International Mobile Subscriber Identification (IMSI):* Is used to identify a subscriber in the CN. The IMSI is a unique number identifying a USIM/SIM

card that is linked to a subscription. It is a 15 digit number containing the Mobile Country Code (MCC), Mobile Network Code (MNC), and a random set of numbers. This number is only used internally in the CN, and should not be known by anyone except the CN and the subscriber. Uses includes subscriber identification in the CN, and authentication of an MS connecting to a network [30].

- *Mobile Station International PSTN/ISDN number (MSISDN):* The MSISDN is basically the telephone number of a mobile subscriber. It is used to query routing information from an HLR when making a call or sending an SMS [30].
- *International Mobile Equipment Identification (IMEI):* The IMEI is used to identify the equipment used by a subscriber. It is a 15 digit number related to the origin, model, and serial number of the equipment. The IMEI is used in relation to the EIR, where it is used to blacklist potentially stolen devices [30].

3.3 Network intrusion detection systems

Intrusion Detection Systems (IDS) are software or devices used to monitor network or system events in order to detect malicious activities. Network Intrusion Detection Systems (NIDS) are specially designed systems made to detect remote attacks on a host or a network. They are strategically placed in networks to gather relevant network traffic that can be used to detect an attack that has happened, or one that is on the verge of being executed. There generally exists two types of NIDS, based on their method of detecting attacks: misuse detection and anomaly detection [31].

3.3.1 Misuse detection

Misuse detection systems detect attacks based on a previously known pattern related to the attack, also referred to as the attack *signature*. A NIDS using misuse detection will look for these signatures in the network traffic to identify an ongoing or upcoming attack. The signatures explains what to look for in the traffic, which may for example be a TCP packet containing a certain payload or a set of packets being sent in a specific order [31].

3.3.2 Anomaly detection

On the other hand, systems based on anomaly detection tries to assess what behavior is "normal" and look for deviations in the network activity. Typically a policy is created based on what is normal and the NIDS uses this policy to detect abnormalities in the network which are referred to as anomalies. Anomalies might for example include a sudden rise in network traffic, or a sudden rise in use of an application protocol [31].

The main goal of an anomaly detection approach is the detection of outliers. Typically an anomaly detection algorithm will be trained using normal data, it

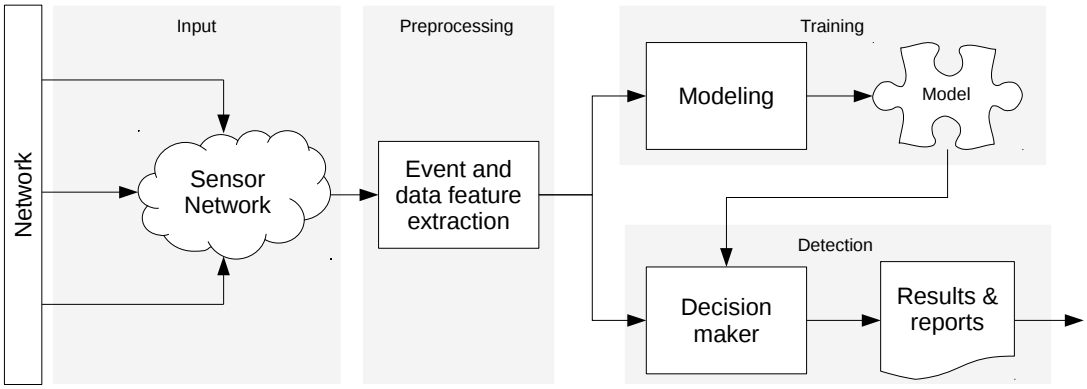


Figure 6: Architecture of a generic A-NIDS. Inspired by descriptions in [32]

.

is then the task of the algorithm to decide if input data is close to or far from the defined normal. The algorithm will look for activities defined as normal, and report on any deviation from normality. As anomaly detection is a central part of this thesis' approach to SS7 security problems, an introduction will be given to the techniques available to detect anomalies in a network.

3.3.3 Anomaly-based network intrusion detection systems

An anomaly-based network intrusion detection system (A-NIDS) is specifically created to detect anomalies in a networked environment. There exists many different approaches to anomaly detection but they all generally consists of the following three modules or stages [32], as shown in Figure 6:

- *Preprocessing stage*: In this stage, the interesting activities and events gathered by the sensor network is preprocessed into meaningful features that can be modeled into behavior metrics.
- *Training stage*: During the training stage, processed events is used to build a model that describes the normal behavior of the network.
- *Detection stage*: In this stage, the model created in the training stage is used to determine if the subsequent events are normal or not. The performance of this stage relies on the models ability to correctly represent normal behavior.

These stages are performed in two different modes that are usually performed separately: construction and detection. It must be assumed that networks and systems evolve, so that the model may need to be reconstructed to adapt to the new environment [32].

There are several approaches that can be used to detect anomalies in a network environment, many of these techniques rely on machine learning and data

mining. To paraphrase Dr. Nguyen in his PhD dissertation [33]:

> *Intrusion detection has been formulated as a statistical pattern recognition task, machine learning is the core to build these systems due to efficiency and effectiveness.*

Machine learning techniques are further used to provide a system with high flexibility and adaptability [34]. As machine learning and data mining techniques are relevant for the solution proposed by this thesis, a short introduction will be provided in the next section.

3.4 Machine learning and data mining

Machine learning is a subfield of computer science used for data analysis and knowledge acquisition. It can be applied as data mining on databases, be used in automatic generation of knowledge bases for expert systems, learning to plan, game playing, etc. The basis of machine learning is "the automatic modeling of underlying processes that have generated the collected data" [35]. In layman terms, machine learning aims to make sense of data, or in more formal terms as defined by Tom M. Mitchell [36]:

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.*

Which simply means that the if a computer program performs some task T, say playing chess. The programs ability to play chess can be determined by how many games it wins, which is the performance measure P. If by playing additional games of chess the computer program wins more games, it is said to have improved its performance based on its experience E. The program is said to have learned and therefore improved.

Data mining is used to discover patterns in larger data sets (commonly referred to as "big data") and uses methods derived from artificial intelligence, machine learning, statistics and database systems. The main goal of data mining is to extract information from a set of data and store it as a understandable structure for further use [37].

3.4.1 Machine learning basics

In machine learning, a system based on learning that wants to achieve a particular task is referred to as an *expert system*. The expert system holds knowledge which is crucial to making a decision, for example determining to which class an input belongs in the case of classification [38].

For each input, a set of measurements referred to as features is used by the learning algorithm to perform its task. Each set of features is called an *instance*,

and will typically be a row in a table containing the input data. Much like a variable, a feature can be of several types: for example an integer, a string or a binary value. These features can either be discrete or continuous [38].

As an example, classification is one of the tasks that machine learning is used to solve. In classification, an input is to be classified to a specific class which is part of a discrete set of classes. The machine learning algorithm will initially learn from a *training set* before being presented with new inputs that should be classified. A training set will contain several *training examples* containing features that is used by the classifier in the classification task. A training example will contain the features relating to a *target variable* which the features describe. For example a target variable **car** may be explained by the features: number of wheels (4), the model (Amazon) and the number of legal passengers (4). As opposed to a **motorcycle** which only has 2 wheels, a different model name and only 1 or 2 legal passengers [38].

There are four broad classes of machine learning: (1) supervised learning, (2) unsupervised learning, (3) semi-supervised learning, and (4) reinforcement learning. Supervised and unsupervised learning will be explained briefly in the following section as they are of relevance to this thesis.

Supervised learning

Supervised algorithms uses background knowledge and input data as its input to the learning algorithm. There are two tasks performed using supervised learning: classification and regression. Classification is used where a prediction is made as to which class an input belongs to, the output is a discrete variable. Regression outputs a numerical continuous variable. A supervised learning algorithm is basically told what to predict. The supervised algorithm's input is labeled data that is generated in conjunction with an expert who has the appropriate background knowledge required to perform the labeling [38].

Unsupervised learning

Unsupervised algorithms aims to find background knowledge automatically in the data. This is generically done by two methods: clustering and density estimation. An unsupervised algorithm is only fed unlabeled data. It is therefore the task of the algorithm to detect and find meaning and patterns in the data [38].

3.4.2 Performing anomaly detection using machine learning

Machine learning techniques can be used to detect anomalies in data. By anomalies we refer to the distinction of what is normal, and what is not. An anomaly detection algorithm basically tries to solve the problem of detecting outliers. Outliers are data points so far from the modeled normal behavior that they are considered abnormal [39]. A visual example of the distinction between normal and abnormal behavior can be seen in Figure 7.

There is a multitude of techniques available to perform anomaly detection

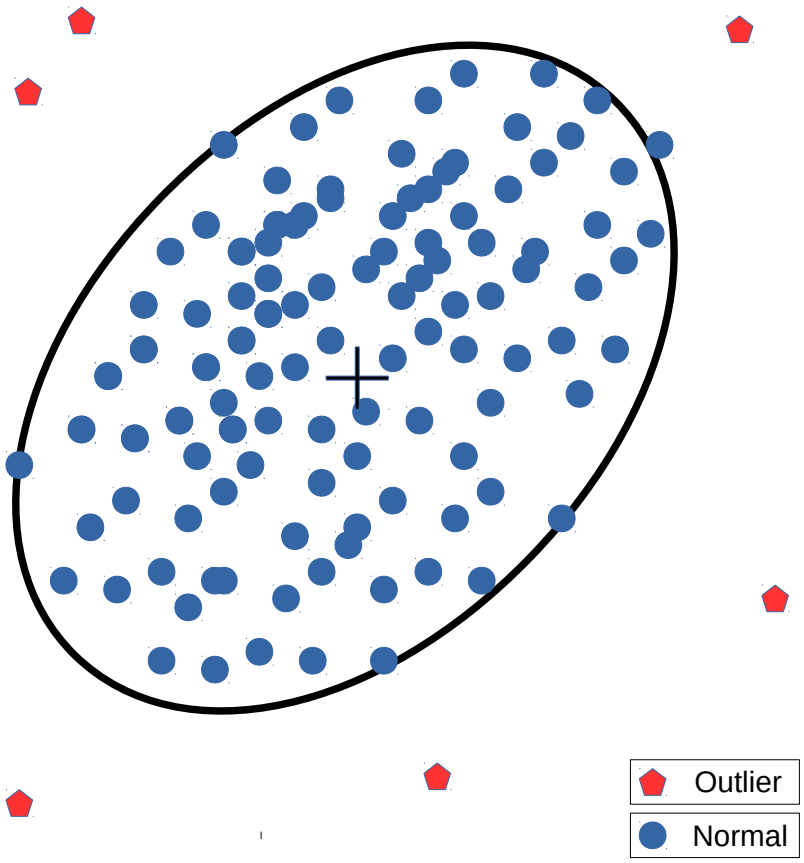


Figure 7: Visual example of outliers falling outside the definition of normal. Inspired from descriptions in [39].

[40], each providing their own pros and cons based on the dataset they are applied on. In this thesis, there are two forms of machine learning algorithms that have been used to solve the anomaly detection task: the Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD), based on the Generalized ESD, and k-means||, a version of the k-means clustering algorithm. These algorithms will be presented in full. More on why these algorithms where chosen and how they were used can be read about in section 6.1.

3.4.3 The Seasonal Hybrid Extreme Studentized Deviate Test

The Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) is an anomaly detection algorithm implemented by Twitter [41]. It is based on the Generalized ESD algorithm [42]. Twitter's AnomalyDetection library is implemented in R and available on GitHub [43].

The S-H-ESD builds on the ESD test, which can be used to detect a single

outlier in a dataset by finding the point furthest away from the mean of the dataset. By computing $G = \frac{|Y_i - \bar{Y}|}{s}$, where $\bar{Y}$ is the mean of the dataset and s is the standard deviation of the dataset. If G is larger than the critical value, the point is an outlier [42]:

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-1}}{N-2+t^2_{\alpha/(2N),N-2}}} \tag{3.1}$$

where $t^2_{\alpha/(2N),N-1}$ is the upper critical value of the t-distribution with $N-2$ degrees of freedom and a significance level of $\alpha/(2N)$.

To test for multiple outliers we can use the Generalized ESD algorithm, where it is assumed that there can be up to r outliers. The algorithm works by iterating the dataset and removing the point with the highest G value calculated from the dataset's new mean and standard deviation. The critical values λ will change with every removed point from the dataset [42]:

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i+1+t^2_{p,n-i-1})(n-i+1)}} \tag{3.2}$$

To decide whether a point is an anomaly or not, the following rule applies: if all of the test statistics are lower than the critical values, there are no anomalies. On the other hand, if any of the test statistics are greater than the critical value, the largest number of points so that the associated test statistic is greater than the critical value are removed as outliers [42].

The Generalized ESD algorithm assumes that the dataset is normally distributed [42], and as real data might include some seasonality it cannot directly be applied. The S-H-ESD algorithm solves this problem by applying R's Seasonal Decomposition of Time Series by Loess (STL) library. STL is used to decompose the data into a seasonal part, a trend part, and the remaining data using local regression (LOESS). LOESS fits a low order polynomial to a subset of the data and merge them together by weighing them. As the trend and seasonal part can be removed using LOESS, the remaining data will be close to normally distributed. Then the Generalized ESD can be applied on the remaining data to detect anomalies [43].

The S-H-ESD implementation splits data into chunks of length *period*, which is analyzed for a maximum number of anomalies *max_anoms*. The statistical significance used to accept or reject an anomaly is given with the option *alpha*, it is also possible to specify the direction in which anomalies should be detected using the *direction* option.

3.4.4 The k-means|| clustering algorithm

The k-means algorithm is an unsupervised clustering technique, popularized in data mining applications [40]. Formally, the technique involves solving the k-means problem given the integer k and a set of n data points $X \subset \mathbb{R}^d$. With the goal of choosing k centers C so the potential function is minimized:

$$\phi = \sum_{x \in X} \min_{c \in C} ||x - c||^2 \tag{3.3}$$

Using these centers, clustering can be done by grouping data points together according to which center each point is assigned to [44]. The original k-means algorithm is denoted in Algorithm 1.

Algorithm 1 The original k-means algorithm, from [44]

1: Arbitrary choose an initial k centers $C = c1, c2, \ldots, c_k$.
2: For each $i \in 1, ..., k$, set the cluster C_i to be the set of points in X that are closer to the c_i than they are to c_j for all $j \neq i$.
3: For each $i \in 1, ..., k$, set c_i to be the center of mass of all points in $C_i : c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
4: Repeat Steps 2 and 3 until C no longer changes.

There are several variants of the k-means algorithm that both improves performance and scalability. A notable example is the k-means++ algorithm, that proposes an improved cluster initialization, therefore improving the performance of k-means [44]. Building upon the k-means++ algorithm, the k-means|| algorithm has been proposed to further improve on the scalability of k-means by addressing its sequential nature. These improvements makes it easier to apply k-means on larger amounts of data. The k-means|| algorithm is the version of k-means used in this thesis, and it is implemented in the Apache Spark machine learning library [45]. The k-means|| algorithm is fully shown in Algorithm 2.

Algorithm 2 The k-means|| algorithm, from [46]

1: $C \leftarrow$ sample a point uniformly at random from X
2: $\psi \leftarrow \phi_X(C)$
3: **for** $O(\log \psi)$ times **do**
4: $\quad C' \leftarrow$ sample each point $x \in X$ independently with probability $p_x = \frac{\ell \cdot d^2(x,C)}{\phi_X(C)}$
5: $\quad C \leftarrow C \cup C'$
6: **end for**
7: For $x \in C$, set w_x to be the number of points in X closer to x than any other point in C
8: Recluster the weighted point in C into k clusters

4 Analysis of SS7 Vulnerabilities

There have been large amounts of disclosed attacks and vulnerabilities discovered by both researchers and companies on the vulnerabilities of SS7. In this chapter, a detailed overview of a number of publicly disclosed attacks against SS7 will be provided.

4.1 Security in the core network

SS7 is now more vulnerable than ever [8, 9]. Directly affecting the security and privacy of subscribers connected to the mobile networks, and the integrity of operators wishing to provide the best service to their subscribers. Engel and Nohl have discovered several attacks making it possible to exploit the vulnerabilities of the SS7 networks [8, 9]. These attacks are all based on the same premises: (1) all messages in the network are legal, (2) no sophisticated equipment is needed, and (3) escalation of attacks is possible with simple steps [14]. One of the larger underlying problems in SS7 network security is the increased attack surface. The networks were previously relying on the walled garden, meaning that all interconnected operators were trusted, this is no longer the case [7, 10].

The vulnerabilities of SS7 and attacks against mobile networks have mostly been disclosed by Tobias Engel [6, 8], Karsten Nohl [9], and Phillipe Langlois [7]. These attacks mostly utilize the SS7 MAP protocol to acquire staging information and in the execution of attacks. In addition to the presentations performed by researchers, a number of white papers has been issued that cover these attacks and further provides some mitigation measures [14, 13, 15]. A master's thesis has also recently been completed by Rao on the security of the CN [47].

The attacks disclosed by Engel, Nohl, and Langlois includes the possible tracking of mobile users down to regional or street level, denial of service, interception of calls and SMS, and fraud to avoid billing and to gain financial benefits. Messages are sent to relevant CN elements to successfully perform the attacks [6, 8, 9, 7]. The attacks discovered by these researchers and corporations will be presented in detail in the following sections.

4.2 Required capabilities of the attacker

To exploit the disclosed vulnerabilities of SS7, the attacker must posses certain capabilities as explained in a white paper by Positive Technologies [14]. The attacker must be: (1) connected to the SS7 network in some manner, (2) able to generate arbitrary SS7 messages at will, and (3) able to imitate a node in the SS7 network by providing SS7 capabilities.

4.2.1 Gaining access to the SS7 network

Arguably one of the more important capabilities of the attacker is actually gaining access to the closed SS7 networks. This problem is discussed by Langlois in one of his presentations [7], identifying several approaches an attacker can use to connect to the SS7 network. Still, SS7 is not a publicly accessible network and it is tightly controlled by the worlds telecommunication operators. Overall, in terms of how SS7 has evolved over the years, there has been an increase in the number of operators and the services they provide using SS7. This has in turn increased the strain on SS7 security, as the number of nodes connected to SS7 has increased. It is therefore harder to determine whether a newly connected node is trusted or not [15]. Some of the entry points to SS7 is based on interconnectivity, misconfiguration, and unauthorized access to equipment.

Interconnectivity

The telecommunication sector relies on large scale networks, uptime, and a wish by subscribers to be able to contact every other subscriber in the world. These premises makes SS7 extensively interconnected in nature, as the number of interconnected operators increase, the chance of providing better service also increase [7].

An attacker may take advantage of this interconnectivity between operators and SS7 elements. By gaining access to an operator's SS7 network, it is possible that the entirety of any other SS7 network and its entities is available to the attacker. As there may be unserious operators in the world, access to SS7 may potentially be bought on the black market [9].

Misconfiguration

With the introduction of SIGTRAN, it is possible to transfer SS7 messages over IP networks. Using SIGTRAN, it is for example possible to bridge two traditional SS7 networks together using IP and the Session Initiation Protocol (SIP). SIP also makes it possible to provide Voice over IP (VoIP) to subscribers over their internet connection [7].

SIGTRAN provides many advantages for the telecommunication sector and its subscribers. Unfortunately, this also opens up the possibility to transfer SS7 messages over the internet. Langlois have discovered several misconfigured CN elements accessible on the internet in his research [7].

Get unauthorized access to equipment

To improve service and increase their coverage, operators may use femto cells. These are small devices that directly connect to the CN via the internet. These devices have been shown to have unsatisfying security measures as they have been successfully hacked by penetration testers [7].

4.2.2 Mapping the core network

Once inside the CN, an attacker must be able to map the CN infrastructure to be able to successfully launch attacks. Langlois in his presentation [7], recommends several approaches that can be used by an attacker to get an overview of the network and its connected elements. Using the SIGTRAN protocol stack, which uses the SCTP procotol, an attacker may utilize several available tools and techniques to acquire information necessary in the staging step of attacks.

Scanning for open ports

An attacker may wish to scan for open application ports on a device in order to check for SCTP support. As SCTP is extensively used in signaling over SIGTRAN, a node providing SCTP support will indicate an element in the CN [7].

SCTPscan [48] is a tool released by P1 Security initially created to scan for machines enabling SCTP. Exploiting the SCTP handshake routine, it can also be used to detect entry points to telecommunication networks [7].

Creating arbitrary SCTP packages

An attacker may wish to create arbitrary SCTP packages at will. This type of functionality can be used to acquire additional information about a CN entity, for example what kind of applications that element is serving [7].

Using Scapy [49], an attacker is able to generate SCTP packages and their content at will. This tool can be used by an attacker to carefully build SCTP packages from scratch in order to get information about a node in the CN [7].

4.3 Attacks on SS7

As discussed in the beginning of the chapter, the attacks made possible on the SS7 network gives attackers the ability to intercept calls and text messages, commit fraud, deny service and track the location of mobile subscribers. Attacks on SS7 are happening, and have been detected in the wild by several companies [15]. In this section, the disclosed attacks using SS7 will be presented in detail.

4.3.1 Interception

Interception is one of the most devastating attacks against a subscriber in a telecommunication network. If an attack is to succeed, an attacker may be able to read, store, and alter subscribers' phone calls, SMS, and data. As a lot of computing is shifting to smart phones and mobile computing is more popular than ever [1], interception may prove to be one of the most critical attacks if successful.

Intercepting calls by decrypting radio traffic

The CN continuously handles the location of the mobile subscriber and makes it possible to smoothly perform a call when the subscriber is on the move. To make this possible, the MSC controls the encryption keys used to establish and encrypt communication. It may happen that the mobile subscriber crosses the

border from one MSC/VLR area to another during a call. In this scenario, the new MSC/VLR will perform a handover process to transfer keys and necessary information from the old MSC/VLR. This handover process is started with the MAP *sendIdentification* message. The sendIdentification message is more specifically sent from the new VLR to the old VLR. The old VLR responds with a message containing the encryption keys required to maintain the current ongoing call [9].

If an attacker has access to SS7, the *sendIdentification* message can be used to recover the encryption key used in an ongoing call. The attacker can simultaneously capture the subscribers' radio traffic over the air interface using a capable device. Effectively making it possible for the attacker to decrypt the ongoing communication. Decryption of the calls does not even have be done live, but can be done at a later time because the attacker has access to both the key and the encrypted call data [9].

Interception of outgoing calls

In the CN, the GSM Service Control Function (gsmSCF) is a logical entity that provides the CAMEL service logic. For a defined number of events, it decides if the event should continue modified, unmodified or be aborted [28]. For example, if a subscriber calls a number without adding the area code (e.g., +47 for Norway), it can be converted to the correct number automatically.

The MAP message *insertSubscriberData* can be sent to an MSC/VLR to change the address of the subscriber's gsmSCF to that of the attacker. When the subscriber then makes an outgoing call, the attacker is able to control the dialed number. For example, the attacker can alter the number to a device controlled by the attacker, record the conversation, and simultaneously forward the traffic to its correct destination. Making it possible for the attacker to listen in on the call, unnoticed by the subscriber [8].

Interception of incoming calls

The MAP message *registerSS* is used to register supplementary services for a subscriber, it can for example be used to enable call forwarding [2].

An attacker can use the registerSS message to enable call forwarding to a device controlled by the attacker. When receiving the call, the attacker can then use the MAP message *eraseSS* to remove call forwarding for the subscriber, and then forward the call back to the victim subscriber. This attack makes it possible for an attacker to listen in on an call without the subscribers' knowledge [13].

Interception of SMS

The MAP message *updateLocation* is part of the location management service in the CN. It is used to notify the HLR when a MS moves to a new MSC/VLR area. This is done so that calls, SMS, and data intended for the subscriber can be routed accordingly [20].

An attacker can use the updateLocation message to tell the HLR that the MS

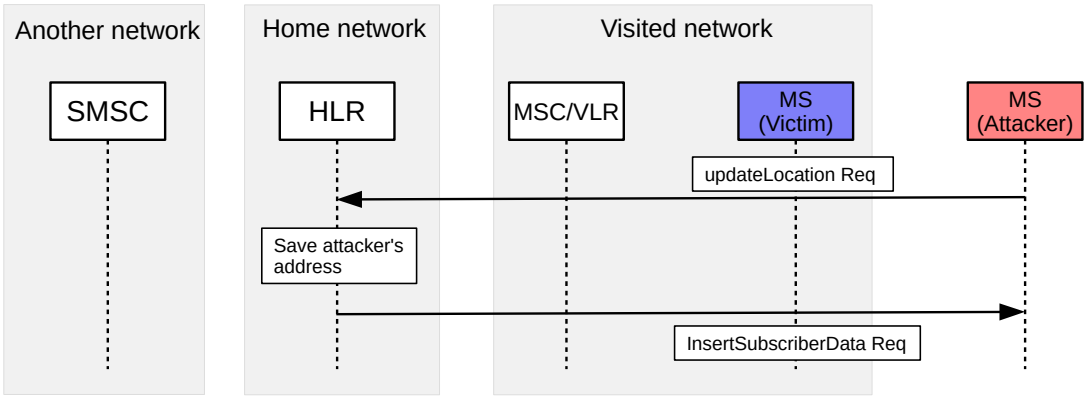


Figure 8: Message flow showing an attacker stealing a subscriber using the MAP updateLocation message. Inspired by drawings in [8].

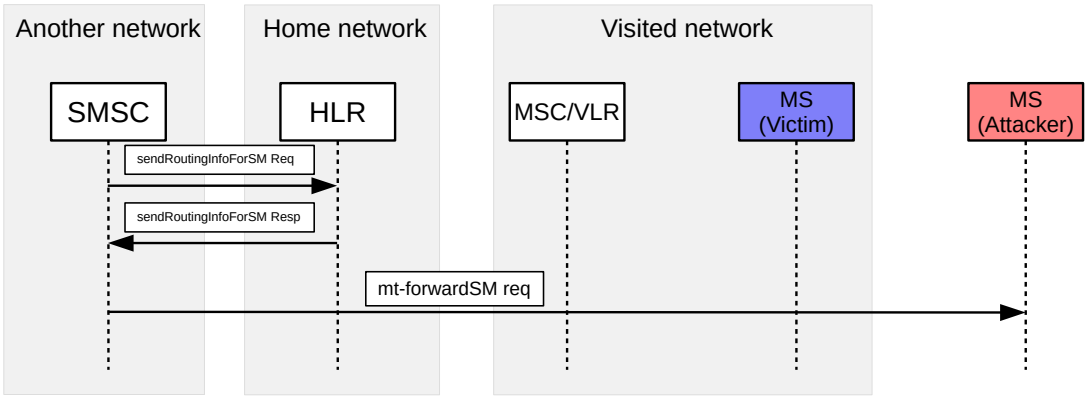


Figure 9: SMS is now sent to the attacker instead of the intended subscriber. Inspired by drawings in [8].

has moved into a fake MSC/VLR area controlled by the attacker, as shown in Figure 8. The HLR will now store the address to the currently serving MSC/VLR, essentially capturing the location of the subscriber [20].

When an SMS intended for the victim subscriber is sent by another MS, the SMSC will first query the subscriber's location for routing information using the *sendRoutingInfoForSM* message. This lookup operation will now return the location of the attacker controlled entity, and the SMS will be forwarded to the attacker, this procedure can be seen in Figure 9. The attacker is now fully controlling the message, and can decide to read it, store it for later use, alter it, or even forward it to the intended subscriber so that neither of the subscribers will notice that they have been attacked. This attack can be devastating when combined with one-time passwords used for example by banks [8].

4.3.2 Fraud

Fraudulent activities done by an attacker in SS7 aims to deprive subscribers of monetary values, by either illegal transfer of funds to another recipient or by draining the victims' funds.

Transferring funds using USSD

Unstructured Supplementary Service Data (USSD) messages is used to provide services to subscribers such as transfer of money, setup call forwarding and much more. To use these services, a subscriber can send a USSD code from their mobile telephone, which in turn is handled by the CN [8].

The MAP message *processUnstructuredSS* can be used by an attacker to transfer a USSD code on a subscribers behalf. Thus making it possible for an attacker to for example transfer money to an unintended receiver. This is possible because the entities in the CN does not confirm the location of the subscriber when accepting USSD messages [8].

Forward calls to premium numbers

The same approach using USSD codes can be used to alter call forwarding settings for a subscriber. For example, an attacker could forward a subscriber's calls to an attacker controlled premium rate number and then call the subscriber's number. Causing the subscriber to receive all costs from calls to the premium service [8].

Unblock stolen devices

Rao present an attack in his master's thesis [47], where he shows how it can be possible to unblock a stolen mobile phone. The Equipment Identity Register (EIR) is used to whitelist, greylist, and blacklist devices. Using the MAP *checkIMEI* message sent to the subscriber's HLR, an attacker is able to whitelist a previously blacklisted device in another country by supplying a new SIM card with a new IMSI.

4.3.3 Denial of service

Denial of Service (DoS) attacks has the goal of disrupting the service for subscribers. Possibly making calls, SMS, and other services unavailable.

DoS by altering subscriber data

The HLR contains data describing what a subscriber is allowed to do, for example if the subscriber is allowed to make calls, send SMS, and general information about the subscription status of the subscriber. Subsequently, the VLR holds a copy of this data when a subscriber is in its area [28].

Using the MAP messages *insertSubscriberData*, *deleteSubscriberData*, and *cancelLocation*, the attacker can alter what the subscriber is allowed to do. The attacker may for example disallow phone calls and SMS, or even remove the subscriber from the VLR altogether [8].

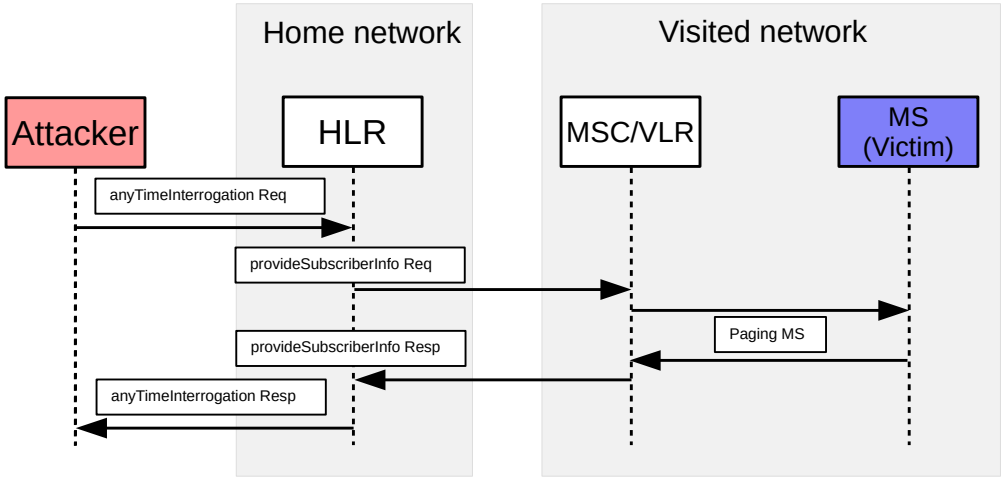


Figure 10: Message flow of the location tracking attack using the anyTimeInterrogation message. Inspired by drawings in [8].

4.3.4 Location tracking

Location tracking in a mobile network makes it possible for an attacker to get insight into subscribers' currently connected cell, or their currently serving MSC and VLR. Based on the density of the serving infrastructure, the attacks may produce location accuracies down to street level in urban areas [8].

Tracking subscribers using anyTimeInterrogation

The MAP *anyTimeInterrogation* message is used internally in the subscriber's home CN for looking up subscriber information. An attacker with access to SS7 can send an anyTimeInterrogation message to the subscriber's HLR. The attacker must only know the subscriber's MSISDN, or telephone number, in order to complete the attack. The message will trigger the HLR to send a provideSubscriberInfo request to the currently serving MSC/VLR. The MSC/VLR will then confirm the location of the subscriber's MS by querying the device, as seen in Figure 10. The attacker will acquire the current identification number of the cell (cell-ID) which the MS is connected too. The cell-ID number can be used to identify the location of the cell in online databases [8].

Tracking subscribers using provideSubscriberInfo

As the anyTimeInterrogation message is only intended to be used internally in an operator's network, some operators have actually started to block the message [8].

As the MAP *provideSubscriberInfo* message is used to request information on a subscriber from a VLR at any time [20]. The attacker can circumvent the HLR and directly query the MSC/VLR instead. To do so, the attacker must first acquire

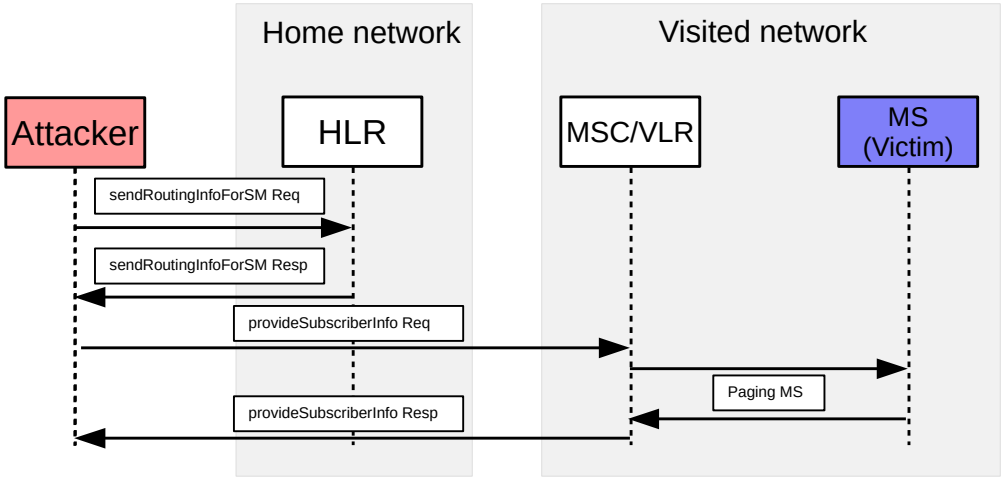


Figure 11: Message flow of the location tracking attack using the provideSubscriberInfo message. Inspired by drawings in [8].

the IMSI of the subscriber and the address to the MSC/VLR. This information can be acquired by first sending a sendRoutingInfoForSM request to the subscriber's HLR which will provide this information [8]. The message flow of this attack can be seen in Figure 11.

4.4 Vulnerability classification

To help operators manage some of SS7's vulnerabilities, the MAP messages used in all disclosed SS7 attacks can be classified based on their need for exposure to external networks. This classification is done in three classes, by messages that have: (1) no need for external exposure, (2) no need to be exposed externally for an operator's own subscribers, but can be received for other operator's roaming subscribers, and (3) legitimate need for external exposure [13]. An overview of the messages with matching classification class is shown in Table 1.

4.5 Initial attack mitigation

There are several techniques that have been recommended to provide some mitigations to the SS7 vulnerabilities. These techniques are not intended to specifically stop attacks, but they provide another layer of security. These are best practice recommendations that makes it harder for an attacker to perform the disclosed attacks, but does not prevent them. Rao [47] goes through some of the recommended approaches in his master's thesis. Essentially, the recommended approaches boils down to monitoring and understanding the SS7 network traffic. Operators are recommended to analyze the SS7 network links for suspicious behavior.

Category	Message	Attack
1	sendIdentification (SI)	Interception
1	anyTimeInterrogation	Tracking
1	anyTimeModification	Tracking
1	provideSubscriberLocation	Tracking
2	insertSubscriberData and gsmSCF	Interception (Outgoing)
2	insertSubscriberData	Denial of Service
2	deleteSubscriberData	Denial of Service
2	provideSubscriberInformation	Tracking
3	sendAuthenticationInfo	Interception
3	registerSS - eraseSS	Interception (Incoming), Fraud
3	updateLocation	Interception (SMS), Denial of Service
3	processUnstructuredSS	Fraud
3	cancelLocation	Denial of Service
3	sendRoutingInformation (-SM, -LCS)	Multiple attacks

Table 1: Classification of SS7 MAP messages used in attacks [13].

4.5.1 SMS home routing

A number of the publicly disclosed attacks, explained previously in this chapter, involves use of SMS procedures. These are procedures originally intended for the GSM architecture and specified by 3GPP [29]. The procedures have some design flaws that enables an attacker to gain staging information for a number of attacks, by for example using the MAP *sendRoutingInfoForSM* message. This message provides the attacker with crucial identifiers such as the IMSI of a subscriber and the address to entities such as the MSC and the VLR.

Instead of disclosing this information, 3GPP have proposed an improved specification for sending SMS labeled "SMS home routing" [50]. Basically, this specification proposes that all SMS intended for a subscriber should be sent via the subscriber's home network. Using this method, it is not needed to hand out the

address of any node except the SMSC and the IMSI need not be disclosed. Instead, a unique identifier, MT-SMS Correlation ID, containing only the MCC, the MNC, and a unique sender ID is used to correctly route the SMS to the destination subscriber.

Using SMS home routing reduces the amount of unique identifiers exposed by the operator to a potential attacker. This approach does not stop attacks specifically, but makes it harder for an attacker so successfully perform an attack as staging information must be acquired using other methods.

5 Detecting Attacks on SS7

In the previous chapter, some of the vulnerabilities and threats towards SS7 was uncovered. This chapter explains some of the challenges in detecting attacks in the SS7 network. To investigate the suggested methods to detect attacks, a simulator was implemented to generate SS7 traffic.

5.1 Challenges in detecting attacks on SS7

Based on the classification of messages used in SS7 attacks as shown previously in section 4.4, there are several options as to how one could deal with the categorized messages.

Category 1 messages can be filtered by relatively simple techniques at the network border. This can be done by looking at the type of message, and assess whether or not the message has been sent from an external SS7 network. Category 2 messages cannot simply be filtered at the network border. An operator must correlate subscriber states and check if the subscriber is roaming in the operator network before a potential message can be blocked. Unfortunately, this does not protect roaming subscribers [13].

To detect attacks using category 3 messages, more sophisticated approaches must be utilized. These are messages that have a legitimate use in the network and cannot simply be filtered. A protection system needs to parse the network message flow and be able to look for change in behaviour of network elements and subscribers. By for example looking at a subscribers' previous location [13].

5.2 Analyzing an SS7 attack

An effort was made to find mechanics as to how SS7 attacks could be detected. Based on inspection of 3GPP standards and following the recommendations of researchers and companies working on SS7, a decision was made to shift all focus on the intercept SMS attack as described in section 4.3.1.

As a recap, looking into the procedure of the attack, the MAP *updateLocation* message is the initial trigger for the successful attack. Which causes the HLR to respond to an SMS routing request with the address to an attacker controlled node. Causing subsequent SMS to be delivered to the attacker.

The updateLocation message in itself carries information that gives insight into what is happening to the subscriber's state. In the message, information such as the IMSI, the new MSC address, the new VLR address, and the pagingArea is being transfered. In particular, the mandatory field MSC/VLR-number and the optional field pagingArea, which contains the new Location Area Code

(LAC) of the subscriber, gives an insight into the new geographical location of the subscriber [20].

Based on this location information, it is possible to get an indication as to where the subscriber is traveling and at what speed. And based on this, it is possible to deviate how a subscriber normally moves as seen by the SS7 network. Using this information it can be possible to get an indication of implausible location updates by comparing the new location update to the previous ones.

As an example, an updateLocation request arrives from an MSC/VLR currently serving a subscriber in the urban part of Oslo, Norway. A few minutes later, an updateLocation request for the same subscriber arrives from an MSC/VLR in Barchelona, Spain, suggesting that the subscriber is now located in this area. This scenario can be considered abnormal, as it is physically impossible that a person is able to travel that sort of distance in such a short amount of time.

5.3 The potential of machine learning and anomaly detection

In summary of section 5.1, detecting attacks using category 2 and 3 messages are quite challenging. In order to detect them, an operator must gather additional information about the behavior of subscribers and the elements in the SS7 network. A detection system can use this information to be able to distinguish what is normal behavior, and what is not. Essentially, a detection system must be able to look for *anomalies* in subscriber and network behavior to detect these sort of attacks.

Machine learning is a tool that can help in anomaly detection, where we solve the problem of detecting patterns in data that deviate from the expected normal behavior. In addition, machine learning can assist in building profiles that can be used to model the normal behavior. The modeled normal behavior provides a measure to look for anomalies [40]. These anomalies might indicate that an attack has been launched towards a subscriber or a network element.

5.4 Capabilities of the operator

Initially, in order to detect attacks in SS7, it is presumed that the operator possesses some capabilities in their network. Firstly, the operator must be in control of their own SS7 elements. Meaning that an operator is able to separate their internal network, or home network, from all other external networks. Secondly, the operator must be able to capture traffic entering the defined border of the network. Making it possible to determine from where a message originated, externally or internally.

Operators needs to understand that SS7 is no longer secure, and they need to separate themselves from other SS7 networks in order to protect their own network and their subscribers. Separation can be done by for example dedicating a Signal Transfer Point (STP), or any other SS7 capable node at the border of the network to perform analytics on incoming traffic. A visual explanation of this

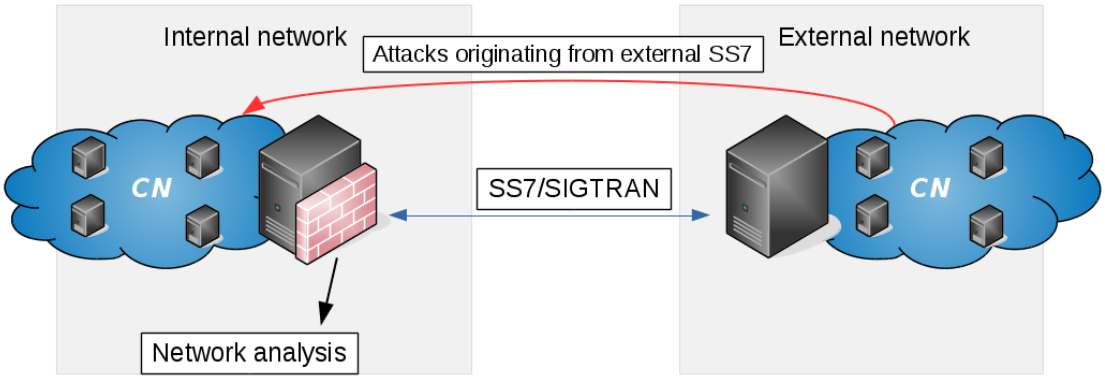


Figure 12: Separating the home network from external networks to be able to distinguish between internal and external SS7 traffic.

separation can be seen Figure 12.

5.5 The SS7 Attack Simulator

To test the feasibility of machine learning and anomaly detection, the goal was to experiment with several anomaly detection techniques on SS7 network traffic. Due to privacy and ethical concerns when accessing an SS7 network and obtaining SS7 traffic data, no real SS7 network traffic was used in the experiments.

To handle the issue of obtaining a dataset, the SS7 Attack Simulator [51] was created using the free and open source SS7 stack jSS7 created by RestComm [23]. The jSS7 implementation was forked and modified in such a way that suited the purpose of the experiments. The SS7 Attack Simulator was released as open source software publicly available on GitHub [51].

The attack simulator was created to serve two distinct purposes relevant to this thesis: (1) simulate the message flow of publicly disclosed attacks, and (2) simulate an SS7 network generating normal and abnormal network traffic which can be analyzed using machine learning techniques. By simulating the message flow of attacks, it is possible to specifically show how some of the attacks pan out. This visualization makes it easier to educate people on the vulnerabilities of SS7 and the corresponding attacks.

5.5.1 Simulator capabilities

The simulator consists of an SS7 network containing three operators communicating using the SIGTRAN stack, a high level overview of the simulation is shown in Figure 13. Two of these operators have a CN containing a set of interconnected CN entities. Operator A and B both have an MSC/GMSC, VLR, HLR, and an SMSC. In addition, operator A also includes a gsmSCF and a SGSN, which

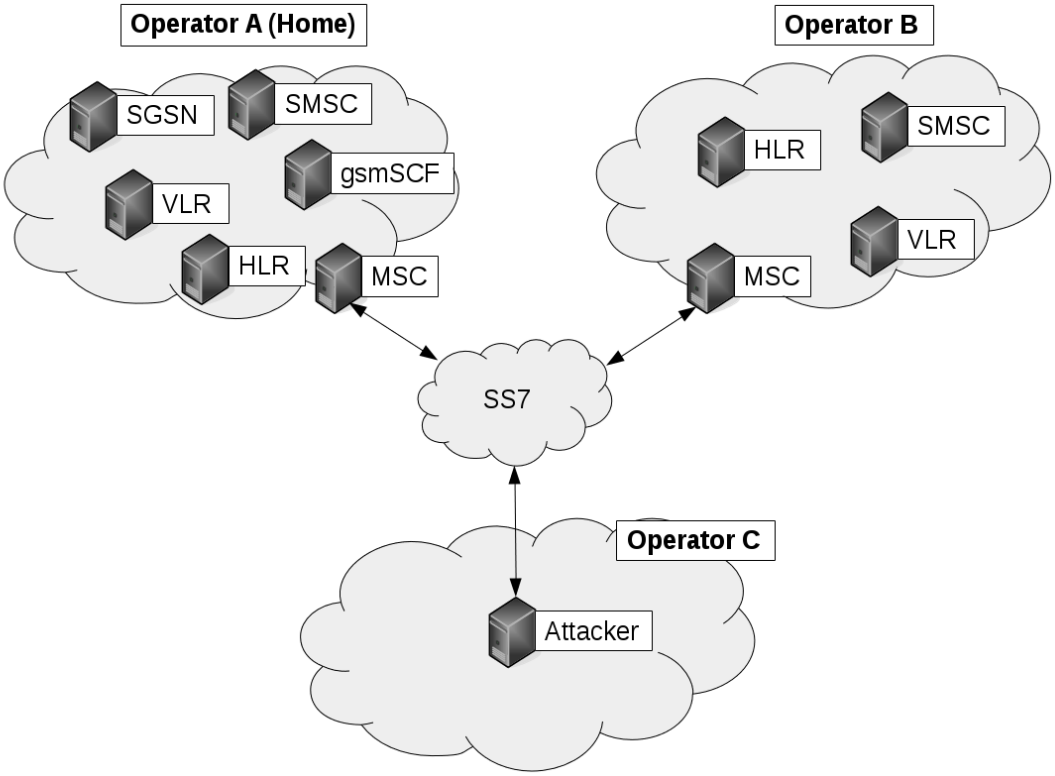


Figure 13: The nodes contained in the simulated network.

handles some additional procedures in the network. To analyze SS7 attacks, the simulator has the capabilities to generate both normal traffic and attack traffic. As each of the entities in the network communicate with each other using the SIGTRAN stack, it is possible to use the simulator on a computer that provides the IP and SCTP protocols. Although this particular implementation is specifically targeting the GNU/Linux operating system.

5.5.2 Generating SS7 traffic

The procedures generated by the entities contained in the network perform operations on a set of subscribers. Each subscriber is defined by a set of parameters created by the simulator, which include: IMSI, IMEI, subscriber state, location information, home network, and currently serving MSC/VLR. One of these subscribers is defined as a very important person (VIP), which in real life could for example be a CEO or a political figure of high value that is an attractive target for an attacker. The messages generated by the CN entities alter the state of the subscribers, making each message have an effect on the state of the subscribers and nodes, thus creating a dynamic system where most of the messages have a purpose. Examples of generated traffic from the simulator can be seen in

Procedure	Service	Communicating nodes
Location Update	Mobility	MSC,VLR,HLR
Purge MS	Mobility	HLR,VLR/SGSN
Delete Subscriber Data	Mobility	HLR,VLR
Any Time Interrogation	Mobility	gsmSCF,HLR
Short Message Mobile Originated	SMS	MSC,SMSC,HLR
Short Message Mobile Terminated	SMS	MSC,SMSC,HLR
Short Message Alert	SMS	MSC,SMSC,VLR
Retrieve Routing Info	SMS	MSC,HLR,VLR
Send Routing Info For GPRS	PDP	SGSN,HLR
Activate Trace Mode	Oam	HLR,VLR
Send IMSI	Oam	HLR,VLR
Registration Procedure	Supplementary	MSC,VLR,HLR
Erasure Procedure	Supplementary	MSC,VLR,HLR

Table 2: Implemented normal MAP procedures in the simulator.

Appendix B.

Generating normal traffic

To as closely as possible create a simulation that can be related to a real operating SS7 network, network traffic representing normal procedures (i.e. not including attacks) where generated to act as background noise in the network. All of this traffic is created using the MAP in the SS7 stack.

The entities in the network uses the jSS7 stack to perform standardized procedures as defined by 3GPP in the MAP technical specification [20]. Thirteen procedures was chosen based on the capabilities of jSS7 and their ease of implementation. Each of these procedures conforms to the MAP standard, and sends the minimum mandatory parameters for each message as defined by 3GPP. All implemented procedures can be seen in Table 2.

Generating abnormal (attack) traffic

In addition to normal behavior generated by the entities, three types of attacks was implemented to act as abnormal traffic. The attacks implemented includes:

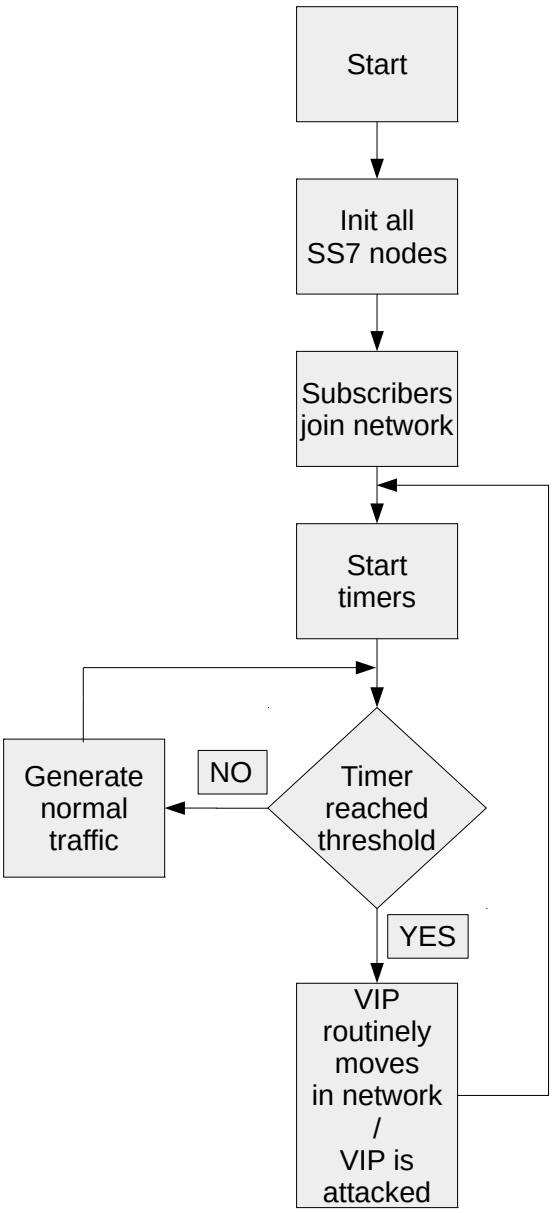


Figure 14: Simple flowchart that describes how traffic is generated in the simulator.

(1) location tracking using anyTimeInterrogation (section 4.3.4), (2) location tracking using provideSubscriberInfo (section 4.3.4), and (3) intercepting SMS by stealing subscribers (section 4.3.1). All of these attacks are launched from an entity located in operator C, to simulate that an attacker has gained access to operator C's SS7 network by some manner. These attacks use one MAP message

from each of the defined classifications of messages, as seen in section 4.4.

5.5.3 Simulator operations

The functionality of the simulator boils down to a set of operations that defines what to do next. A simplified flowchart of these operations can be seen in Figure 14.

The operations of the simulator heavily relies on timers to decide what the next action should be. There is one main timer, that decides whether or not the VIP should be influenced, and several other timers that decide what sort of action should be done in regards to the VIP. When this timer has not reached a certain threshold, the simulator will generate normal procedures that serve as background traffic. The choice of which normal procedure that should be generated is decided based on the built in Java pseudorandom generator.

5.6 Simulating a real life scenario

As a basis for testing the feasibility of machine learning and anomaly detection in an SS7 network, a real life scenario was implemented as part of the simulator.

In a real life SS7 network, there will be a large amount of subscribers associated with a network all the time. If an operator wishes to create profiles on these subscribers in order to detect attacks against them, it would not be feasible to do this for every subscriber due to the wast amount of subscribers. So it is assumed that an operator would focus on a smaller amount of subscribers. These subscribers could in real life be important individuals, for example a CEO of a larger company or a political figure. In essence, these subscriber are attractive targets for an attacker.

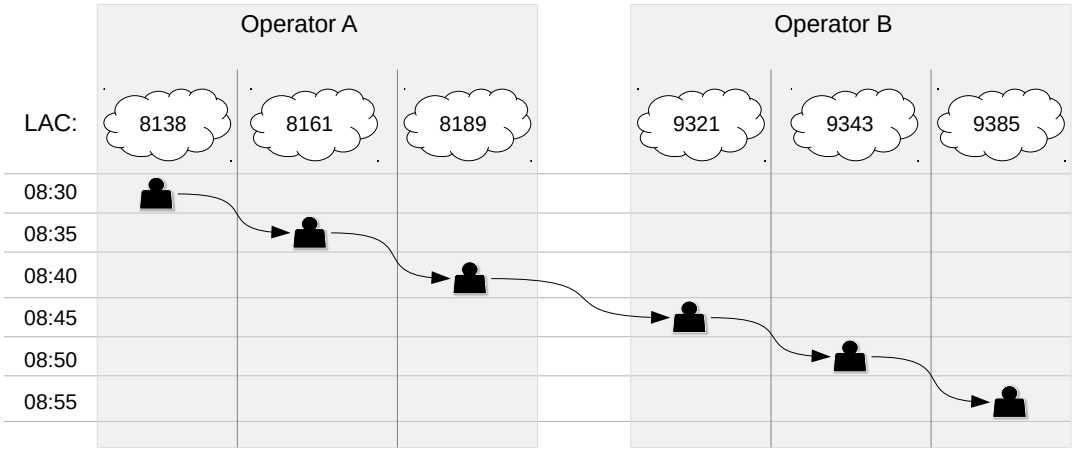


Figure 15: Example of a subscriber moving trough different location areas at different times, indicated by their location area code (LAC).

Based on these assumptions, the Very Important Person (VIP) subscriber is introduced as part of the simulation. This is a subscriber that is an attractive target for the attacker, and is therefore the only subscriber being attacked in the network.

5.6.1 VIP movements

To address the possibly normal behavior of a subscriber as seen by the network, the VIP subscriber is simulated to routinely move between different areas defined by their Location Area Code (LAC). For example, in real life this could be that the individual is moving to and from work every day. This is illustrated in the network by the subscriber moving trough three LACs in operator A's network, and three LACs in operator B's network. The subscriber's home is located in a location area defined by operator A, and the work location is defined by operator B. An example of these simulated movement can be seen in Figure 15, where it is shown that the VIP is moving to work in the morning trough 6 location areas.

6 Applying Machine Learning to Detect SS7 Attacks

To explore the feasibility of applying machine learning and anomaly detection in an SS7 network, the SS7 Attack Simulator was paired with machine learning techniques. In this chapter, a proof of concept will be presented, including offline testing and a demonstration of an implemented Anomaly-Based Network Abuse Detection System (A-NADS). All data used in this chapter was gathered from the running SS7 Attack Simulator, as explained in the previous chapter.

6.1 Anomaly detection technique

In order to solve the anomaly detection problem, a capable technique and algorithm must be chosen. There are a vast number of algorithms and techniques available in the machine learning and anomaly detection field [40]. If one wishes to select the most fitting approach, there are several questions that must answered. One of the more important ones is the choice of an overall machine learning technique.

There are two recommended approaches that can be used when detecting anomalies: a supervised learning approach or an anomaly detection approach, which can be viewed as an unsupervised detection of outliers. The detection of outliers was briefly introduced in section 3.4.2. Choosing one of these techniques is a decision based on the available data [40].

A dataset can for example consist of many known normal and abnormal events and activities. If there are a large number of labeled normal and abnormal events, a supervised algorithm would be best. On the other hand, a dataset may consist of many known normal events but only a few abnormal events are known. In this scenario there may also be a large number of different anomaly types. If this is the case, an anomaly detection algorithm is best fitted to solve the problem [40].

In the case of publicly disclosed SS7 attacks, there is a large amount of normal events and a small amount of abnormal events [15]. Due to the nature of SS7 and its vulnerabilities, it is difficult to strictly determine if an event is normal or abnormal. The best approach is therefore to use an unsupervised anomaly detection technique. This decision is also strengthened by the fact that the SS7 attacks are more or less indistinguishable compared to normal network traffic.

6.2 Anomaly detection algorithm

Based on the analysis of the intercept SMS attack seen in section 5.2, it is possible to get insight into how a subscriber moves geographically. As an attack will

disturb the normal behavior of the subscriber, one might think to compare a metric to a defined threshold in order to detect an attack. Due to the dynamic nature of subscriber behavior, this approach will not always be applicable. As it is hard to manually find a threshold that encompasses all subscribers and subsequently define rules that apply for all of them [40].

In order to model the behaviour of subscribers, it is proposed to build user profiles to describe this behavior. Based on the fact that when attacked, the attack in itself will introduce abnormalities in the user's behavior. This behavior can be described by analyzing the previous behavior of the subscriber. The challenge lies in the difficulty of representing and generalizing the behavior of the subscriber in order to use it for anomaly detection. This is where the potential of machine learning shows itself. Machine learning, as explained in Section 3.4, provides techniques to automatically learn user behavior and represent user profiles mathematically. Machine learning additionally handles the dynamic nature of user behavior very well. The model created using machine learning can be a tool that is used in order to detect anomalies [40].

In order to detect anomalies, we are essentially applying machine learning to create a distribution function of the user's normal behavior. This function serves as a model that can be used to detect behavior that highly deviates from the normal behavior. Thus shifting the focus to statistical techniques, instead of many other available techniques that can be used to solve the anomaly detection problem [40].

The algorithm used in offline testing was the Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) algorithm, implemented by Twitter. This algorithm is implemented in the R language and is explained in detail in section 3.4.3. S-H-ESD have been reported as highly effective in Twitter's anomaly detection applications [41]. The algorithm was mainly chosen because it handles unsupervised data, it is based on statistical techniques, and because it is able to detect seasonal and trend components not detectable by other algorithms [41].

6.3 Applying the S-H-ESD algorithm

A sizeable dataset of SS7 MAP messages was gathered using the SS7 Attack Simulator, as explained in section 5.5. The simulator was run in *complex* mode, indicating that both normal and attack traffic was generated. Generated network traffic was captured using Wireshark, a packet capturing tool with support for SS7/SIGTRAN[52].

Preprocessing was done by a set of simple scripts implemented in the Python language. These scripts were used to extract MAP *updateLocation* messages for the VIP subscriber. The extracted messages was further preprocessed to extract a set of features that can be used to detect the intercept SMS attack.

Description	Variable type
Time since previous location update	Continuous
Distance traveled since previous location update	Continuous
Byte length of location update	Continuous
Frequency of location updates	Continuous
Message network origin	Nominal

Table 3: Features selected to detect anomalies in subscriber behavior.

6.3.1 Feature selection

To detect the intercept SMS attack, which is mostly based on the *updateLocation* message, a number of features are proposed to describe the behavior of the subscribers. These features can be seen in Table 3 and will be explained in more detail as follows.

Time since previous location update

How long ago the subscriber's MS updated its location will give an indication as to how long ago the subscriber moved. Which in turn will give insight into the normal intervals the subscriber moves at.

Distance traveled

The distance traveled since the last location update will explain how far the subscriber has geographically moved since last update. Which in turn will give insight into how far the subscriber normally moves. It is assumed that an operator can provide some metric to describe how far it is between different LACs. An operator may also use the areas covered by an MSC/VLR area as basis for this metric, or any other approach that provides a metric as to how far the subscriber has moved.

Byte length

This is a feature typically used in anomaly detection systems when detecting network intrusions. It is assumed that an SS7 message created by an attacker may deviate, if only slightly, from a normal SS7 message.

Frequency of location updates

The frequency of location updates will give an indication as to how rapidly the subscriber changes location. In essence, an increase in frequency of location updates may indicate that an attack has been launched towards the subscriber.

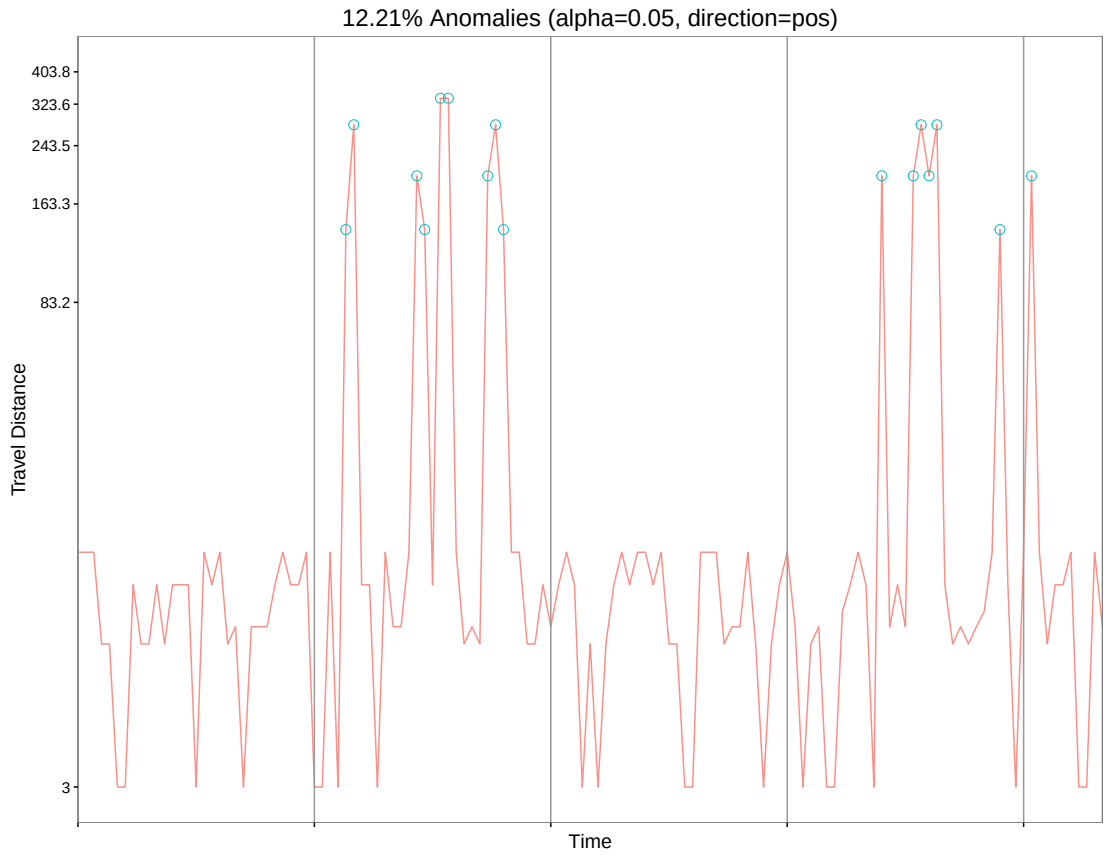


Figure 16: The results of using Twitter's AnomalyDetection on the dataset. Showing that anomalies in the travel speed of a subscriber is detectable.

Message network origin

The message's origin will indicate from which SS7 network and operator the message originated from. It is assumed that there might be a smaller number of networks from where attacks are launched. If this pattern is discovered in an operating SS7 network, it may indicate that an element in that operator's network has been compromised. This feature will also indicate in which operator's network a subscriber is typically located in.

6.3.2 Offline test results

Using the S-H-ESD algorithm, the proposed features where tested as an indication of machine learning's feasibility in detecting the intercept SMS attack. A visual example of the results can be seen in Figure 16, where the travel distance was investigated.

Using a dataset of 59682 samples generated using the SS7 attack simulator, the algorithm spent about 4 minutes and 24 seconds on a modern dual core laptop.

Out of these samples, a number of 6495 samples were detected as anomalies. Based on expert knowledge in regards to the dataset, 3712 of these detected anomalies where found to be actual attacks, i.e. true positives. Leaving the number of 2783 to be false positives, indicating detected attacks which are not actually attacks. The accuracy of the offline detection of attacks was 57% in this iteration.

Furthermore, the analyzed results indicated 100% detection rate, which is expected due to the artificial nature of the gathered data. On the other hand, the False Positive Rate (FPR) was calculated to be 4.7%. This high FPR was expected due to the nature of the subscriber's behavior as seen by the network. When an attack is launched against a subscriber, the subscriber will be seen as moving a possibly great distance, indicating an anomaly. When the subscriber is "returning" to its real location indicated by a new *updateLocation* message, the subscriber will move this great distance once again. Meaning that when an intercept SMS attack is launched, there will not necessarily only be a single anomaly. This implication causes a high amount of false positives without further processing of the results.

6.4 An anomaly-based network abuse detection system

As part of the contribution in this thesis, a prototype Anomaly-Based Network Abuse Detection System (A-NADS) is presented. This demonstration provides an example of how an operator may approach the online anomaly detection problem. It is mostly based on the premisses of a network *intrusion* detection system. As it is a requirement that an attacker has already infiltrated the SS7 network, it is more accurate to use the *abuse* term as a replacement for intrusion. The entire implementation is available as open source software on GitHub [11].

6.4.1 Challenges in online detection

There are several challenges when detecting attacks in a live system. A system of this nature must be effective, reliable, and in general be able to solve the task it was created for. One can for example follow the recommendations provided by NIST in their guide for intrusion detection and prevention systems [53].

The challenges faced by an online abuse detection system includes performance, reliability, interoperability, and scalability [53]. In general, the detection system must solve these challenges in a satisfactory manner in order to provide the efficiency to react and stop an attack without putting too much strain on a production system. There are of course numerous other considerations when deploying an intrusion detection and prevention system, but the topic of optimal intrusion detection and prevention system performance and operations is out of scope for this thesis.

6.4.2 A-NADS stages

To solve the challenges faced in online detection, it is needed to use software that is up to the task. The implemented A-NADS uses several open source software

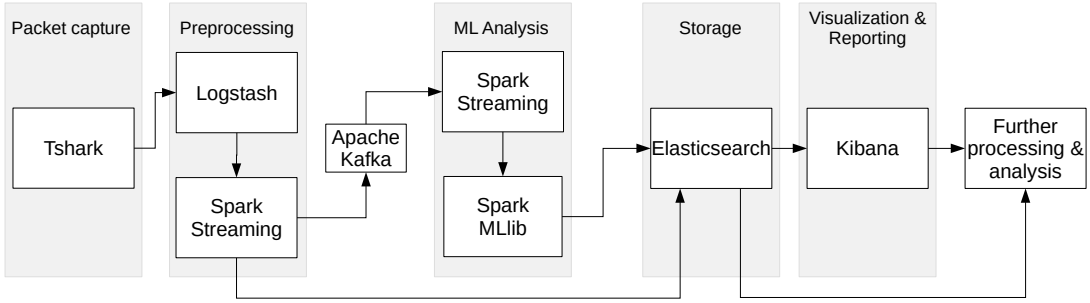


Figure 17: Components of the implemented anomaly-based network abuse detection system.

tools that is designed to handle large amount of data. The software used in the A-NADS can be described using a set of common traits: high availability, scalability, and highly customizable applications that are created to perfectly fit in a big data environment with high demands for performance, throughput, and storage. An overview of the different stages and the software used in the implementation can be seen in Figure 17. In summary, the software used was: Wireshark [54], the ELK stack [55], consisting of Elasticsearch [56], Logstash [57] and Kibana [58], Apache Kafka [59], and Apache Spark [45]. Several stages was defined in order to perform online anomaly detection, these stages will be described as follows. More technical details on the implementation of the stages is shown in Appendix C.

Packet capture

IP packets are captured from the network interface and dissected using tshark, the command line version of Wireshark. The main goal of this stage is to extract the necessary information carried by the MAP required to create features.

Preprocessing

The packets captured from the network interface are processed in order to create features. This stage uses logstash to transfer information from tshark to an Apache Kafka topic. The Spark Streaming library is used to stream the captured packages from Kafka. The main goal of this stage is to create features from the network stream by continuously analyzing incoming traffic. Apache Kafka is used to transfer messages within the preprocessing stage, and from the preprocessing stage to the machine learning analysis stage.

Analysis using machine learning

In this stage, Spark Streaming is used to read the preprocessed features from a Kafka topic and input them to the Spark machine learning library (MLlib). MLlib contains implementations of several machine learning techniques and algorithms

that can be used to perform analysis on the preprocessed features.

Storage

This stage serves as a reference point for captured and analysed data. Elasticsearch was used in the implementation because of its initially simple usage, but in practice any other database or search engine could be used in this stage. The main goal of this stage is to provide a reference point for further analysis, visualization, and for creating reports of attacks and any other behavior in the network.

Visualization and reporting

In this stage, the network flow and results of the analysis is visualized and reported. This could be an overview of the current threats and other information, which could for example be used in an operation center performing live analysis of the SS7 network. Kibana provides dashboards which can be used for this purpose [58]. An example of such a dashboard can be seen in Figure 18, which was used to present some information about the running A-NADS.

Further preprocessing and analysis

This is a recommended step for the A-NADS. In this step, one could for example employ a human with expert knowledge that can analyse the reported anomalies. The main focus in this step is to further investigate the reported anomalies to confirm whether an attack was launched or not.

6.4.3 Online anomaly detection

The implemented A-NADS provides tools to deploy several machine learning algorithms through Apache Spark's MLlib. To provide an example of how one these algorithms can be used, a demonstration was implemented using the A-NADS and the SS7 Attack Simulator. In this demonstration, the k-means|| algorithm was used. K-means|| is an unsupervised clustering algorithm, and is explained in detail in Section 3.4.4. To implement the anomaly detection using Apache Spark, the same approach as in [60] was used.

Known normal events was extracted from the dataset to serve as initialization for the clustering algorithm. Further inputs are compared against the clusters to determine the distance between the new point and the cluster centroids. A point with significant distance from the centroids will indicate an anomaly [60].

Finding a suitable k

The accuracy of the k-means algorithm heavily relies on a properly set number of clusters, k. To select a decent k, the Euclidean distance was used to find the distance between the training data and centroids determined by the k number of clusters. This comparison provides a clustering score for each set k. Basically by traversing a number of k values makes it possible to select a clustering score that provides adequate performance [60].

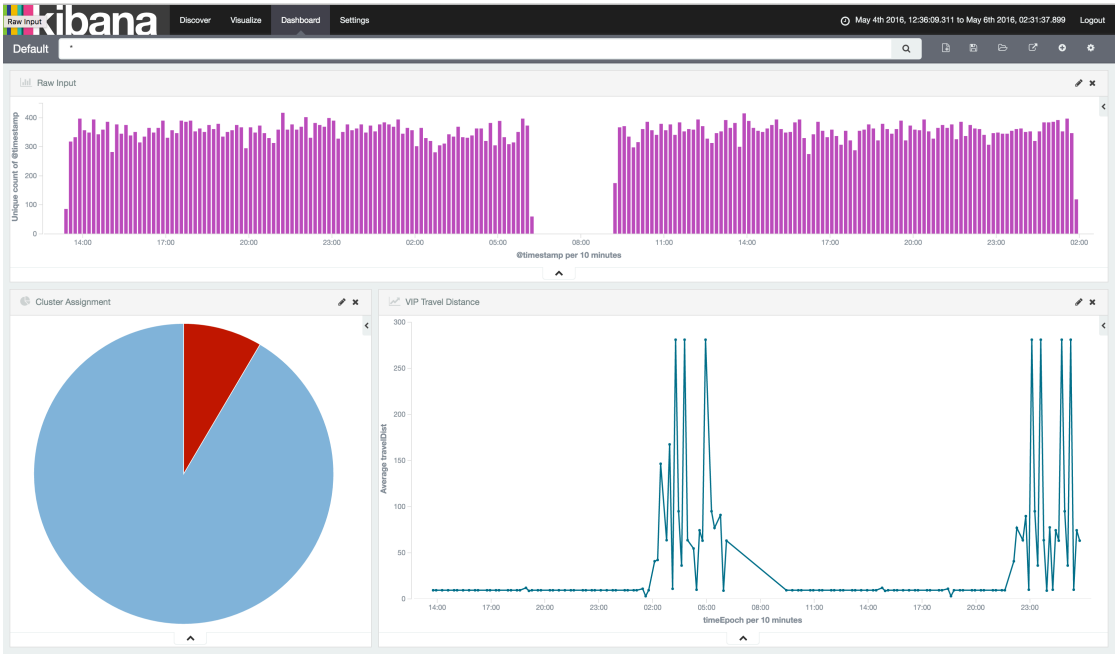


Figure 18: Example of a real time analytics dashboard monitoring SS7 traffic using Kibana. Showing the input flow to the system, the cluster assignments and the distance traveled by the VIP subscriber.

Scaling the feature set

The k-means algorithm is fairly sensitive to the scale of the input features [60]. In order to provide a more accurate result, the dataset was scaled using its standard score based on the datasets' mean and standard deviation. Apache Spark MLlib provides the class `StandardScaler` which were used on the training dataset and any sequential input.

Selecting a threshold

The difficulty of providing an accurate results when using this approach is selecting a proper threshold. Ideally, one might like to use a statistical machine learning approach in order to make these decisions easier, for example like the S-H-ESD algorithm used in section 6.3.2. Due to limitations in both time and available implementations, a statistical approach was not feasible in this prototype. A threshold was therefore selected on a best effort basis by combining expert knowledge of the dataset with several comparisons of known attacks and normal traffic.

Online anomaly detection results

Using the same dataset as in the offline experiment presented previously in this chapter, the k-means|| algorithm was applied to detect anomalies. As parameters for the algorithm, a number of $k = 219$ clusters was selected paired with a threshold value of $t = 0.2$. A number of 7250 anomalies was detected out of the 59682 samples. The number of false positives were 3528, compared to 3712 actual attacks in the dataset. This resulted in a detection rate of 100% (as expected), and a False Positive Rate (FPR) of 5.6%.

Once again, as discussed in regards to the offline test results in Section 6.3.2, these number are expected in this scenario. Based on the fact that a minimum of one anomaly will occur with every attack. There is also room for improvement in how the machine learning algorithm, k-means|| in this case, is being used. This is not the optimal way of applying this sort of anomaly detection, and the author leaves it for further work to find the best way to perform anomaly detection on this sort of data.

7 Discussion

The methods used in this thesis are all subject for further discussion. In this chapter, a discussion regarding the general approach about this thesis will be presented. Including some considerations an operator must address when deploying the methods used in this thesis.

7.1 Experimenting with artificial data

The general approach of this thesis has been based on artificially generated network traffic. This approach has several pros and cons that have a need to be discussed.

Using artificial data as opposed to working with real data, one can avoid the ethical and legal considerations of sensitive information carried by network traffic. As SS7 traffic contains magnitudes of information that is considered sensitive, such as IMSIs and SMS [2]. On the downside, a simulation like this will never fully represent a real network and its generated traffic. It is therefore hard to claim that the results in this thesis is an absolute fact. Still, the results give an indication that if the claimed assumptions hold in a real world SS7 network, the approach used in this thesis will also work in real life.

7.2 Considerations when deploying an A-NADS

If an operator wishes to deploy an A-NADS in a production environment, there are several factors that must be considered. These considerations are related to the impact of the A-NADS on the network, in regards to configuration and performance of the system. As the approach is fairly similar, it is possible to compare the abuse detection system to an intrusion detection approach. In which the recommendations by NIST [53] provides a comprehensive overview of most of the concerns when assessing such a system.

7.2.1 Requirements of an A-NADS

An A-NADS must provide adequate performance if it is to be considered deployed at all. This is not only anomaly detection performance, as in the number of detected attacks shown by metrics such as false positive rate. But an operator must also consider performance metrics such as throughput, scalability, cost of maintenance and several other factors when deploying such a system [53].

7.2.2 Deploying anomaly detection and machine learning

Anomaly detection and machine learning techniques have received a lot of praise and are considered highly effective in multiple applications [40]. There are also

several downsides of applying both techniques, as discussed in [61] and [62] where they share their skepticism on deploying machine learning in an anomaly based detection system. These concerns are mostly grounded in the fact that attackers are possibly able to alter what an anomaly detection algorithm considers normal. An operator must address the weaknesses of using anomaly detection and machine learning in its operations by proposing and implementing countermeasures to these risks.

7.2.3 Risk analysis of an SS7 attack detection system

When deploying a protection solution using machine learning, such as the A-NADS proposed in section 6.4, there are several risks that must be considered. Some of these considerations are discussed for intrusion detection systems in [63], and can be applied to an A-NADS:

- **Skills and resources of attackers:** SS7 is not a common, nor well known technology used daily by the general public. It is a closed network not accessible to people outside of the telecommunication sector. Thus making it relatively difficult to experiment on and therefore makes it hard to gain expert knowledge on SS7 without proper training or special interest. It is safe to assume that people with knowledge of SS7 and telecommunication networks are experts, as the networks and technology are not easily accessible. Attackers are therefore more likely to be professional, and there is a scarcity of "script-kiddies".
- **Security of an A-NADS:** The A-NADS itself must be secure, as it can serve as an attack vector for the attacker. This issue also extends to the machine learning algorithm used for anomaly detection. There are several weaknesses of the anomaly detection approach and using machine learning as a detection mechanism [61, 62]. Operators must employ countermeasures for these risks, and handle them correctly lest the detection system be a liability instead of a tool in attack mitigation. The A-NADS potentially handles sensitive information that should remain confidential, for example the IMSI of a subscriber and the subscribers' current location. Handling such information is governed by the European data protection law in the European Union [64] and under "Personopplysningloven" in Norway [65]. An operator must take caution to ensure that the A-NADS itself is not compromised and potentially leak sensitive subscriber information. This would not only lead to disclosure of sensitive personal information, but may lead to legal repercussions.
- **The cost of a missed attack:** If the A-NADS is to miss an attack, there are several consequences for both the subscriber and the operator. The privacy of the subscriber will be violated, potentially disclosing sensitive information about the subscriber. On the other hand, if a successful attack is to be known publicly, the operator may experience loss of face which in

turn can reduce the operator's reputation and its subscription base. From another viewpoint, the cost of a missed attack is not necessarily critical to the operation of an operator. But still, the information disclosed is sensitive information that should not be handed out easily by an operator.

- **Preventing detected attacks:** Prevention of attacks is suggested in 8.3 as a potential topic for further research. An A-NADS with these capabilities will have to specifically stop SS7 network messages in order to mitigate attacks in the network. There are several concerns when an A-NADS is given such potentially destructive power. One must ask questions such as: what if the detected attack was a false positive? Misclassified attacks may be devastating in some scenarios, potentially making an operator deny service for its own subscribers. Perhaps a critical message flow was interrupted, causing an SS7 element to stagnate and not perform its tasks. Decisions made in such circumstances must be well documented and be done with high confidence. Correlation and data mining techniques can be used to identify patterns in evidence from multiple sources. By performing an automated forensic approach using pattern analysis, the level of confidence when making such a decision can be increased [66]. An operator must definitively address these concerns when wanting to deploy such capabilities. Network traffic move fast, and there may not be much time to make a decision on such a matter if the SS7 network should not slow down.

Overall, an operator must understand the consequences of deploying a protection system and its implication on the existing infrastructure. By understanding the threats and vulnerabilities of SS7, while simultaneously keeping in mind the risks of deploying a protection system, an operator may substantially improve the security of their SS7 network.

8 Future Work

Based on the methods and results in this master's thesis, several other questions and concerns emerge. In this chapter, ideas and recommendations for future work on this topic are presented.

8.1 Performing the experiment in a real SS7 network

The generic approach in this thesis has been based on simulation and generation of artificial data. To further prove the applicability and feasibility of the proposed detection methods, experiments has to be performed in a real life SS7 network using real life data. This has to be done in order to verify that the claims made in this thesis holds in real life.

Additional work must also include the detection of other types of attacks, not just the once discussed in this thesis. This can be done by using a similar approach as done in section 5.2. As attacks on SS7 changes the behavior of the subscribers and network elements, features must be discovered that can map the change in behavior for other attacks. These features can be detected by carefully analyzing SS7 standards and network behavior.

8.2 Optimal performance of an A-NADS

The detection mechanisms used in this thesis is not necessarily the most effective way to apply machine learning tools in regards to performance and efficiency. An operator wishing to implement an A-NADS must consider every form of algorithm and method available. There is no catch-all algorithm when it comes to applying anomaly detection and machine learning. Several considerations must be taken, such as the performance, detection rate, false positive rate, and much more [40].

8.3 Prevention of SS7 attacks using an A-NADS

The main goal of the implemented A-NADS is rooted in the detection of attacks. Still, it would be interesting to implement prevention capabilities of such a system. This could for example be done by implementing an Application Program Interface (API) able to intercept and stop messages used in attacks. An operator that wishes to implement such features should do so with caution as the implications of stopping messages is unknown. Some of these problems are discussed in Section 7.2.3.

8.4 Extend the simulator to create a security testbed

The SS7 Attack Simulator that was implemented in this thesis (Section 5.5) has further potential. The simulator may for example be extended to be part of a security testbed. In this testbed, the simulator can be used to test the resilience of SS7 elements by generating attacks and any other sort of traffic that may be used to test the security and reliability of SS7 elements. Such a testbed will prove to be useful for an operator wanting to test the current infrastructure and test new features. Further, a testbed of this caliber can be used to research the implications of SS7 attacks and provide a tool for testing potential SS7 attack detection methods.

9 Conclusions

This thesis has given insight into Signaling System No. 7 (SS7), the nervous system of telecommunication systems. SS7 was created in a time where there were a small amount of larger and trusted operators. Today's telecommunication market is severely different, the market has been deregulated and there are a larger amount of small operators potentially connected to the closed SS7 network. Combined with the fact that the industry and standardization bodies are actively transitioning to IP technologies. The SS7 networks has experienced an increased number of connected nodes and therefore increased attack surface.

In recent years, several SS7 vulnerabilities have been disclosed. Resulting in SS7 being possibly exploited by serious attacks such as location tracking, interception, denial of service, and fraud. These attacks threaten the privacy of subscribers and the integrity of operators' SS7 networks. The problem with detecting and preventing SS7 attacks is that all messages in the network is inherently legal. Causing attacks and legal traffic to be close to indistinguishable. An operator cannot simply stop SS7 traffic, as it will possibly disrupt network operations and deny service to subscribers.

In this thesis, an intuitive protection solution based on machine learning techniques was proposed to detect attacks against SS7. Successful attacks where detected to alter the behavior of subscribers and the corresponding network elements. Therefore, it was proposed to create user profiles to map subscriber and network behavior in order to detect abnormalities.

To test these claims, the SS7 Attack Simulator was implemented to simulate a real life scenario by generating normal and abnormal SS7 network traffic. Using the data generated by the simulator, an offline and an online experiment was conducted to provide a method of applying machine learning. To further show how operators may apply machine learning in a live SS7 environment, an Anomaly-Based Network Abuse Detection System (A-NADS) was implemented and demonstrated. Based on the results discovered in this thesis, a scientific paper was submitted to the International Conference on IT Convergence and Security 2016, which can be viewed in Appendix A.

In conclusion, this master thesis has given a demonstration of how machine learning and anomaly detection can help to detect attacks against SS7. Operators must understand that SS7 is no longer secure and there must be done more research to improve its security and reliability. Operators must deploy measures to analyze and understand the traffic being sent in SS7 networks. This knowledge must be used to incorporate countermeasures to avoid abuse of subscribers' privacy and the integrity of operators' networks.

Bibliography

[1] Association, G. 2016. The mobile economy. [Online]. Available: http://www.gsmamobileeconomy.com, Accessed: 31.05.16.

[2] Dryburgh, L. & Hewett, J. 2005. *Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services*. Cisco Press.

[3] Timberg, C. August 24 2014. For sale: Systems that can secretly track where cellphone users go around the globe. *The Washington Post*. [Online]. Available: http://www.washingtonpost.com/, Accessed: 01.10.2015.

[4] Timberg, C. December 18 2014. German researchers discover a flaw that could let anyone listen to your cell calls. *The Washington Post*. [Online]. Available: http://www.washingtonpost.com/, Accessed: 01.10.2015.

[5] Goodwin, B. August 14 2015. Security flaw exposes billions of mobile phone users to eavesdropping. *Computer Weekly*. [Online]. Available: http://www.computerweekly.com/, Accessed: 01.10.2015.

[6] Engel, T. Locating Mobile Phones using Signaling System #7. [Online]. Available: https://events.ccc.de, Accessed 07.11.2015.

[7] Langlois, P. 2010. Getting in the SS7 kingdom: hard technology and and disturbingly easy hacks to get entry points in the walled garden. [Online]. Available: http://www.hackitoergosum.org/2010/HES2010-planglois-Attacking-SS7.pdf, Accessed: 25.11.2015.

[8] Engel, T. December 2014. SS7: Locate. Track. Manipulate. [Online]. Available: https://media.ccc.de, Accessed 22.10.2015.

[9] Nohl, K. December 2014. Mobile self-defence. [Online]. Available: https://media.ccc.de, Accessed 22.10.2015.

[10] Vauboin, P.-O. & Oliveira, A. D. April 2014. Worldwide attacks on SS7 network.

[11] Jensen, K. 2016. SS7 Anomaly Detection, GitHub Repository. [Online]. Available: https://github.com/polarking/ss7-anomaly-detection.

[12] P1 Security. December 2014. SS7map: SS7 Networks Exposure. [Online]. Available: http://ss7map.p1sec.com, Accessed 24.11.2014.

[13] Mourad, H. 2015. The Fall of SS7 - How Can the Critical Security Controls Help? *SANS Institute InfoSec Reading Room*.

[14] Positive Technologies. December 2014. Signaling System 7 (SS7) Security Report. [Online]. Available: http://www.ptsecurity.com, Accessed: 31.10.2015.

[15] AdaptiveMobile. 2016. Shielding the core: An analysis of real-world attacks on the ss7 network. [Online]. Avvailabe: http://www.adaptivemobile.com/downloads/shielding-the-core, Accessed: 29.03.15.

[16] Mayer-Schonberger, V. & Strasser, M. 1998. Closer look at telecom deregulation: The european advantage. *Harv. JL & Tech.*, 12, 561.

[17] 3GPP TS 36.300. March 2016. Overall description - Stage 2 (Release 13).

[18] Telecommunication Standardization Sector of ITU. March 1993. Introduction To CCITT Signalling System No. 7. *Q.700 Specifications of Signalling System No. 7*.

[19] The International Engineering Consortium (IEC). Signaling System 7 (SS7). [Online]. Available: http://www.cs.rutgers.edu/~rmartin/teaching/fall04/cs552/readings/ss7.pdf, Accessed: 07.10.2015.

[20] 3GPP TS 29.002. December 2015. Mobile Application Part (MAP) specification (Release 13).

[21] 3GPP TS 29.078. September 2014. CAMEL Application Part (CAP) specification (Release 12).

[22] 3GPP TS 02.78. December 2001. CAMEL Service Definition - Stage 1 (Release 1998).

[23] Restcomm. 2015. jss7 github repository. [Online]. Available: https://github.com/Restcomm/jss7/, Accessed: 17.11.2015.

[24] Stewart, R. September 2007. Rfc 4960: Stream control transmission protocol.

[25] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., & Sharp, C. 1999. Rfc 2719: Framework architecture for signaling transport. *The Internet Society*.

[26] Morneault, K. & J., P.-B. September 2006. Rfc 4666: Signaling system 7 (ss7) message transfer part 3 (mtp3) - user adaptation layer (m3ua).

[27] Morneault, K., Dantu, R., Sidebottom, G., Bidulock, B., & Heitz, J. September 2002. Rfc 3331: Signaling system 7 (ss7) message transfer part 2 (mtp2) - user adaptation layer.

[28] 3GPP TS 23.002. September 2015. Network architecture (Release 13).

[29] 3GPP TS 23.040. December 2015. Technical realization of the Short Message Service (SMS) (Release 13).

[30] 3GPP TS 23.003. December 2015. Numbering, addressing and identification (Release 13).

[31] Ghorbani, A. A., Lu, W., & Tavallaee, M. *Network Intrusion Detection and Prevention: Concepts and Techniques*, chapter Detection Approaches, 27–53. Springer US, Boston, MA, 2010. URL: http://dx.doi.org/10.1007/978-0-387-88771-5_2, doi:10.1007/978-0-387-88771-5_2.

[32] Estevez-Tapiador, J. M., Garcia-Teodoro, P., & Diaz-Verdejo, J. E. 2004. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, 27(16), 1569 – 1584. URL: http://www.sciencedirect.com/science/article/pii/S0140366404002385, doi:http://dx.doi.org/10.1016/j.comcom.2004.07.002.

[33] Nguyen, H. T. *Reliable Machine Learning Algorithms for Intrusion Detection Systems*. PhD thesis, Gjøvik University College, 2012.

[34] García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1–2), 18 – 28. URL: http://www.sciencedirect.com/science/article/pii/S0167404808000692, doi:http://dx.doi.org/10.1016/j.cose.2008.08.003.

[35] Kononenko, I. & Kukar, M. 2007. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited.

[36] Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

[37] Witten, I. H. & Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[38] Harrington, P. 2012. *Machine Learning In Action*. Manning.

[39] Aggarwal, C. C. 2015. Outlier analysis. In *Data Mining*, 237–263. Springer.

[40] Chandola, V., Banerjee, A., & Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 15.

[41] Kejariwal, A. January 2015. Introducing practical and robust anomaly detection in a time series. [Online]. Available: https://blog.twitter.com/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series, Accessed: 07.05.2016.

[42] Rosner, B. 1983. Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2), 165–172. URL: http://www.jstor.org/stable/1268549.

[43] Twitter Inc. 2016. Anomaly Detection with R. [Online]. Available: https://github.com/twitter/AnomalyDetection, Accessed: 07.05.2016.

[44] Arthur, D. & Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

[45] The Apache Software Foundation. 2016. Apache Spark. [Online]. Available: http://spark.apache.org.

[46] Bahmani, B., Moseley, B., Vattani, A., Kumar, R., & Vassilvitskii, S. 2012. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7), 622–633.

[47] Rao, S. P. Analysis and Mitigation of Recent Attacks on Mobile Communication Backend. Master's thesis, University of Tartu, 2015.

[48] Inc., P. S. 2016. SCTPscan: SCTP Network and Port Scanner. [Online]. Available: http://www.p1sec.com/corp/research/tools/sctpscan/, Accessed: 09.05.2015.

[49] Biondi, P. 2015. Scapy Project Home Page. [Online]. Available: http://www.secdev.org/projects/scapy/, Accessed: 17.11.2015.

[50] 3GPP TS 23.840. March 2007. Study into routeing of MT-SMs via the HPLMN (Release 7).

[51] Jensen, K. 2016. SS7 Attack Simulator GitHub Repository. [Online]. Available: https://github.com/polarking/jss7-attack-simulator.

[52] Wireshark Foundation. 2016. Ss7 - the wireshark wiki. [Online]. Available: https://wiki.wireshark.org/SS7, Accessed: 16.05.2016.

[53] Scarfone, K. & Mell, P. 2007. Guide to intrusion detection and prevention systems (idps). *NIST special publication*, 800(2007), 94.

[54] Wireshark Foundation. 2015. Wireshark. [Online]. Available: https://www.wireshark.org, Accessed: 23.11.2015.

[55] Elastic. 2016. The Elastic Stack. [Online]. Available: https://www.elastic.co/products.

[56] Elastic. 2016. Elasticsearch. [Online]. Available: https://www.elastic.co/products/elasticsearch.

[57] Elastic. 2016. Logstash. [Online]. Available: https://www.elastic.co/products/logstash.

[58] Elastic. 2016. Kibana. [Online]. Available: https://www.elastic.co/products/kibana.

[59] The Apache Software Foundation. 2016. Apache Kafka. [Online]. Available: http://kafka.apache.org.

[60] Owen, S. Oktober 2014. A gentle introduction to apache spark and clustering for anomaly detection. [Online]. Available: http://conferences.oreilly.com/strata/stratany2014/public/schedule/detail/36035, Accessed: 07.05.2016.

[61] Barreno, M., Nelson, B., Sears, R., Joseph, A. D., & Tygar, J. D. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 16–25. ACM.

[62] Fogla, P. & Lee, W. 2006. Evading network anomaly detection systems: formal reasoning and practical techniques. In *Proceedings of the 13th ACM conference on Computer and communications security*, 59–68. ACM.

[63] Sommer, R. & Paxson, V. May 2010. Outside the closed world: On using machine learning for network intrusion detection. In *Security and Privacy (SP), 2010 IEEE Symposium on*, 305–316. doi:10.1109/SP.2010.25.

[64] European Union Agency for Fundamental Rights and the Council of Europe. April 2014. Handbook on European data protection law. [Online]. Available: http://www.echr.coe.int/Documents/Handbook_data_protection_ENG.pdf, Accessed: 17.12.2015.

[65] Lovdata. October 2015. Lov om behandling av personopplysninger (personopplysningsloven). [Online]. Available: https://lovdata.no/dokument/NL/lov/2000-04-14-31, Accessed: 17.12.2015.

[66] Flaglien, A., Franke, K., & Arnes, A. *Advances in Digital Forensics VII: 7th IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL, USA, January 31 – February 2, 2011, Revised Selected Papers*, chapter Identifying Malware Using Cross-Evidence Correlation, 169–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. URL: http://dx.doi.org/10.1007/978-3-642-24212-0_13, doi:10.1007/978-3-642-24212-0_13.

A Paper submitted to ICITCS2016

Better Protection of SS7 Networks With Machine Learning

Kristoffer Jensen*, Thanh van Do*† Hai Thanh Nguyen*†, André Årnes*†,
* NTNU, Norwegian University of Science and Technology, Norway
† Telenor ASA, Snarøyveien 30 1331 Fornebu, Norway
Email: kristoffer_jensen@icloud.com, {thanh-van.do, haithanh.nguyen, andre.arnes}@telenor.com

***Abstract*—Deregulation and migration to IP have made SS7 vulnerable to serious attacks such as location tracking of subscribers, interception of calls and SMS, fraud, and denial of services. Unfortunately, current protection measures such as firewalls, filters, and blacklists. are not able to provide adequate protections of SS7. In this paper, a method for detection of SS7 attacks using machine learning is proposed. The paper clarifies the vulnerabilities of SS7 networks and explains how machine learning techniques can help improving SS7 security. A proof-of-concept SS7 protection system using machine learning is also described thoroughly.**

***Index Terms*—mobile network security, SS7 vulnerabilities, SS7 security, cyber security, cyber attacks, machine learning based security**

I. Introduction

The Signaling System No. 7 (SS7) can be viewed as the nervous system of both the telecommunication network and the mobile communication network, which allows network elements to communicate, collaborate and deliver telecommunication services to the users. In fact, the communication networks cannot function without SS7. Originally, SS7 is inherently well protected because the communication network was operated by trusted state-owned telecom operators. With migration to IP and deregulation, it has become fairly easy for third parties to get access to the SS7 network. Unfortunately, the larger the number of individuals having access to SS7, the higher the risk of abuse. Indeed, SS7 is carrying information such as user subscription information, user location, and short messages, which has high value and is consequently attractive to attackers. Recent frauds and abuses on SS7 networks show that current security measures are not adequate and better protection methods are urgently needed.

There has been done considerable work that reveal the weaknesses of SS7 and propose protection measures but none of them suggests the usage of machine learning in the improvement of SS7 security.

The SRLabs [1] in Berlin led by the famous German Cryptographer and security researcher Karsten Nohl has considerable activities related to detection of mobile phone tapping and location tracking of the user by taking advantage of the SS7 vulnerabilities. SRLabs has a collection of tools for the assessment of SS7 network security.

P1 Security (Priority One Security) [2] is a company led by Philippe Langlois, a well-known security expert, which is dedicated to providing top security products and services for high-expertise security areas. P1 Security has a Telecom Security Task Force, which is a research think tank and consulting network in the telecom sector.

With the company Sternraute [3], Tobias Engel works on solutions for increasing privacy and data protection. As his company specializes in telecommunications networks, in particular SS7 technology, mobile phone companies worldwide draw on his expertise. He has identified serious SS7 vulnerabilities.

This paper presents an innovative protection system for SS7 networks using machine learning, starting with an overview of the vulnerabilities and threats in SS7 networks. The main part of the paper will concentrate on clarifying how machine learning can help improving SS7 security. The focus will be on elucidating what makes machine learning superior to other techniques. Last but not least, a proof-of-concept SS7 protection using machine learning is described thoroughly.

II. Threats and Vulnerabilities of SS7 Networks

There have been several reports on the vulnerabilities of SS7, both from researchers publishing their findings and from media attention [4]–[6]. Researchers and companies have publicly disclosed several attacks and entry points to the network. In this section, the currently known attacks and what makes them possible are highlighted.

A. The garden walls are gone

The security of SS7 originally relied on the principle of the walled garden, meaning a closed space where every party is trusted. For a long time, this has been an essential perspective for both telecommunication operators and the corresponding technology. Every operator was trusted, no one had any dishonest intentions. Due to deregulation and the industries' move to IP, this is no longer the case. It is now easier than ever to become an operator, for example by becoming a mobile virtual network operator (MVNO) and therefore gain access to the previously closed SS7 network.

B. Why are attacks possible?

As SS7 relies on trust between parties and elements of the core network (CN), the standards do not specify any authentication between nodes nor any other security controls. The SS7 application protocols are crucial to provide service to subscribers, so the messages must flow freely between

This paper was submitted to the International Conference on IT Convergence and Security 2016 (http://icatse.org/icitcs/) and is currently undergoing review. Accepted papers will be peer-reviewed and published in IEEE Xplore. The review process will end on the 10th of June, 2016.

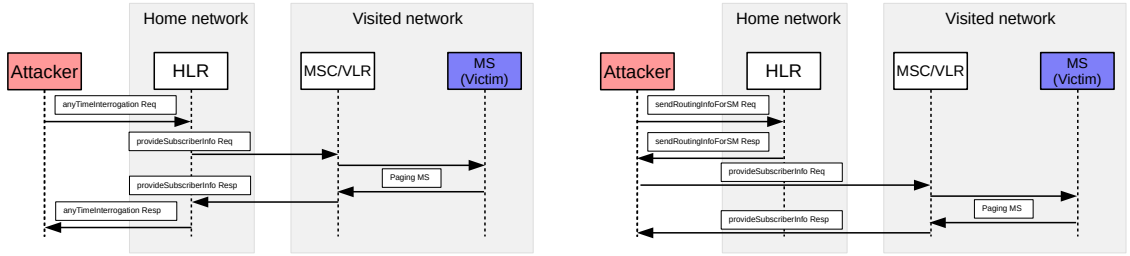


Figure 1: (a) Tracking a subscriber using the MAP anyTimeInterrogation message. (b) Tracking a subscriber using the MAP provideSubscriberInfo message.

operators. Therefore, every message sent in the network is considered legal, especially between roaming partners. Because of the walled garden approach, there has not been a strong requirement to provide extensive security in SS7 based on the fact that every party has been considered trusted.

C. Publicly disclosed attacks

Several attacks have been publicly disclosed in the course of the recent years, made possible by the lack of authentication between nodes and more easily obtained access to the SS7 network. To be able to carry out attacks on SS7, an attacker must usually require some preliminary capabilities, an attacker must: (1) be connected to the SS7 network, (2) be able to generate arbitrary messages, and (3) be able to imitate an element in the core network by providing SS7 capabilities [7]. Attackers are possibly able to track the location of subscribers, intercept calls and SMS, commit fraud, and deny service to subscribers.

To be able to perform attacks in the SS7 network, the attacker must have access to the network by some manner. There are several tactics that can be utilized to find an entry point to SS7, for example: (1) purchase access on the black market, (2) gain entry via a misconfigured node made accessible on the internet, or (3) gain unauthorized access to a 3G femtocell [8].

To clarify how the attacks can be carried out we will describe the two most discussed and menacing attacks, namely location tracking and interception.

1) Location Tracking: An attacker with access to the SS7 network would be able to track the location of the users by using a set of Mobile Application Part (MAP) messages sent to various core network (CN) elements. The goal of this type of attack is to get the identification number of the subscriber's currently connected cell. Based on the cell ID, an attacker is able to track a subscriber with accuracy down to street level in some urban areas.

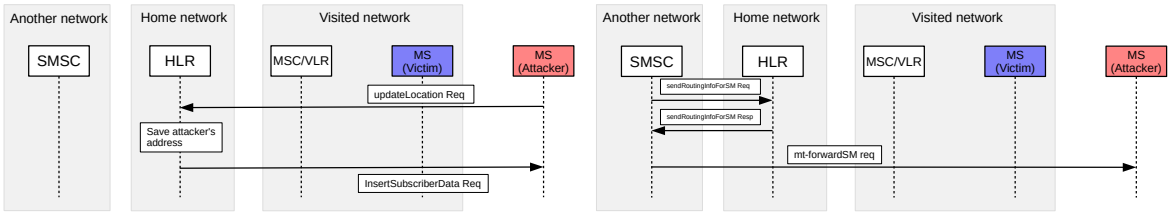
a) Using anyTimeInterrogation: The anyTimeInterrogation (ATI) message is used between the GSM Service Control Function (gsmSCF) and the Home Location Register (HLR) to gain access to information about a given subscriber [9]. An attacker can pose as a gsmSCF and send the ATI message to the HLR which triggers the HLR to send a provideSubscriberInfo (PSI) message to the currently serving Visitor Location Register (VLR). The VLR will then page the Mobile Station (MS) to get its current information. The information acquired includes subscriber's location information, including the currently connected Cell-ID and the state of the subscriber. The message flow of the ATI message can be seen in Figure 1a [10].

b) Using provideSubscriberInformation: If the ATI message has been blocked by the HLR, the attacker can circumvent the HLR by sending a PSI message directly to the VLR. To be able to send a PSI message, the attacker must first acquire the International Mobile Subscriber Identity (IMSI) of the victim and the address of the currently serving VLR which are mandatory parameters used in the PSI message. These mandatory parameters can for example be obtained by first sending a sendRoutingInformationForSM (SRI-SM) message to the subscribers HLR [10]. The message flow is shown in Figure 1b.

2) Interception: Interception refers to the activity of gaining knowledge and data originally intended for another party. It is one of the most devastating attacks against a subscriber, as the attacker might be able to record important conversations, read one-time passwords, obtain information about subscriber activities, and possibly access confidential information not intended for third parties. Using a set of MAP messages paired with technologies such as CAMEL, an attacker will be able to intercept subscriber's SMS (Short Message Service) and calls.

a) Intercepting SMS: The MAP updateLocation message is part of the mobility services provided by the CN. The CN's mobility services help ensure that the network keeps track of where subscribers are located at every moment so that data can be routed accordingly. The MAP updateLocation message in particular is used to notify the HLR that a subscriber has moved to a new MSC/VLR area [9].

An attacker can intercept SMS by notifying the HLR that the victim subscriber has moved to an MSC/VLR area controlled by the attacker, as shown in Figure 2a. When an SMS destined for the victim subscriber is sent, the Short Message Service Center (SMSC) will look up where the subscriber is currently



(a) (b)

Figure 2: (a) Stealing a subscriber using the MAP updateLocation message. (b) SMS meant for the subscriber is now sent to the attacker instead.

located using SRI-SM so that the SMS can be routed to the subscriber.

Now the SMS intended for the subscriber is sent to the attacker, as shown in Figure 2b. The attacker is now controlling the message and can store it, alter it, and possibly send it to the the original subscriber. This attack can be particularly dangerous when combined with one-time passwords which may be issued from banks or social media. Giving the attacker access to the victims account and be able to commit even worse criminal activities [10].

b) Intercepting calls by decrypting radio traffic: When a subscriber moves from one MSC/VLR area to another, the MSC initiates a handover process which aims to provide continuous service to the subscriber without dropouts and stuttering in calls. In order to provide this service, the MSCs must exchange cryptography key information to be able to encrypt the radio traffic between the operator and the MS [11].

An attacker in close proximity to a victim's MS can capture the encrypted radio traffic, while simultaneously using SS7 to obtain the encryption keys. The keys can be acquired using the MAP sendIdentification message, which is normally used between the new VLR and the old VLR when a subscriber changes MSC/VLR areas. Using the encryption keys, an attacker can decrypt the captured radio traffic and listen in on a call or decrypt other data communicated between the CN and the MS [11].

III. Challenges in Detecting Attacks on SS7

To protect their networks, mobile operators must recognize the fact that SS7 is no longer secured. It is hence necessary to separate their home SS7 portion from the global network and provide adequate protection. As shown in Figure 3, the outermost Signaling Transfer Point (STP) of the Home SS7 network can perform border control of the traffic entering and exiting the home SS7.

From the attacks disclosed by researchers it is possible to identify and classify the SS7 MAP messages that are used, into three categories as follows [12]:

- **Category 1:** Messages that have no legitimate use case for external exposure: sendIdentification (SI) – anyTimeInterrogation (ATI) – anyTimeModification (ATM) – provideSubscriberLocation (PSL).
- **Category 2:** Messages that have no legitimate need to be exposed externally for the operator's own subscribers, but can be received for other operator's roaming subscribers. These are the following: provideSubscriberInformation (PSI) – insertSubscriberData (ISD) + gsmSCF – deletedSubscriberData (DSL).
- **Category 3:** Messages that have legitimate need for external exposure. These are the following: updateLocation (UL) – sendAuthenticationInfo (SAI) – registerSS – eraseSS – processUnstructuredSS (PSU) – cancelLocation (CL) - sendRoutingInformation(SRI-SM, SRI-LCS)

For category 1 messages, a simple filter can be used to identify and block them to prevent attacks.

For category 2, more advanced filters using the correlation between roaming users and their home operators can be employed to block unwanted messages. Unfortunately, such filtering will not be able to protect roaming users.

For category 3, there is unfortunately no usable filter to detect attacks because complex correlations with further information on the current user state , e.g., last cell ID. Indeed, the signatures for attacks using category 3 messages can hardly be determined.

In brief, providing adequate protection against attacks using category 2 and 3 messages is quite challenging. To be able to detect them, one must acquire additional information about the behavior of subscribers and network elements and look for deviations in their normal behavior. This is precisely where machine learning can come to the rescue. Machine learning can be a tool that assists in anomaly detection, where we aim to solve the problem of detecting patterns in data that deviate from the expected normal behavior [13]. Indeed, machine learning can help us to build profiles of normal behavior such that unexpected subscriber and network behavior can be detected. These anomalies will help indicate when and if and attack has been launched against a subscriber.

IV. An SS7 Protection Proof-of-Concept

To demonstrate the feasibility of using machine learning as a mechanism to detect anomalies in SS7 traffic, a proof-of-concept implementation was created using open source software. Due to privacy concerns when obtaining and using

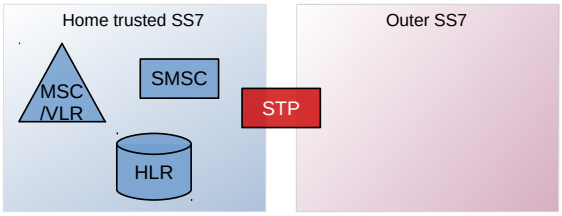


Figure 3: Separating the home SS7.

SS7 network data, we did not access a real working SS7 network nor any real SS7 data as part of our experiments.

To solve this problem, we developed the SS7 Attack Simulator based on RestComm's jSS7, a free and open source implementation of the SS7 stack made in Java [14]. The attack simulator was used to generate SS7 network traffic that we analyzed using machine learning techniques. This analysis provided us with a measure to explore the feasibility of machine learning using an anomaly detection approach. The source code for the attack simulator was released under a free license and is publicly available online on GitHub [15]. In our proof-of-concept, we focused on the intercept SMS attack as described in section II.

A. The SS7 Attack Simulator

We used the simulator to create a dataset that could be used to test machine learning algorithms as a tool to detect attacks on an SS7 network. The attack simulator was created by forking RestComm's jSS7, and it was altered in such a way that it was capable of performing some standardized normal CN procedures and some of the publicly disclosed attacks against CN elements. The attack simulator was created to serve two distinct purposes: (1) simulate the message flow of publicly disclosed attacks, and (2) simulate an SS7 network containing normal and abnormal network traffic that can be analyzed using machine learning techniques.

1) Simulator capabilities: The simulation consists of three SS7 networks, each representing an operator connected to the international SS7. Each operator has a set of functional CN entities which communicate using the SIGTRAN stack. An overview of the simulated network can be seen in Figure 4. These entities communicate using the MAP application protocol as defined in 3GPP's MAP technical specification [9]. A number of thirteen procedures were selected to represent the normal traffic, or background noise, which did not include any form of attacks. These procedures where chosen based on the capabilities of the jSS7 stack and their ease of implementation. Normal procedures generated by the SS7 attack simulator is chosen at random based on the inbuilt Java pseudorandom generator. All messages generated by the simulator conform to the mandatory requirements set by the MAP technical specification.

In addition to the normal standardized MAP procedures, three attacks were implemented and used to attack the simulated subscribers belonging to operator A. The attacks implemented was: (1) location tracking using anyTimeInterrogation, (2) location tracking using provideSubscriberInfo, and (3) intercepting SMS by stealing subscribers with updateLocation. These attacks were simulated as being launched from an entity controlled by an attacker located in operator C's network.

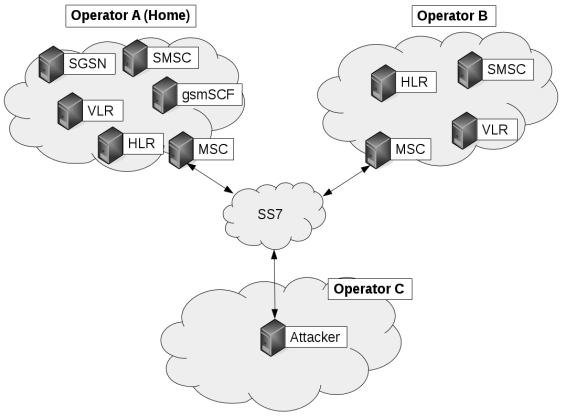


Figure 4: Overview of the simulated SS7 network.

The entities generate normal traffic based on a common set of subscribers, which number is defined by a user set option. Each subscriber is defined based on parameters such as IMSI, MSISDN, current location, and current serving MSC/VLR. The entities contained in the simulator perform operations on these subscribers so that messages are routed accordingly, making the simulator a dynamic system where most of the messages have an effect on the state of the subscribers and the network itself.

2) Simulating a real life scenario: To test machine learning's capabilities in an SS7 network on a real life scenario, we defined the notion of a Very Important Person (VIP) as one of the subscribers belonging to operator A. This subscriber is assumed to be a high value target for an attacker and in real life could for example be a CEO of a large company or an important politician. Therefore, the VIP is the only target of the attacks in the simulated network.

In our simulations, we assume that the VIP subscriber is traveling through a number of areas when going to work in the morning, and the same areas in the opposite direction when he/she is going home in the afternoon. These areas are defined by their own Location Area Code (LAC) which is used in telecom networks to indicate where a subscriber is located to route data to the subscriber accordingly. We simulate that the VIP is moving at frequent patterns as seen by the MAP updateLocation requests issued by the MSC to the HLR. Every day, the VIP is moving through three location areas when going to work in an area covered by operator B, and moving through three areas when going home to the area covered by operator A. An example of these frequent movements are visually shown in Figure 5, where the subscriber is traveling

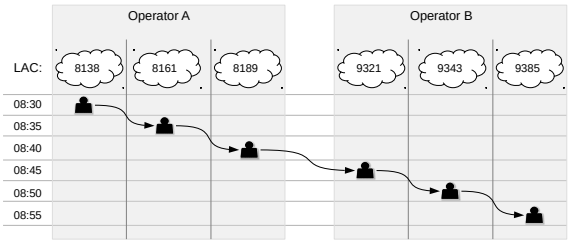


Figure 5: Example of a subscriber moving through different location areas at different times, indicated by their location area code (LAC).

to work in the morning through different location areas.

B. Detecting SS7 attacks using machine learning

We deployed machine learning techniques using the dataset obtained from our simulation in an attempt to detect the attacks against the VIP subscriber. A set of features was selected to be used as input to the machine learning algorithm. These features where used to create a subscriber profile that can be used to detect attacks against subscribers indirectly using SS7. Even though our efforts where focused on the intercept SMS attack, we believe that the same approach can be used to create features that effectively can capture the abnormal behavior introduced by the other attacks.

1) Selected features: To be able to detect the intercept SMS attack, we propose a set of features that can be used to distinguish between normal and abnormal subscriber behavior. These features where selected based on common approaches used in anomaly detection, and some features specifically for SS7 network traffic. The selected features can be seen in Table I.

To be able to use these features, we make some assumptions on the capabilities of an operator, the operator must be able to:

- Gather all SS7 network traffic being sent in its core network. By, for example, using sensors strategically placed in the network.
- Distinguish where an SS7 message originated.
- Devise a metric which describes the distance between different location areas in the network. To be used as the distance traveled by a subscriber within a time frame.

If an operator possesses these capabilities, it will be possible to detect anomalies in how a particular subscriber moves geographically. As seen in our simulations, the VIP subscriber will move at a steady pace every day. When an attack occurs using the updateLocation message, depending on where the "new" location of the subscriber is, an anomaly will occur as the subscriber will possibly move a great distance in a short amount of time. One might think to set a threshold to detect this abnormal behavior. But because of the dynamic nature in subscriber behavior, this approach will not always be applicable. It is therefore hard to manually define the threshold

Table I: Features Selected to Detect Anomalies in Subscriber Behavior

Description	Variable type
Time since last location update	Continuous
Distance traveled since last location update	Continuous
Byte length of location update	Continuous
Frequency of location updates	Continuous
Message network origin	Nominal

for so many subscribers, and further define rules that can be applied to all of them.

We propose to do the analysis of user behavior by building user profiles, based on the fact that attackers add abnormalities in the behavior of users. In this case, normal behavior of the user can be seen by analyzing past behavior. The challenge is how to represent, or generalize, normal user behavior and how to use this for anomaly detection. To solve this problem, machine learning (ML) provides an automatic way to learn user behavior and represent user profiles mathematically. ML can also handle the dynamic nature of user behavior well. We can then use this learnt model to detect anomalies [13].

Using ML, what we are essentially doing in order to detect anomalous behavior is to first build a distribution function of user's normal behavior. Then use the built model to detect the behavior that highly deviate from the normal one. This shifts the focus on the statistical approaches instead of many other ML approaches for anomaly detection [13].

As a proof-of-concept, we selected the Seasonal Hybrid ESD algorithm for our experiments. This is also because the algorithm was successfully applied by Twitter to detect the trend and seasonal anomalies in the data [16]. The details on this algorithm is given in the following paragraphs.

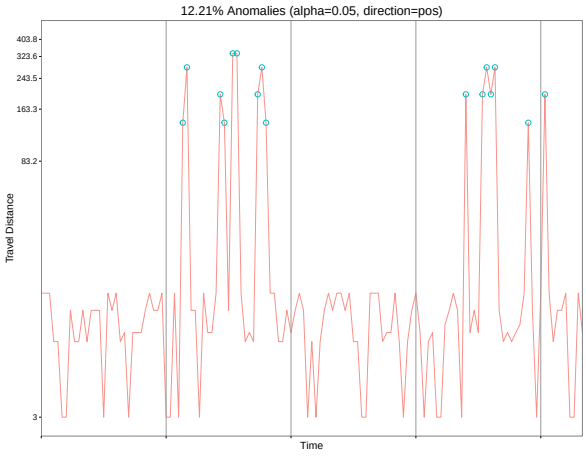


Figure 6: The results of using Twitter's AnomalyDetection on part of our dataset. Showing that anomalies in the travel speed of a subscriber is detectable.

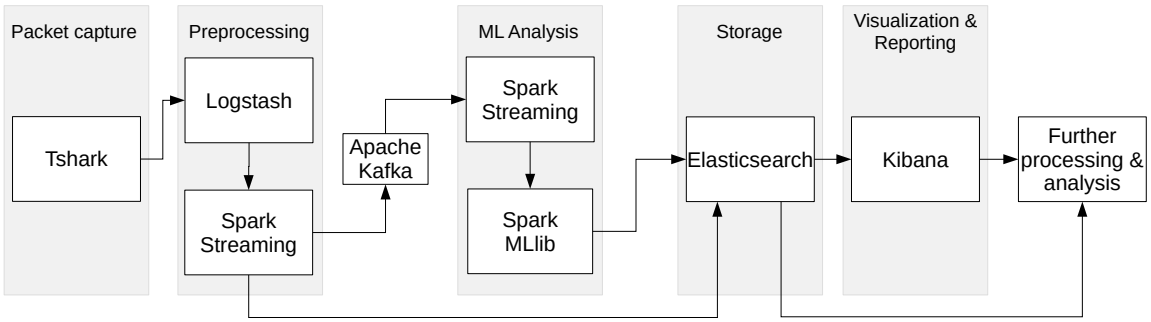


Figure 7: Components of the implemented anomaly-based network abuse detection system.

2) The Seasonal Hybrid ESD algorithm: The S-H-ESD algorithm has been implemented by Twitter in the R language [17] and provides a tool to discover statistically meaningful anomalies based on the input vector [16].

The S-H-ESD builds on the ESD test, which can be used to detect a single outlier in a dataset by finding the point furthest away from the mean of the dataset. By computing $G = \frac{|Y_i - \bar{Y}|}{s}$, where $\bar{Y}$ is the mean of the dataset and s is the standard deviation of the dataset. If G is larger than the critical value, the point is an outlier [18]:

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-1}}{N-2+t^2_{\alpha/(2N),N-2}}} \quad (1)$$

, where $t^2_{\alpha/(2N),N-1}$ is the upper critical value of the t-distribution with $N-2$ degrees of freedom and a significance level of $\alpha/(2N)$.

To test for multiple outliers we can use the Generalized ESD algorithm, where it is assumed that there can be up to r outliers. The algorithm works by iterating the dataset and removing the point with the highest G value calculated from the dataset's new mean and standard deviation. The critical values λ will change with every removed point from the dataset [18]:

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i+1+t^2_{p,n-i-1})(n-i+1)}} \quad (2)$$

To decide whether a point is an anomaly or not, the following rule applies: if all of the test statistics are lower than the critical values, there are no anomalies. On the other hand, if any of the test statistics are greater than the critical value, the largest number of points so that the associated test statistic is greater than the critical value are removed as outliers [18].

The Generalized ESD algorithm assumes that the dataset is normally distributed [18], and as real data might include some seasonality it cannot directly be applied to our dataset. The S-H-ESD algorithm solves this problem by applying R's Seasonal Decomposition of Time Series by Loess (STL) library. STL is used to decompose the data into a seasonal part, a trend part, and the remaining data using local regression (LOESS). LOESS fits a low order polynomial to a subset of the data and merge them together by weighing them. As the trend and seasonal part can be removed using LOESS, the remaining data will be close to normally distributed. Then the Generalized ESD can be applied on the remaining data to detect anomalies [17].

The S-H-ESD implementation splits data into chunks of length *period*, which is analyzed for a maximum number of anomalies *max_anoms*. The statistical significance used to accept or reject an anomaly is given with the option *alpha*, it is also possible to specify the direction in which anomalies should be detected using the *direction* option.

3) Offline test results: Our results can be seen in Figure 6, where it is indicated where anomalies in the form of distance traveled by the subscriber is discovered. These anomalies indicate that an attack using the updateLocation message has been launched towards the subscriber. Using a dataset of 59682

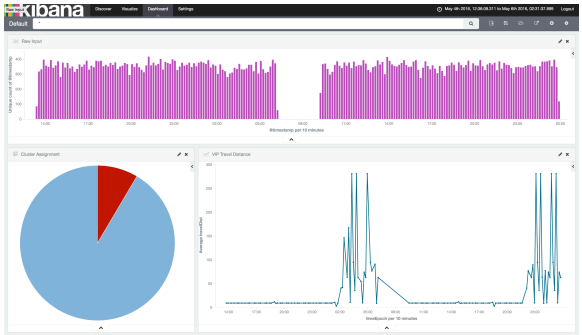


Figure 8: Example of a real time analytics dashboard monitoring SS7 traffic using Kibana. Showing the input flow to the system, the cluster assignments and the distance traveled by the VIP subscriber.

samples, running the algorithm took about 4 minutes and 24 seconds on a modern dual core laptop. We experienced the expected **100%** accuracy in detecting attacks based on the traveling speed of the VIP. On the other hand we experienced the false positive rate to be about **4.7%**. This high number is once again expected as we will see an anomaly when the subscriber is moving rapidly when being attacked, but also when "returning" home indicated by a new updateLocation message. Causing a minimum of two anomalies with every launched attack.

C. An SS7 anomaly-based network abuse detection system

As part of our contribution we demonstrate a prototype of a fully functional anomaly-based network abuse detection system (A-NADS), which is based on free and open source software. This system was implemented using readily available big data software like tshark [19], the ELK stack [20], Apache Kafka [21] and Apache Spark [22]. All of these technologies can be described with a few common keywords: high availability, scalable, and highly customizable applications that fit perfectly in a big data environment with high demands for performance, throughput and storage. The implementation and a guide to its use can be found online in a Git repository on GitHub [23].

The A-NADS is specifically built to analyze SS7 network traffic, using the tools to capture packets, preprocess the information, extract relevant features, and analyze the features for further applying machine learning. The overview of the different technologies used, and the information flow of the system, can be seen in Figure 7.

The ELK stack provides the capability of displaying relevant live information using its dashboards. This can for example be used in a real time analysis operating center that continuously monitors network activity. An example of a dashboard displaying information about the simulated network can be seen in Figure 8.

V. Conclusion

In this paper we have described some of the current threats towards the SS7 networks, and its current lack of protection due to deregulation and the industries move to IP. We propose that machine learning can be applied as a tool to improve the security of SS7 and further secure the subscribers indirectly using SS7. Machine learning paired with a well selected set of features has been shown to be an excellent approach to solving the security issues based on the nature of SS7. As it is difficult to stop attacks by using more common approaches such as firewalls or signature detection systems.

Our claims where tested by developing a simulator capable of simulating a larger SS7 network. In this simulator we generated both normal and attack traffic, and made an effort to detect the attacks by using anomaly detection algorithms and implementing a real time anomaly-based abuse detection system using open source software.

We believe that a similar approach can be used to improve the security of a real life SS7 network, possibly mitigating the threats and vulnerabilities against SS7 which have been recently disclosed. This can be done by carefully selecting additional features that will be able to model the behavior of subscribers and network elements.

Furthermore, the attack simulator that we developed can be further extended and used as part of a testbed to improve other SS7 protection systems and improve the overall resilience and security of SS7. As future work we would like to see a similar solution implemented and tested in an operational SS7 network, to see if our claims hold there.

References

[1] Security Research Labs, "SRLabs Open Source Projects," 2016, [Online]. Available: https://opensource.srlabs.de.

[2] P1 Security S.A.S, "Priority One Home Page," 2016, [Online]. Available: http://www.p1sec.com/corp/.

[3] Sternraute, "Sternraute Home Page," 2016, [Online]. Available: http://www.sternraute.de.

[4] C. Timberg, "For sale: Systems that can secretly track where cellphone users go around the globe," *The Washington Post*, August 24 2014, [Online]. Available: http://www.washingtonpost.com/.

[5] C. Timberg, "German researchers discover a flaw that could let anyone listen to your cell calls." *The Washington Post*, December 18 2014, [Online]. Available: http://www.washingtonpost.com/.

[6] B. Goodwin, "Security flaw exposes billions of mobile phone users to eavesdropping," *Computer Weekly*, August 14 2015, [Online]. Available: http://www.computerweekly.com/.

[7] Positive Technologies, "Signaling System 7 (SS7) Security Report," December 2014, [Online] Available: http://www.ptsecurity.com.

[8] P. Langlois, "Getting in the SS7 kingdom: hard technology and and disturbingly easy hacks to get entry points in the walled garden." 2010, [Online]. Available: http://www.hackitoergosum.org/2010/HES2010-planglois-Attacking-SS7.pdf.

[9] 3GPP TS 29.002, "Mobile Application Part (MAP) specification (Release 13)," December 2015.

[10] T. Engel, "SS7: Locate. Track. Manipulate," December 2014, [Online]. Available: https://media.ccc.de.

[11] K. Nohl, "Mobile self-defence," December 2014, [Online]. Available: https://media.ccc.de.

[12] H. Mourad, "The Fall of SS7 - How Can the Critical Security Controls Help?" *SANS Institute InfoSec Reading Room*, 2015.

[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[14] Restcomm, "jss7 GitHub Repository," 2015, [Online]. Available: https://github.com/Mobicents/jss7/.

[15] K. Jensen, "SS7 Attack Simulator GitHub Repository," 2016, [Online]. Available: https://github.com/polarking/jss7-attack-simulator.

[16] A. Kejariwal, "Introducing practical and robust anomaly detection in a time series," January 2015, [Online]. Available: https://blog.twitter.com/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series.

[17] Twitter, Inc, "Anomaly Detection with R," 2016, [Online]. Available: https://github.com/twitter/AnomalyDetection.

[18] B. Rosner, "Percentage points for a generalized esd many-outlier procedure," *Technometrics*, vol. 25, no. 2, pp. 165–172, 1983. [Online]. Available: http://www.jstor.org/stable/1268549

[19] Wireshark Foundation, "Wireshark," 2015, [Online]. Available: https://www.wireshark.org.

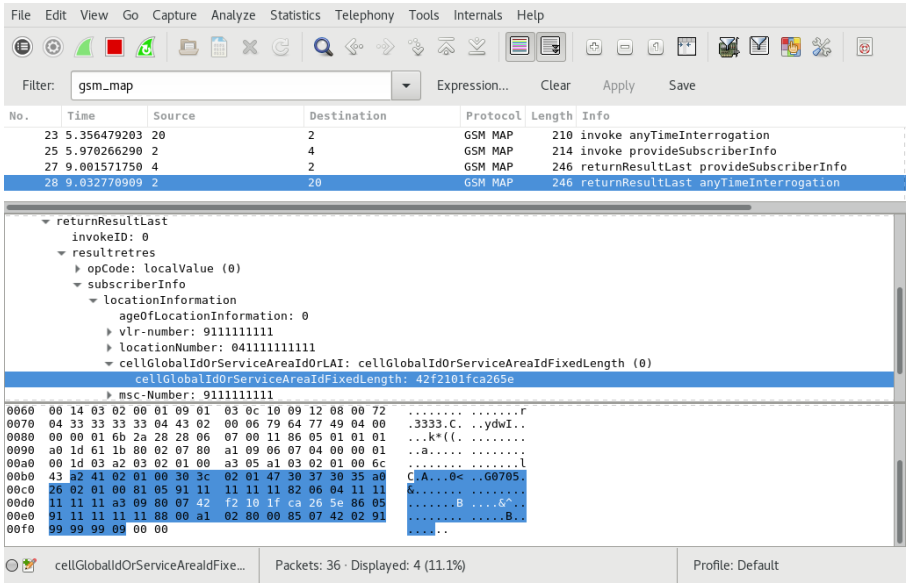
[20] Elastic, "The Elastic Stack," 2016, [Online]. Available: https://www.elastic.co/products.

[21] The Apache Software Foundation, "Apache Kafka," 2016, [Online]. Available: http://kafka.apache.org.

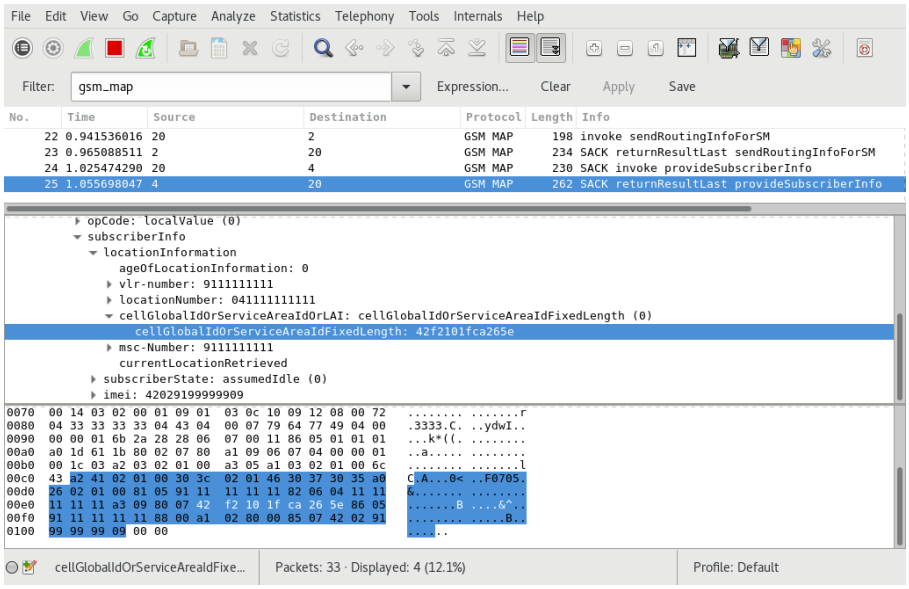
[22] The Apache Software Foundation, "Apache Spark," 2016, [Online]. Available: http://spark.apache.org.

[23] K. Jensen, "SS7 Anomaly Detection GitHub Repository," 2016, [Online]. Available: https://github.com/polarking/ss7-anomaly-detection.

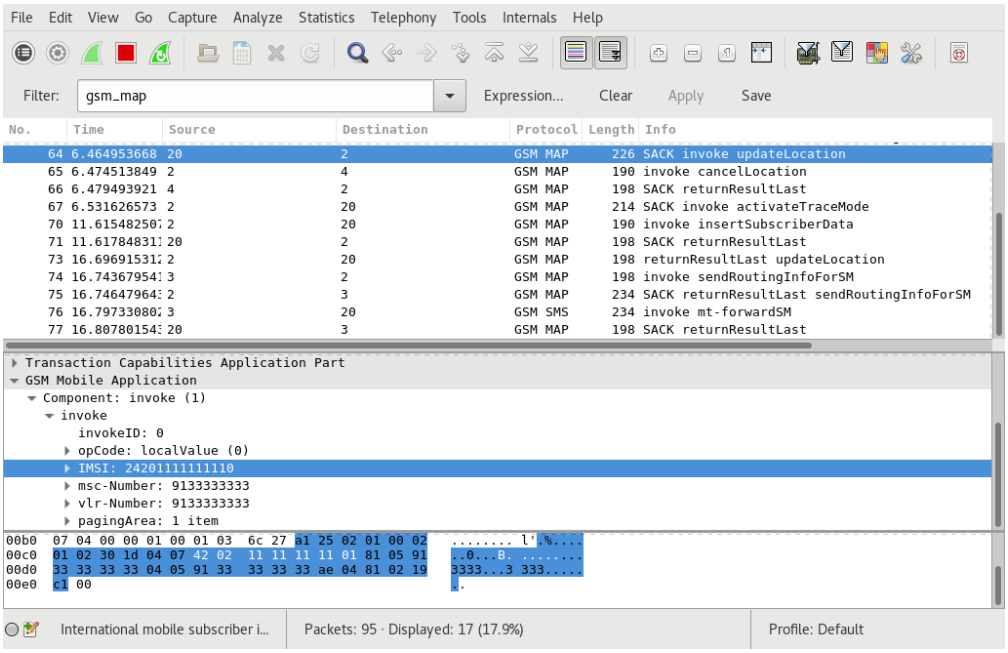
B Screen shots of the running SS7 Attack Simulator



Wireshark capture of the tracking attack using the MAP anyTimeInterrogation message.



Wireshark capture of the location tracking attack using the MAP provideSubscriberInfo message.



Wireshark capture of the intercept SMS attack using the MAP updateLocation message.

C A-NADS - technical details

The implemented Anomaly-Based Network Abuse Detection System (A-NADS) is explained in more technical detail in this appendix. These details includes selected code fragments from the implementation and some examples of data used in the experiments.

C.1 Packet capture using tshark and logstash

Output from tshark is continuously piped to logstash. Logstash handles the input by filtering it into comma separated values and sending them to an Apache Kafka topic. For further reference, the raw network traffic captured by tshark is stored in Elasticsearch. The following listing shows the configuration file read by logstash.

```
input {
  pipe {
    tags => tshark
    command => "tshark -Q -B 30 -f sctp -i lo -Y gsm_map
        -T fields \
                  -e m3ua.protocol_data_opc \
                  -e m3ua.protocol_data_dpc \
                  -e frame.time_epoch \
                  -e _ws.col.Length \
                  -e _ws.col.Info \
                  -e sccp.calling.digits \
                  -e sccp.calling.ssn \
                  -e sccp.called.digits \
                  -e sccp.called.ssn \
                  -e gsm_map.imsi \
                  -e gsm_map.ms.imsi \
                  -e gsm_map.ch.imsi \
                  -e gsm_map.sm.imsi \
                  -e gsm_map.om.imsi \
                  -e gsm_map.address.digits \
                  -e gsm_map.tbcd_digits \
                  -e gsm_map.ms.msc_Number \
                  -e gsm_map.ms.lac \
                  -E separator=, \
                  -E quote=d \
                  -E occurrence=f"
  }
```

```
}

filter {
  csv {
    autogenerate_column_names => false
    columns => [
      "opc",
      "dpc",
      "time_epoch",
      "length",
      "message",
      "cggt",
      "cgssn",
      "cdgt",
      "cdssn",
      "imsi",
      "imsi-ms",
      "imsi-ch",
      "imsi-sm",
      "imsi-om",
      "address-digits",
      "tbcd",
      "msc-number",
      "lac"
    ]
  }
}

output {
  kafka {
    topic_id => "ss7-raw-input"
    codec => plain {
      format => "%{time_epoch},%{opc},%{dpc},%{length
        },%{message},%{cggt},%{cdgt},%{imsi},%{imsi-ms
        },%{imsi-ch},%{imsi-sm},%{imsi-om},%{address-
        digits},%{tbcd},%{msc-number},%{lac}"
    }
  }
  elasticsearch {
    index => "ss7-raw-input"
```

```
64      user => "****"
65      password => "****"
66    }
67 }
```

C.2 Preprocessing using Spark Streaming

Preprocessing of the input is done to extract features from the raw network traffic. Apache Spark Streaming is a library capable of streaming from several inputs, in this case from an Apache Kafka topic. The Spark Streaming Context defined as *ssc* will continuously check for new messages on the specified topic in which logstash submits the raw network traffic. The following listing is a cutout of the most important part of the preprocessing. This is an Apache Spark application implemented in the Scala language. In brief, the application reads raw traffic from an Kafka topic, creates features based on its contents, stores the preprocessed data in Elasticsearch, and sends the preprocessed data on another Kafka topic. Every process is commented by the use of the comment notation "//".

```
1 //Stream messages from Kafka: network capture
2 KafkaUtils.createDirectStream[String, String, StringDecoder,
    StringDecoder](ssc, kafkaParams, topics)
3   .flatMap(_._2.split("\n")).foreachRDD(ss7Record => {
4     val ss7Input = ss7Record.collect()
5     ss7Input.foreach(input => {
6       val line = input.split(",")
7       val mapMessage = line(4)
8
9       //Only interested in updateLocation requests
10      if(mapMessage.contains("invoke updateLocation") &&
          !mapMessage.contains("returnResultLast")) {
11        val imsi = line(13)
12
13        //Looking for location updates for the VIP subscriber
14        if(imsi == "24201111111110") {
15          val timeEpoch = line(0).trim.toDouble
16          val byteLength = line(3).trim.toDouble
17          val lastUpdate = timeEpoch - prevLocUpdate.timeEpoch
18          val newLac = LAC.lacDecode(line(15).trim)
19          val travelDist =
20            if (prevLocUpdate.prevLac == 0.0) //If it's the first
                read updateLocation.
21              travelDistance(newLac, newLac)
```

```
22            else
23              travelDistance(newLac, prevLocUpdate.prevLac)
24
25          prevLocUpdate = LocationUpdate(timeEpoch, byteLength,
                travelDist, lastUpdate, newLac)
26
27          val preProcessedDate = Map(
28            "timeEpoch" -> new Date(timeEpoch.toInt * 1000L),
29            "byteLength" -> byteLength.toInt,
30            "newLac" -> newLac,
31            "lastUpdate" -> lastUpdate,
32            "travelDist" -> travelDist,
33            "label" -> label
34          )
35
36          //Store preprocessed values in elasticsearch for further
                analysis and visualization
37          val preProcRDD = sc.makeRDD(Seq(preProcessedDate))
38          preProcRDD.saveToEs("ss7-ml-preprocessed/preprocessed")
39
40          //Send preprocessed data on Kafka for ML analysis
41          val kafkaOutString = timeEpoch.toString + "," +
                label.toString + "," + byteLength.toString + "," +
                lastUpdate.toString + "," + travelDist.toString + ","
                + newLac.toString
42          kafkaSink.value.send("ss7-preprocessed", kafkaOutString)
43
44            label += 1
45          }
46        }
47      })
48    })
```

C.3 Machine learning using Spark MLlib

Machine learning is done using the Apache Spark Machine Learning library (MLlib) which includes several machine learning algorithms and techniques. In order to train the k-means|| algorithm, already preprocessed features is read from a specified text file. Using Spark Streaming, preprocessed data is then read from an Apache Kafka topic and used as input to the k-means|| algorithm. Scores from clustering the new input features is continuously stored in Elasticsearch.

The following listing shows the most important part of this Apache Spark application.

```
1 //Read training data from text file.
2 val trainingData = sc.textFile(trainingDataPath).map(line =>
      extractFeatures(line))
3
4 //Creating a standardizer object that scales features.
5 val scaler = new StandardScaler().fit(trainingData)
6
7 //Scale training data.
8 val scaledTrainingData = trainingData.map(d =>
      scaler.transform(d)).cache
9
10 //Train K Means model on existing data contained in regular file
11 val kmeans = new KMeans().setK(numClusters).run(scaledTrainingData)
12 val threshold = 0.2 //Threshold used to detect outliers.
13
14 //Start stream to read from Kafka
15 val messages =
      KafkaUtils.createDirectStream[String,String,StringDecoder,StringDecoder](ssc,
      kafkaParams, topics)
16
17 //Process each message from Kafka, extract features and make
      prediction
18 messages.flatMap(_._2.split("\n")).foreachRDD(rdd => {
19   rdd.collect.foreach(message => {
20     //Expected features:
          timeEpoch,label,byteLength,lastUpdate,travelDist,newLac
21     val split = message.split(",")
22     val timeEpoch = split(0).toDouble
23     val label = split(1).toInt
24     val byteLength = split(2).toDouble
25     val lastUpdate = split(3).toDouble
26     val travelDist = split(4).toDouble
27     val newLac = split(5).toInt
28
29     val isInternal = isInternalMessage(newLac)
30     val internalMsg = if (isInternal) 1 else 0
31     val externalMsg = if (isInternal) 0 else 1
32
```

```
33    val point = LabeledPoint(label,
          scaler.transform(Vectors.dense(byteLength, lastUpdate,
          travelDist, internalMsg, externalMsg)))
34
35    val distScore = distToCentroid(point.features, kmeans)
          //Checking this points distance to the centroid
36
37    val esMap = Map("timeEpoch" -> new Date(timeEpoch.toInt *
          1000L), "label" -> point.label, "score" -> distScore)
38
39    if (distScore > threshold) { //Possible anomaly.
40      sc.makeRDD(Seq(esMap)).saveToEs("ss7-ml-results/anomaly")
41    } else { //Possible normal traffic.
42      sc.makeRDD(Seq(esMap)).saveToEs("ss7-ml-results/normal")
43    }
44  })
45 })
```

C.4 Examples of data

```
"19823","1458317350.900747000","4","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19824","1458317350.901078000","4","1","GSM MAP","182","returnResultLast ",,"7","1111","8",,,,,,,,
"19825","1458317351.901919000","4","2","GSM MAP","190","invoke purgeMS ","1114","7","1112","6",,"42:02:11:11:11:11:93",,,,,,
"19826","1458317351.902438000","2","4","GSM MAP","182","returnResultLast ",,"6","1114","7",,,,,,,,
"19827","1458317352.902264000","14","2","GSM MAP","198","invoke updateLocation ","2224","7","1112","6",,"42:02:11:11:11:11:21",,,,,,
"19828","1458317352.902770000","2","4","GSM MAP","186","invoke cancelLocation ","1112","6","1114","7","42:02:11:11:11:11:21",,,,,,,
"19829","1458317352.903223000","4","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19830","1458317352.953136000","2","14","GSM MAP","194","invoke activateTraceMode ","1112","6","2224","7",,,,,"42:02:11:11:11:11:21",,,
"19831","1458317353.902559000","4","2","GSM MAP","182","invoke sendIMSI ","1114","7","1112","6",,,,,,"91:14:11:11:61",,
"19832","1458317353.903037000","2","4","GSM MAP","194","returnResultLast sendIMSI ",,"6","1114","7",,"42:02:11:11:11:11:51",,,,,,
"19833","1458317354.902975000","14","2","GSM MAP","190","invoke readyForSM ","2224","7","1112","6",,,,"42:02:11:11:11:11:22",,,,
"19834","1458317354.903237000","2","13","GSM MAP","190","invoke alertServiceCentre ","1112","6","2223","8",,,,,,,,"91:14:11:11:32"
"19835","1458317354.903778000","13","2","GSM MAP","182","returnResultLast ",,"8","1112","6",,,,,,,,
"19836","1458317355.903406000","5","2","GSM MAP","190","invoke sendRoutingInfoForGprs ","1115","149","1112","6",,"42:02:11:11:11:11:95",,,,,,
"19837","1458317356.903759000","14","2","GSM MAP","198","invoke updateLocation ","2224","7","1112","6",,"42:02:11:11:11:11:04",,,,,,
"19838","1458317357.904009000","4","2","GSM MAP","182","invoke sendIMSI ","1114","7","1112","6",,,,,,"91:14:11:11:45",,
"19839","1458317357.907501000","2","4","GSM MAP","194","returnResultLast sendIMSI ",,"6","1114","7",,"42:02:11:11:11:11:35",,,,,,
"19840","1458317357.968510000","2","14","GSM MAP","186","invoke insertSubscriberData ","1112","6","2224","7",,,,,,,,
"19841","1458317357.969157000","14","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19842","1458317357.969185000","2","14","GSM MAP","194","returnResultLast updateLocation ",,"6","2224","7",,,,,,,,
"19843","1458317357.969185000","2","14","GSM MAP","190","returnResultLast readyForSM ",,"6","2224","7",,,,,,,,
"19844","1458317358.105474000","2","4","GSM MAP","186","invoke cancelLocation ","1112","6","1114","7","42:02:11:11:11:11:04",,,,,,,
"19845","1458317358.106054000","4","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19846","1458317358.165473000","2","14","GSM MAP","194","invoke activateTraceMode ","1112","6","2224","7",,,,,"42:02:11:11:11:11:04",,,
"19847","1458317358.904389000","2","4","GSM MAP","186","invoke deleteSubscriberData ","1112","6","1114","7",,"42:02:11:11:11:11:23",,,,,,
"19848","1458317358.904848000","4","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19849","1458317359.904870000","1","4","GSM MAP","210","invoke eraseSS ","1111","8","1114","7",,,,,,,,
"19850","1458317359.905439000","4","1","GSM MAP","182","returnResultLast ",,"7","1111","8",,,,,,,,
"19851","1458317359.907458000","2","4","GSM MAP","186","invoke insertSubscriberData ","1112","6","1114","7",,,,,,,,
"19852","1458317359.907890000","4","2","GSM MAP","182","returnResultLast ",,"7","1112","6",,,,,,,,
"19853","1458317360.905205000","13","2","GSM MAP","194","invoke sendRoutingInfoForSM ","2223","8","1112","6",,,,,,,,"91:14:11:11:72"
"19854","1458317360.905802000","2","13","GSM MAP","210","returnResultLast sendRoutingInfoForSM ",,"6","2223","8",,,,"42:02:11:11:11:11:11:62",,,,
"19855","1458317360.955618000","13","3","GSM SMS","230","invoke mt-forwardSM ","2223","8","1113","8",,,,"42:02:11:11:11:11:62",,,,
"19856","1458317360.956145000","3","13","GSM MAP","182","returnResultLast ",,"8","2223","8",,,,,,,,
"19857","1458317361.005874000","3","2","GSM MAP","194","invoke sendRoutingInfoForSM ","1113","8","1112","6",,,,,,,,"91:14:11:11:72"
"19858","1458317361.006355000","2","3","GSM MAP","210","returnResultLast sendRoutingInfoForSM ",,"6","1113","8",,,,"42:02:11:11:11:11:11:62",,,,
"19859","1458317361.056303000","3","1","GSM SMS","230","invoke mt-forwardSM ","1113","8","1111","8",,,,"42:02:11:11:11:11:62",,,,
"19860","1458317361.056795000","1","3","GSM MAP","182","returnResultLast ",,"8","1113","8",,,,,,,,
"19861","1458317361.057244000","3","12","GSM MAP","202","invoke reportSM-DeliveryStatus ","1113","8","2222","6",,,,,,,,"91:14:11:11:72"
"19862","1458317361.057941000","12","3","GSM MAP","194","returnResultLast reportSM-DeliveryStatus ",,"6","1113","8",,,,,,,,
"19863","1458317362.058947000","1","3","GSM SMS","222","invoke mo-forwardSM ","1111","8","1113","8",,,,,,,,"91:14:11:11:63"
"19864","1458317362.059527000","3","2","GSM MAP","194","invoke sendRoutingInfoForSM ","1113","8","1112","6",,,,,,,,"91:14:11:11:74"
"19865","1458317362.060000000","2","3","GSM MAP","210","returnResultLast sendRoutingInfoForSM ",,"6","1113","8",,,,"42:02:11:11:11:11:64",,,,
```

Examples of raw SS7 network traffic captured with tshark.

```
2016-05-05T02:45:58+02:00,246,0.1307719087600708E2,15,9343
2016-05-05T03:23:21+02:00,246,0.8005490064620972E1,9,9321
2016-05-05T04:47:07+02:00,210,0.13876506805419922E2,9,8189
2016-05-05T04:46:46+02:00,246,0.27071521401405334E3,137,9343
2016-05-05T03:35:28+02:00,210,0.8907562971115112E1,9,8189
2016-05-05T03:00:03+02:00,246,0.7255486965179443E1,15,9321
2016-05-05T03:00:28+02:00,210,0.8156479835510254E1,12,8138
2016-05-05T05:25:15+02:00,210,0.8055685997009277E1,12,8138
2016-05-05T09:27:51+02:00,210,0.13129460096359253E2,8,8189
2016-05-05T10:19:14+02:00,210,0.8157486915588379E1,12,8138
2016-05-05T05:25:07+02:00,210,0.8004781007766724E1,8,8161
2016-05-05T05:24:59+02:00,210,0.13824933052062988E2,9,8189
2016-05-05T10:05:06+02:00,210,0.14550154099464417E4,12,8161
2016-05-05T06:02:37+02:00,210,0.8056103944778442E1,12,8138
2016-05-05T10:05:42+02:00,246,0.11018570184707642E2,3,9385
2016-05-05T10:05:23+02:00,246,0.8258591890335083E1,9,9321
2016-05-05T11:28:50+02:00,210,0.10015939950942993E2,12,8138
2016-05-05T10:42:23+02:00,246,0.8157886981964111E1,9,9321
2016-05-05T11:16:31+02:00,246,0.8155844926834106E1,9,9321
2016-05-05T11:28:40+02:00,210,0.8055498123168945E1,8,8161
2016-05-05T11:50:39+02:00,210,0.8006168842315674E1,8,8189
2016-05-05T11:50:47+02:00,246,0.8055556058883667E1,9,9321
2016-05-05T11:51:03+02:00,246,0.8056336879730225E1,3,9385
2016-05-05T12:24:53+02:00,210,0.8004915952682495E1,8,8189
2016-05-05T12:02:57+02:00,246,0.7204629898071289E1,15,9321
2016-05-05T15:23:40+02:00,210,0.12763249909877777E4,12,8161
2016-05-05T12:38:07+02:00,210,0.8004812002182007E1,12,8138
2016-05-05T13:15:58+02:00,210,0.8055212020874023E1,12,8138
2016-05-05T13:38:01+02:00,246,0.8055766820907593E1,9,9321
2016-05-05T14:14:22+02:00,246,0.8005650043487549E1,3,9385
2016-05-05T13:37:44+02:00,210,0.13059679479599E4,12,8161
2016-05-05T14:27:20+02:00,210,0.885551381111145E1,9,8189
2016-05-05T12:37:37+02:00,246,0.1724833607673645E2,15,9321
2016-05-05T13:15:34+02:00,246,0.7205180883407593E1,15,9321
```

Examples of preprocessed SS7 features used in machine learning for anomaly detection. Showing time, byte length, time since last location update, travel distance, and new LAC