FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Design, Implementation and Evaluation of a Secure VoIP System Based on Theoretical and Empirical Analysis of Threats and Attacks

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

eingereicht von

DI Markus Gruber, BSc
Matrikelnummer 0625544

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig

Diese Dissertation haben begutachtet:

(Univ.Prof. DI Dr.
Thomas Grechenig)

(Privatdoz. DI Dr.
Engin Kirda)

Wien, 22.04.2014

(DI Markus Gruber, BSc)

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Design, Implementation and Evaluation of a Secure VoIP System Based on Theoretical and Empirical Analysis of Threats and Attacks

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

DI Markus Gruber, BSc
Registration Number 0625544

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Thomas Grechenig

The dissertation has been reviewed by:

(Univ.Prof. DI Dr.
Thomas Grechenig)

(Privatdoz. DI Dr.
Engin Kirda)

Vienna, 22.04.2014

(DI Markus Gruber, BSc)

Erklärung zur Verfassung der Arbeit

DI Markus Gruber, BSc
Salzachstraße 13/16, 1200 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Abstract

Classical telephony has been radically changed by the ongoing networking of systems and services and the associated crossover into Internet services, in both the private and business domains. Voice over IP (VoIP) systems have become well established and have gained widespread acceptance. However, the very popularity of these VoIP systems means they now face new forms of attacks and types of attackers. Therefore securing VoIP systems is now of paramount importance to companies and organizations, for example to thwart industrial espionage or the compromising of their communications.

Establishing adequate VoIP security mechanisms is a continuous process which must be adapted to evolving threats. The threats to VoIP systems must be evaluated on the basis of known attacks as well as by collecting ongoing attacks on VoIP systems, in order to better understand the pattern of new attacks and the behavior of attackers.

A combination of theoretical and empirical analysis (by capturing real-world attacks using a honeynet) was used to gain information about the real-world threats to VoIP systems. This information was used to establish security measures for the most important attacks, which were then implemented in a transparent VoIP security layer. This layer offers a resource optimized protocol for the end-to-end encryption of client-server communication (SIP) and client-client communication (RTP). Therefore, it protects both the conversation content and the meta-data associated with the communication. By using strong authentication and encryption mechanisms (which take into account the disadvantages of previous approaches), the risk of identity theft and eavesdropping on a conversation are lowered to an acceptable level. A proof of concept of this security layer, implemented with a standard VoIP architecture, proved its applicability and usefulness for mobile telephony.

This applied security approach shows, that by integrating the VoIP security layer, the security of currently deployed VoIP systems can be raised to an appropriate level and they can be used without reservation for critical communications.

Kurzfassung

Die ständige Vernetzung von Systemen und Diensten sowohl in privaten als auch in geschäftlichen Bereichen und die laufende Überführung in Internetdienste veränderte auch die klassische Telefonie. Voice over IP (VoIP) Systeme wurden etabliert und verbreitet. Diese VoIP-Systeme müssen sich nun neuen Angriffsformen wie auch neuen Typen von Angreifern stellen. Das Absichern von VoIP-Systemen ist speziell für Firmen und Organisationen von essentieller Bedeutung, um zum Beispiel Wirtschaftsspionage oder eine Kompromittierung der Verfügbarkeit zu verhindern. Die Etablierung von adäquaten Sicherheitsmechanismen ist ein laufender Prozess, der an die aktuellen Bedrohungen gegen VoIP-Systeme angepasst werden muss. Die Bedrohungen durch Angriffe auf VoIP-Systeme müssen sowohl auf Basis von bekannten Angriffen als auch durch Erhebung neuer Angriffe gegen VoIP-Systeme ermittelt werden, um Angriffsmuster und Angriffsverhalten der Angreifer besser verstehen zu können.

Durch theoretische und empirische Analysen von VoIP-Angriffen mit Hilfe eines VoIP-Honeynets konnten essentielle Informationen gewonnen werden, um Sicherheitsmechanismen gegen die wichtigsten erkannten Sicherheitsprobleme zu etablieren. Diese Schutzmaßnahmen wurden in einem eigenen VoIP-Security-Layer ressourcenschonend umgesetzt, welcher eine Ende-zu-Ende Verschlüsselung für Client-zu-Server (SIP) und Client-zu-Client Kommunikationen (RTP) bietet, um sowohl Gesprächsinhalte als auch Metadaten zu schützen. Durch den Einsatz von starken Authentifizierungs- und Verschlüsselungsverfahren, welche auch die Nachteile von bisherigen Sicherheitsmechanismen für VoIP-Systeme berücksichtigen, wird das Risiko eines Identitätsdiebstahls sowie das Abhören von Gesprächen auf ein akzeptables Risiko gesenkt. Eine Proof-of-Concept Implementierung zeigt die technische Umsetzbarkeit sowie die für den Anwender transparente Nutzbarkeit des vorgestellten VoIP-Security-Layers für Mobiltelefone in einer üblichen VoIP-Infrastruktur.

Dieser angewandte Sicherheitsprozess zeigt, dass durch die Integration des VoIP-Security-Layers die Sicherheit von aktuell verwendeten VoIP-Systemen erhöht werden kann und diese für kritische Telefonate verwendet werden können.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AES Advanced Encryption Standard.

BASE Basic Analysis and Security Engine.

BSI Bundesamt für Sicherheit in der Informationstechnik.

CA Certification Authority.

CBC Cipher Block Chaining.

CCM Counter with CBC-MAC.

CSRC Contributing Source.

CTR Counter Mode.

DDoS Distributed Denial of Service.

DNS Domain Name System.

DoS Denial of Service.

DTLS Datagram Transport Layer Security.

ECC Elliptic Curve Cryptography.

FHMQV Fully Hashed Menezes-Qu-Vanstone.

FTP File Transfer Protocol.

GCM Galois/Counter Mode.

GSM Global System for Mobile Communications.

GUI Graphical User Interface.

HMAC Hash-based Message Authentication Code.

HMQV Hashed Menezes-Qu-Vanstone.

HTTP Hyper Text Transfer Protocol.

IAS Information Assurance & Security.

IAX InterAsterisk eXchange.

ICE Interactive Connectivity Establishment.

ICMP Internet Control Message Protocol.

IDS Intrusion Detection System.

IETF Internet Engineering Task Force.

IP Internet Protocol.

IPS Intrusion Prevention System.

ISDN Integrated Services Digital Network.

IT Information Technology.

ITU International Telecommunication Union.

ITU-T ITU Telecommunication Standardization Sector.

JSON JavaScript Object Notation.

MAC Message Authentication Code.

MGCP Media Gateway Control Protocol.

MIKEY Multimedia Internet KEYing.

MitM Man in the Middle.

MOS Mean Opinion Score.

MQV Menezes-Qu-Vanstone.

MTU Maximum Transmission Unit.

NAT Network Address Translation.

NIDS Network Intrusion Detection System.

NIPS Network Intrusion Prevention System.

NSA National Security Agency.

OCB Offset Codebook Mode.

OCB1 Offset Codebook Mode Version 1.

PBX Private Branch Exchange.

PC Personal Computer.

PFS Perfect Forward Secrecy.

PKI Public Key Infrastructure.

PSTN Public Switched Telephone Network.

QoS Quality of Service.

RFC Request For Comments.

RMIAS Reference Model of Information Assurance & Security.

RSA Rivest-Shamir-Adleman.

RTCP Real-Time Control Procotol.

RTP Real-Time Transport Protocol.

S/MIME Secure/Multipurpose Internet Mail Extensions.

SAS Short Authentication String.

SCTP Stream Control Transmission Protocol.

SDES Session Description Protocol Security Descriptions.

SDP Session Description Protocol.

SIP Session Initiation Protocol.

SIPS Secure Session Initiation Protocol.

SMTP Simple Mail Transfer Protocol.

SPIT Spam over IP Telephony.

SQL Structured Query Language.

SRTP Secure Real-Time Transport Protocol.

SSH Secure Shell.

SSL Secure Socket Layer.

STUN Simple traversal of UDP over NAT.

TCP Transmission Control Protocol.

TLS Transport Layer Security.

TURN Traversal Using Relay NAT.

UA User Agent.

UAC User Agent Client.

UAS User Agent Server.

UDP User Datagram Protocol.

UI User Interface.

URI Uniform Resource Identifier.

VoIP Voice over IP.

VoIPSA VoIP Security Alliance.

VPN Virtual Private Network.

VSL VoIP Security Layer.

WebRTC Web Real-Time Communication.

WWW World Wide Web.

ZRTP Phil Zimmermann's Real-Time Transport Protocol.

CHAPTER 1

Introduction

Contents

1.1 Motivation

Since the beginning of the Internet, communications (such as chats, email or phone calls) have shifted from traditional communication infrastructures (such as postal services or Public Switched Telephone Network (PSTN) systems) to the Internet. In recent years, modern telephony in particular has made extensive use of the Internet instead of traditional infrastructures and has gained widespread acceptance in the form of Voice over IP (VoIP). [44, 148]

However, the increasing use of VoIP systems has also attracted the attention of attackers, as well as criminal organizations and government intelligence agencies [149]. Additionally, communication systems are generally of great interest to different attackers as a means of tracing connections between parties and obtaining the content of their communication. The recent revelations regarding the Internet surveillance program of various intelligence services [146, 32] have changed the requirements for secure communication systems, as the default Internet security mechanisms cannot be trusted any more. The main question is, how can we communicate in a private and secure way on the Internet?

Initially the main applications of the Internet were file transfers and e-mail. With the introduction of the World Wide Web (WWW), the Internet changed into a global and open information distribution channel. And more recently the Internet has become a real-time communication channel (for services such as VoIP) that integrates all the earlier multimedia capabilities. [148]

Conventional VoIP systems consist of various components (e.g., clients, servers, gateways, etc.) and are very popular nowadays. In particular, VoIP on mobile devices is gaining in popularity, but the security aspects are often neglected or the measures chosen are not suitable. To increase the level of security in VoIP systems, each component must be secured, e.g., the VoIP proxies, the VoIP terminals, smartphones and the connection between each of the devices involved. Comprehensive security tests for each component are needed to protect the whole VoIP infrastructure, because one poorly secured component can be sufficient to cause substantial damage (weakest-link problem [132]).

The interconnection between VoIP and PSTN systems is an important functionality in telephony, which enables calls between the VoIP network and the PSTN system. However, for calls within VoIP networks no additional call charges usually arise, whereas in most cases calls to PSTN systems are not for free. When an attacker compromises a VoIP system with the goal of establishing free calls to a PSTN system, the costs are incurred by the operator of the VoIP system.

In general, attacks on VoIP systems are becoming more imaginative and many attacks can cause damage, e.g., gain money for attackers, create costs for the victim or violate the privacy of the communicating parties. Therefore, only accepted and authenticated participants should be able to process the communication data and no third-party should be able to intercept and interpret the data.

Real-world VoIP attacks have to be captured and analyzed to identify the main threats to VoIP systems. The analysis of the attacks gives valuable details about the pattern of the attacks as well as the attackers' behavior, which can then be used to create a risk assessment matrix. Based on the major risks identified, countermeasures were implemented to improve the security level of VoIP systems and to decrease the risks to an acceptable level.

1.2 Research Challenges and Questions

This work adopts a conventional security process to improve the security level of VoIP systems. This process consists of capturing/detecting real-world attacks against VoIP systems, analyzing and evaluating the identified threats and risks, implementing countermeasures against the major vulnerabilities and finally examining the effectiveness of the protection mechanisms. To achieve this goal the following questions will be discussed in this thesis:

Efficient attack collection:

- How can real-world attacks be collected independently of the network protocol or the services offered?
- Is a generic approach for collecting attacks possible and how can this be done?
- How is the separation of attack data and real communication data, i.e., collecting VoIP communications from attackers and not from regular users, possible?

Analysis of VoIP attacks:

- How can the collected attacks be automatically analyzed to retrieve meaningful information about the attacker and the attacks?
- Is it possible to use additional information sources to extend the attack data and to get even more information about the attacker or their behavior?
- Which real-world attacks are currently carried out against VoIP infrastructures?
- Is it possible to uncover the attackers' business models from the individual attacks?
- What are the security problems and the risks to current VoIP systems?

Improving the Security of VoIP systems:

- How can a VoIP system be secured, using the findings of the attack analysis and evaluation?
- Is it possible to use existing VoIP protocols in a secure and non-traceable way?
- How can a secure and private VoIP communication be established in a closed network?
- Can the protection mechanisms be tuned to work with the limited resources of mobile devices?

1.3 Contributions

The contribution of the answers to the questions from part one is to design and implement a generic method to identify and detect real-world attacks against VoIP systems. The second part of the work is the analysis and evaluation of the collected attacks to gain more information about the attacks, and to understand the behavior of the attackers and the different kinds of attacks, e.g., Denial of Service (DoS) or fraudulent calls, detection of attack patterns, etc.

The empirical analysis is important to investigate state-of-the-art VoIP attacks and to identify the current security problems of VoIP systems. Based on these findings, countermeasures are implemented in part three to better protect current VoIP solutions (i.e., create new protection mechanisms or adapt current ones) and to ensure private and secure VoIP communication.

The major contributions and novel findings of this work are as follows:

A honeynet approach for detecting and capturing VoIP attacks: A generic honeynet approach was defined and implemented for VoIP systems, including a VoIP attack analyzing engine for investigating VoIP attacks. This approach offers a VoIP system with attractive services for attackers, in order to get meaningful information about attacks and the attacker behavior. The honeynet only captures attack data, because the honeynet itself has no legitimate activity and each connection to it can be classified as an attack.

Analyzing and evaluating real-world VoIP Attacks: The automated and flexible VoIP attack analyzing engine makes it possible to expand our knowledge about VoIP attacks as well as identifying new VoIP attacks very quickly. Different information sources can be used to extend the attack data with meaningful information (such as country information or calling code information). The results of the analysis by the VoIP attack analyzing engine helped to identify the weakest links of VoIP systems and can be used to improve the current protection mechanisms, to be warned about further attacks or to initiate countermeasures.

Introducing a transparent security layer to secure VoIP communications: Based on the identified and evaluated threats and risks to VoIP systems, countermeasures were developed and implemented with the intention of ensuring private and secure VoIP communications. The security layer introduced is compatible with existing VoIP protocols (e.g., Session Initiation Protocol (SIP) or Real-Time Transport Protocol (RTP)) and only the trusted parties of the communication have access to the communication data and the metadata of a call.

1.4 Organization of This Thesis

This thesis builds on academic papers that were published at conferences during the research period. The individual papers, which focused on specific aspects of the research, are expanded and integrated into a cohesive presentation of how the security of VoIP systems can be improved by using the analysis of real-world attacks. The following description of the thesis structure references the relevant research papers where appropriate.

The first chapter of this thesis covers the motivation, research challenges and questions, and the contribution of the work. The main part of the thesis is divided into following chapters:

- **Chapter 2 - Fundamentals of IT Security for VoIP Systems:** This chapter gives an introduction to Information Technology (IT) security and describes terms and definitions. An introduction to the basics of VoIP systems and their security concerns is also given in this chapter. Identification of real-world attacks using a honeynet and the fundamentals of cryptography in communication systems are described as well.
- **Chapter 3 - State-of-the-Art of VoIP Security:** This chapter presents a short introduction to common security workflows. Current VoIP systems as well as their known security problems and countermeasures are described in this chapter. Also considered are current honeynet solutions, as well as the state-of-the-art of cryptography in communication systems.

Part I - VoIP Attack Collection – Automated Capturing Using a Honeynet Approach

- **Chapter 4 - Introduction to VoIP Protocols for Capturing Attacks:** The most common VoIP protocols are presented in this chapter. To capture and analyze the attack data of VoIP systems it is essential to know details about the protocols, because each modification of the protocol or the parameters may have an impact on the security level.
- **Chapter 5 - Design and Implementation of a VoIP Honeynet for Capturing and Analyzing VoIP Attacks [1, 3]:** The concept of a VoIP honeynet and a highly flexible VoIP attack analyzing engine for identifying real-world threats is presented in this chapter. This approach is used to capture the data of real-world VoIP attacks by providing enticing honeypots for the attackers.

Part II - Attack Analysis – Investigating VoIP Attacks

- **Chapter 6 - Analysis and Evaluation of VoIP Specific Attacks [2]:** This chapter presents the results of the analysis and evaluation of the captured data from two VoIP specific honeynets. The various honeypots are only accessible via VoIP and do not have an uplink to PSTN systems. The VoIP specific honeynets were deployed to get valuable details about VoIP specific attacks. This analysis helps to identify and to evaluate the main security problems of current VoIP specific systems.
- **Chapter 7 - Analysis and Evaluation of PSTN Specific VoIP Attacks [4]:** The analysis and evaluation of PSTN specific VoIP attacks are presented in this chapter. This third honeynet had a PSTN uplink and could call numbers in a PSTN system. Through collection, analysis and evaluation of fraudulent calls, the PSTN specific VoIP honeynet provides details about the attack model for fraudulent calls and the business model behind them.

Part III - Hardening VoIP Systems – Countering Real-World Attacks

- **Chapter 8 - Design Criteria for a Secure VoIP Solution [6]:** Based on the results of the theoretical analysis and the investigation of the real-world attacks, this chapter identifies and evaluates the current risks to VoIP systems and presents design criteria for secure and non-traceable VoIP communications.
- **Chapter 9 - Design and Implementation of a Transparent Security Layer to Enable Anonymous VoIP Calls [6]:** This chapter presents the design and implementation of an additional transparent security layer to conventional VoIP systems, which ensures private, anonymous and secure VoIP communications in closed environments. The VoIP security layer was implemented for mobile devices as a proof of concept.
- **Chapter 10 - Evaluation and Discussion of the Security and Voice Quality Aspects of the Proposed Solution:** The implementation of the proposed security measures is evaluated in this chapter and shows that the proposed measures reduce the risk level of VoIP systems to an acceptable level. Also discussed in this chapter is the additional security overhead for VoIP communications and the impact on Quality of Service (QoS).

Chapter 11 concludes the thesis, summarizes the findings of the author in the context of improving security for VoIP systems using a honeynet approach, and discusses possible further research.

1.5 List of Publications

The work in this thesis was presented and published at academic and peer-reviewed conferences.

1. M. Gruber, F. Fankhauser, S. Taber, C. Schanes, and T. Grechenig. Trapping and analyzing malicious VoIP traffic using a honeynet approach. In *The 6th International Conference on Internet Technology and Secured Transactions (ICITST)*, pages 442–447, 2011. [57]

2. M. Gruber, F. Fankhauser, S. Taber, C. Schanes, and T. Grechenig. Security status of VoIP based on the observation of real-world attacks on a honeynet. In *The Third IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*, pages 1041–1047, 2011. [56]

3. M. Gruber, C. Schanes, F. Fankhauser, M. Moutran, and T. Grechenig. Architecture for trapping toll fraud attacks using a VoIP honeynet approach. In J. Lopez, X. Huang, and R. Sandhu, editors, *Network and System Security*, volume 7873 of *Lecture Notes in Computer Science*, pages 628–634. Springer Berlin Heidelberg, 2013. [60]

4. M. Gruber, C. Schanes, F. Fankhauser, and T. Grechenig. Voice calls for free: How the black market establishes free phone calls – trapped and uncovered by a VoIP honeynet. In *Eleventh Annual International Conference on Privacy, Security and Trust (PST)*, pages 205–212, 2013. [59]

5. M. Gruber, P. Wieser, S. Nachtnebel, C. Schanes, and T. Grechenig. Extraction of ABNF rules from RFCs to enable automated test data generation. In L. Janczewski, H. Wolfe, and S. Shenoi, editors, *Security and Privacy Protection in Information Processing Systems*, volume 405 of *IFIP Advances in Information and Communication Technology*, pages 111–124. Springer Berlin Heidelberg, 2013. [61]

6. M. Gruber, M. Maier, M. Schafferer, C. Schanes, and T. Grechenig. Concept and Design of a Transparent Security Layer to Enable Anonymous VoIP Calls. In *Proceedings of the International Conference on Advanced Networking, Distributed Systems and Applications (INDS)*, 2014. [58]

7. B. Isemann, M. Gruber, C. Schanes, M. Grünberger, and T. Grechenig. Chaotic Ad-hoc Data Network – A Bike Based System for City Networks. In *The 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, 2014. [72]

8. M. Schafferer, M. Gruber, C. Schanes, and T. Grechenig. Data Retention Services with Soft Privacy Impacts: Concept and Implementation. In *Proceedings of the International Conference on Software Engineering and Service Science (ICSESS)*, 2014. [128]

CHAPTER 2

Fundamentals of IT Security for VoIP Systems

Contents

2.1 Introduction to IT Security

Nowadays, a lot of information is created, stored, transformed or reprocessed by IT systems. The goal of IT security is to ensure the protection of the digitally stored information and its processing, as Bedner and Ackermann [11] presented. Schneier [133] and Bishop [16] describe computer security based on the three aspects *confidentiality*, *integrity* and *availability* (the CIA triad). Confidentiality means that only authorized users can read sensitive information or resources. A system ensures confidentiality when no unauthorized user can access sensitive information or resources, e.g., by employing cryptographic mechanisms. Data integrity means that no one has modified the data without permission. Availability means, that an authorized subject has access to the desired information or resource, e.g., to a system.

Bishop [16] describes a *threat* as a potential violation of any of these three security aspects. The actions which cause a violation are called *attacks*. Those who execute the attacks are called *attackers*. Eckert [41] describes an attack as illegal and unauthorized activity, for example to damage resources, files or programs. As Bishop [16] describes, the specific failure of the controls, so that someone can break into a computer system, is called a *vulnerability*. Newman [104] defines an attack as:

> *An attack on a computer system or network involves the exploitation of the vulnerabilities, which can result in a threat against the resource.* (Newman [104])

In 2013, an extension of the CIA triad (called Information Assurance & Security (IAS) Octave) was presented by Cherdantseva and Hilton [24]. The IAS Octave is a Reference Model of Information Assurance & Security (RMIAS), which aims to address the evolving trends in the IAS domain and covers the security principles *confidentiality, integrity, availability, privacy, authenticity & trustworthiness, non-repudiation, accountability and auditability* [24].

The security principles of the IAS Octave are described as [24]:

- **Accountability**: Each action by a user should be traceable, to make users responsible for their actions.
- **Auditability**: All actions performed by humans or machines within the system should be persistently monitored in a non-by-passable way.
- **Confidentiality**: Only authorized users should be able to access information.
- **Integrity**: Unauthorized modifications (including completeness and accuracy) should not be possible in any component of a system.
- **Availability**: Authorized users should have access to all system components when they are required.
- **Authenticity/Trustworthiness**: Verifying the identity and establishing trust in a third party and in the information it provides should be possible.
- **Non-repudiation**: The occurrence/non-occurrence of an event or participation/non-participation of a party in an event should be provable.
- **Privacy**: Privacy legislation should be followed and individuals should be able to control their personal information.

Bishop [16] describes the goal of security as mechanisms to prevent an attack, detect an attack or recover from an attack. These mechanisms may be used together or separately. The attackers mostly use the weakest security mechanism (weakest-link problem [132]) to attack a system.

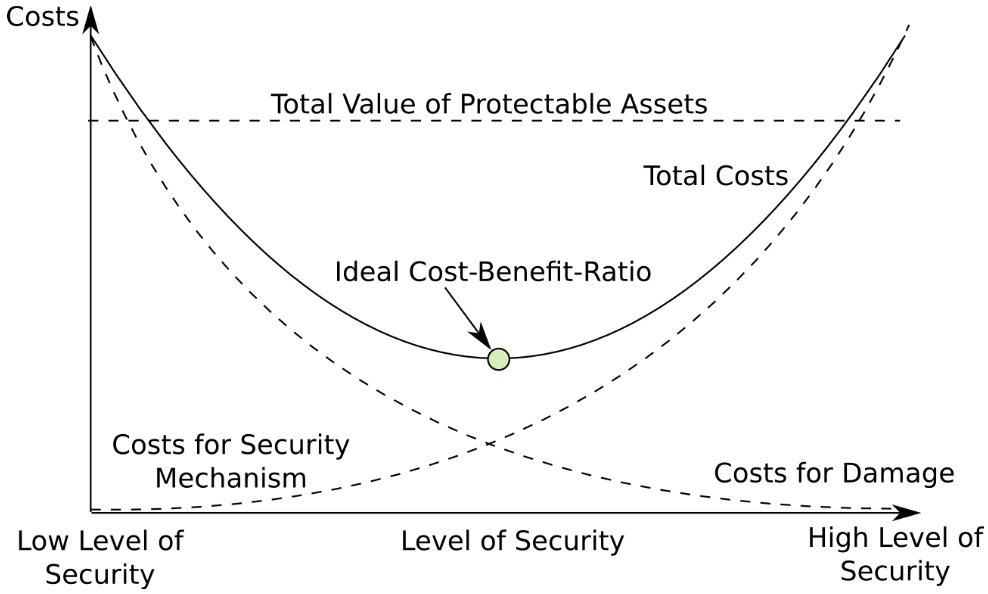


Figure 2.1: Ideal Cost-Benefit-Ratio of security mechanism based on Raepple [113].

Figure 2.1 shows the connection between the value of the protected information (assets) and the costs of security mechanisms required to protect that information. Only as many security mechanisms as are necessary should be introduced, so that the costs for attacking a system are higher than the value of the protected assets. To find the ideal Cost-Benefit-Ratio is essential, because if this is not the case, either the costs for the security mechanisms are too high or the costs for the damage are too high. It also shows that a maximum level of security requires almost infinite costs and is therefore hard to ensure in practice. In security engineering, a security process based on risk assessment (e.g., ISO/IEC 27001 [82]) is often used to find and evaluate the risks to the assets, and to select the right security mechanisms to ensure an acceptable value of the Cost-Benefit-Ratio.

Every device and piece of software is a potential security risk and the security mechanisms used should be evaluated and adapted periodically to avoid evolving attack models. Schneier [133] described this as: “Security is a process, not a product”. But the security of systems cannot be covered by technology alone, organizational activities are also needed, e.g., establishing awareness of security among employees. [133]

> *If you think technology can solve your security problems, then you don’t understand the problems and you don’t understand the technology.* (Schneier [133])

An attack can have different impacts and different intentions. Newman categorizes an attack into four general categories [104]:

- **Interception**: The data of a transmission are obtained for some unauthorized use.
- **Interruption**: The transmission of data is inhibited by an interruption in a communication channel.
- **Modification**: The data contained in the transmissions have been modified.
- **Fabrication**: The deceit of an unsuspecting user by fabricating a deception.

To operate a system (such as a VoIP system) each of the components, whether protected or unprotected, should be monitored to detect successful and unsuccessful attacks against the system. Computer systems that are not under attack exhibit the following characteristics, as Bishop [16] described:

1. The actions of processes and users of a computer system generally match statistically predictable patterns.
2. The actions of processes and users of a computer system do not include commands (or sequences of commands) which subvert the security policy of a system.
3. The actions of processes of a computer system match a set of specific allowed actions.

Denning [31] hypothesized that a system under attack failed at least one of the characteristics above.

The presented IT security concepts are used for the analysis of threats and risks to VoIP systems and also for the design, implementation and evaluation of the protection mechanisms.

2.2 Basics of VoIP Systems

VoIP systems enable advanced communication (such as voice or video) over the Internet or other data networks and are replacing more and more traditional phone infrastructures. Nowadays, VoIP is widely used in organizations, companies and private environments, as it has the advantage of flexibility and low costs. Many existing devices and applications use standardized VoIP protocols (e.g., SIP or RTP). Current market analysis of VoIP predicts that the overall VoIP service market will grow from $63 billion in 2012 to $82.7 billion in 2017 [71].

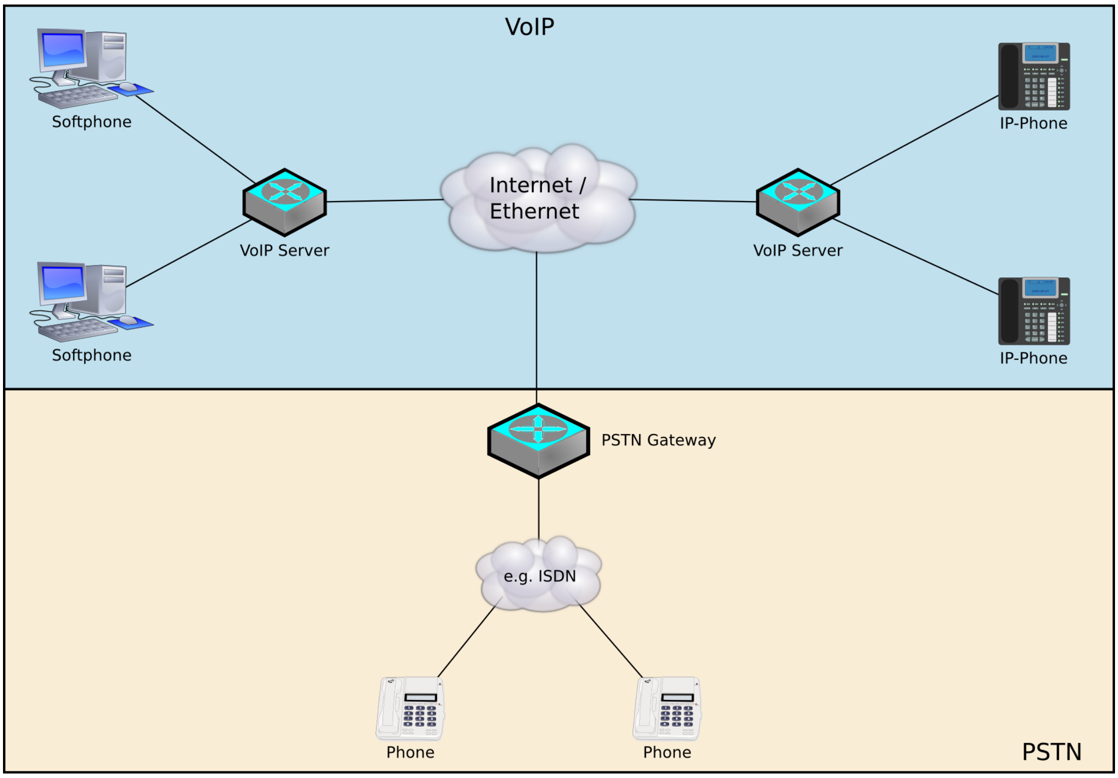


Figure 2.2: Conventional VoIP architecture based on Eren and Detken [44].

VoIP makes it possible to communicate via Internet Protocol (IP) based networks, instead of using the traditional PSTN infrastructure. PSTN is an interconnected circuit-switched network that is built, owned and operated by private or government organizations. To connect a conventional phone service to a VoIP service a special PSTN gateway is necessary. [148]

In VoIP systems it is possible to communicate via different devices, e.g., softphones, mobile devices or traditional devices, which are connected to the system. Figure 2.2 presents an overview of a conventional VoIP architecture based on Eren and Detken [44]. Various VoIP phones (i.e., softphones and IP-Phones) are connected to different VoIP servers. This VoIP architecture has also a connection to the PSTN network via a PSTN gateway.

In general, a VoIP call is divided into a *signaling phase* and a *media transmission phase*. Usually, signaling occurs at the beginning when a call is initiated and again at the end when the call has ended. For certain purposes signaling messages can also occur during the communication. [148, 44]

Network Path of VoIP Communications

In a typical VoIP call different protocols are involved, e.g., SIP for signaling, RTP for media transmission and Real-Time Control Procotol (RTCP) for control of the media transmission. Figure 2.3 presents a possible network connection of a VoIP call between a VoIP server and two User Agents (UAs) based on SIP and RTP. The VoIP server is not always necessary, because a direct call setup between both UAs is also possible if they know each other's location.

In contrast to many other services in the Internet (such as email), VoIP clients not only establish a connection to the VoIP server, but in many cases the clients also establish a connection to each other for the media transmission. To secure a VoIP system all possible communication paths must be secured, it is not sufficient to secure only one path. For example, in the signaling phase lots of metadata, i.e., descriptive information about the participating parties, can be derived even if the media transmission is encrypted. This information can provide valuable details about call duration or who is talking to whom.

Additional configuration is needed to make VoIP calls in Network Address Translation (NAT) networks, because the media transmission is usually carried out directly and both parties must be reachable from the Internet. As Badach [7] described, it is possible to use VoIP in NAT networks, by using *symmetric response*, *symmetric RTP/RTCP*, Simple traversal of UDP over NAT (STUN), Traversal Using Relay NAT (TURN) or the more common Interactive Connectivity Establishment (ICE) solution. The various possible VoIP network architectures make it more difficult to secure all the communication channels, e.g., STUN needs an additional request to SIP and RTP.

Entities of a SIP Architecture

A possible SIP architecture consists of the following main entities, as described in Request For Comments (RFC) 3261 [124]:

- **User Agent**: A logical entity that can act as both a User Agent Client (UAC) (which creates new requests) and User Agent Server (UAS) (which generates responses) for the duration of a session.
- **Proxy Server**: An intermediary entity whose primary role is to route a message to another entity. Proxies can also modify parts of the request before forwarding it.

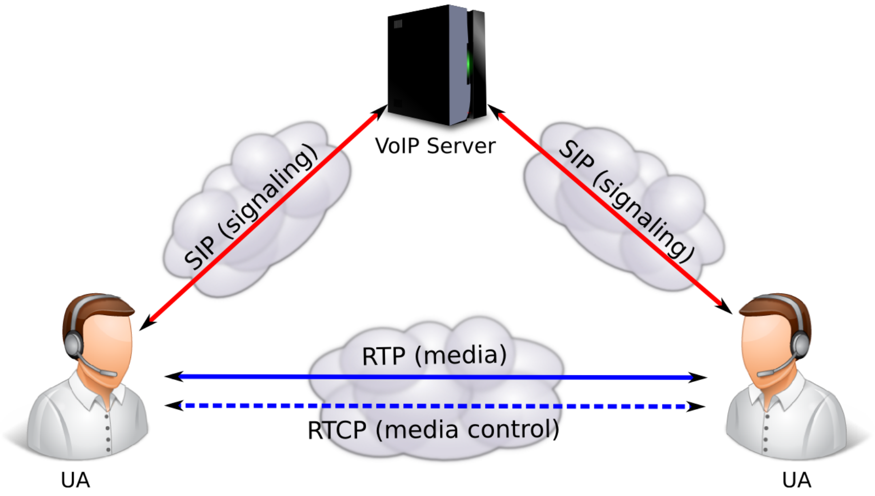


Figure 2.3: Conventional VoIP communication channels between a VoIP server and two user agents based on SIP and RTP.

- **Redirect Server**: A redirect server provides a list of the alternative locations which a user should contact to reach another entity.
- **Registrar Server**: The location of a user (based on the SIP or Secure Session Initiation Protocol (SIPS) Uniform Resource Identifier (URI) of the contact field from the REGISTER message) is bound to the username and stored on the location server.
- **Location Server**: The location server provides information about a callee's possible locations (which are bound to a username).

Figure 2.4 shows a high-level view of the interactions of SIP entities based on Keromytis [81]. User Alice registers on the Registrar Server in SIP Domain A ①, which stores the registration in the Location Server ②. When user Bob calls user Alice, he first contacts the Proxy Server in Domain B ③, which looks up the Location Server ④. After the Proxy Server knows the location of the other Proxy Server, the Proxy Server from Domain B forwards the call to the Proxy Server from Domain A ⑤. The Proxy Server uses the Location Server from Domain A ⑥ to get the location information for User Bob in order to forward the call to the final endpoint ⑦. Afterwards both endpoints transmit the media packets directly ⑧ to each other, but a communication via a Proxy Server is also possible. [81]

In this work a VoIP system refers to all the elements described above, i.e., the end-to-end solution including the user agents at the periphery. A VoIP server refers to the centralized elements, i.e., proxy server, redirect server, registrar server and location server.

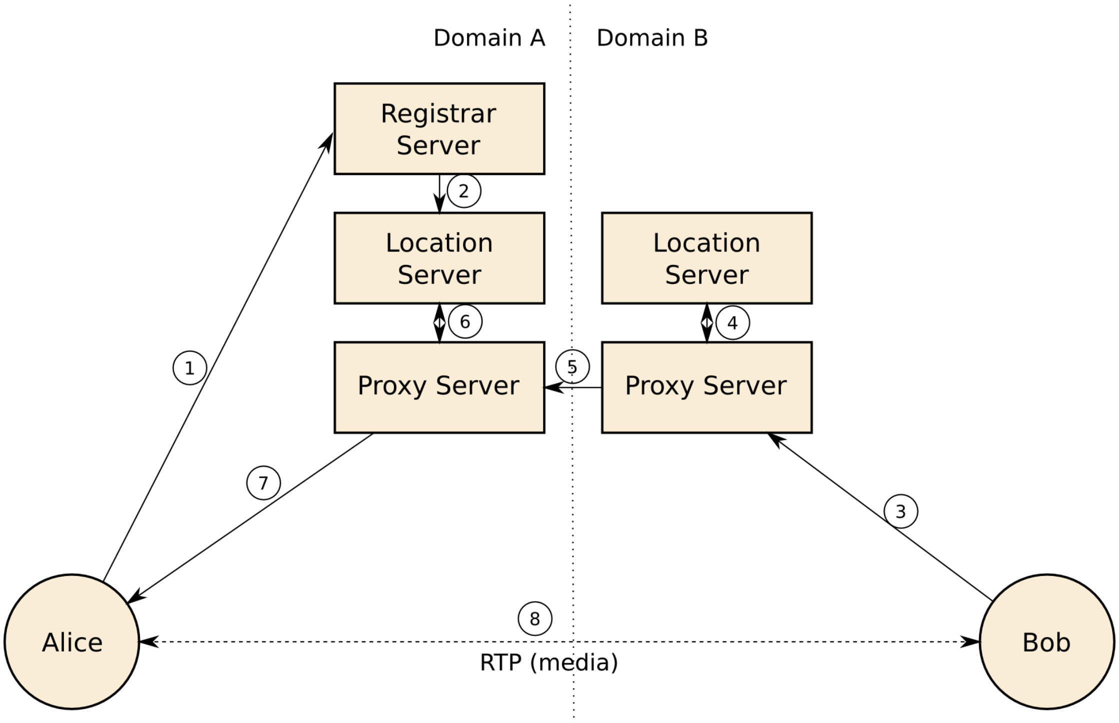


Figure 2.4: High-level view of interaction of SIP entities based on Keromytis [81].

VoIP Call Setup

Figure 2.5 (based on Keromytis [81]) shows an example of a common VoIP call setup between two UAs and a VoIP server with the signaling protocol SIP and the media transmission protocol RTP. SIP uses different methods (e.g., REGISTER, INVITE or OPTIONS) for invoking a particular operation. Alice sends a SIP INVITE message to the VoIP server, to initiate a new call session with User Bob. The VoIP server forwards this message directly to Bob, if Alice and Bob are registered in the same domain. If Bob is registered in a different domain, the message will first be forwarded to the VoIP server in Bob's domain and then directly to Bob. While Bob processes the INVITE message, he sends a TRYING and a RINGING response message to the VoIP server, which forwards the packets to Alice. Once Bob has accepted the call, an OK message is sent to Alice via the VoIP server. Alice responds with an ACK message, in order to start the media transmission. The media transmission takes place directly between Alice and Bob. To finish the communication Bob terminates the session with a BYE message and Alice confirms the termination with an OK message. [81]

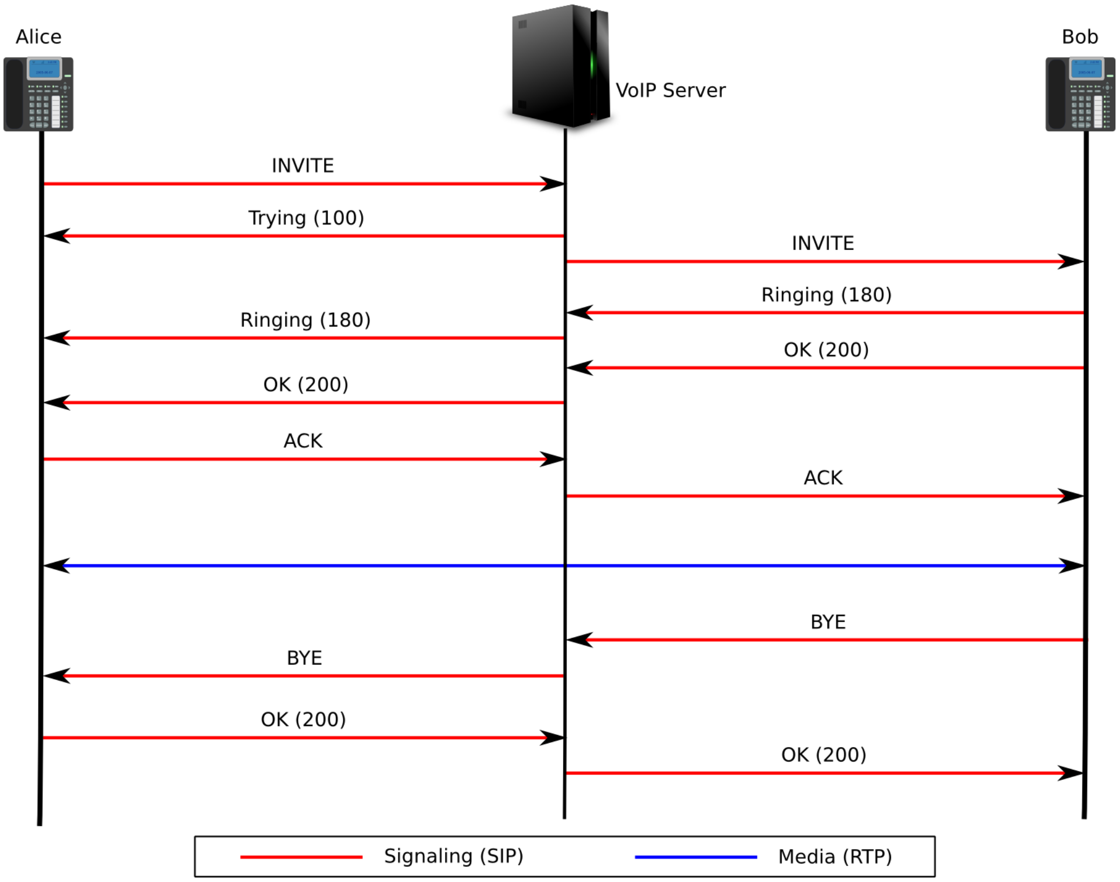


Figure 2.5: Common VoIP call setup with SIP and RTP based on Keromytis [81].

VoIP Security Problems and Taxonomy

VoIP uses the IP protocol stack to transmit data from the source to the destination, which implicitly brings additional security threats with it. [44]

Based on the findings of Werapun *et al.* [156] we categorize security threats and vulnerabilities of VoIP systems into horizontal and vertical attack surfaces. The vertical attack surface (as seen in Figure 2.6) focuses on all attacks from the transmission medium to the SIP and RTP protocols (e.g., DoS, replay attacks, spoofing, sniffing and Man in the Middle (MitM) attacks). The horizontal attack surface (as shown in Figure 2.7) covers all conceptual, implementation or operational vulnerabilities of VoIP servers themselves (e.g., Structured Query Language (SQL) injection or buffer overflow).

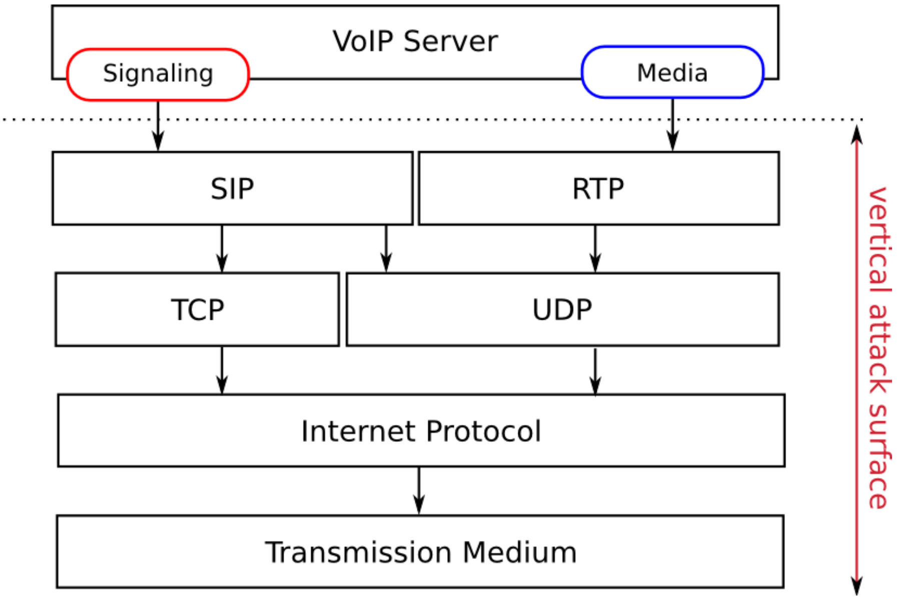


Figure 2.6: Vertical attack surface of VoIP systems based on Eren and Detken [44].

To classify the actually known threats to VoIP systems we use the taxonomy provided by the VoIP Security Alliance (VoIPSA). VoIPSA[1] is an open, vendor-neutral organization and is composed of VoIP and information security vendors, organizations and individuals with the aim of securing VoIP systems. The key elements of the VoIPSA security threat taxonomy [153] (interpreted from Keromytis [81]) are:

1. **Social threats** are aimed directly against humans, e.g., phishing, theft of service, or unwanted contact. [81]

2. **Eavesdropping, interception, and modification threats** are unauthorized or unlawful interception or modification of signaling or media data, e.g., call re-routing and interception of unencrypted RTP sessions. [81]

3. **Denial of service threats** deny access to VoIP services for the users, e.g., VoIP-specific attacks, VoIP-agnostic attacks, or attacks against physical components. [81]

   Malformed Messages are very popular: Al-Allouni *et al.* [2] divide malformed message attacks into structure malformed messages and syntax malformed messages. The structure malformed messages conform to the RFC 3261 [124] syntax, but the whole structure of the message is overly complex, resulting in time consuming processing for the parser to

[1] http://www.voipsa.org/

execute the message. In a bad implementation of VoIP systems also errors can occur, e.g., buffer overflows. Syntax malformed messages do not conform to the RFC 3261 syntax. They violate the SIP protocol rule in such a way that SIP parsers are unable to successfully categorize the received messages. [2]

In this work all malformed messages, independent of syntax or structural malformation, are handled as malformed messages.

4. **Service abuse threats** cover the improper use of one or more VoIP services. Especially services in a commercial setting are popular targets (e.g., for toll fraud and billing avoidance). [81]

   Very popular attacks are:

   - **Identity Theft**: attackers try to get a valid identity from a user of the VoIP system [153]. One variant of identity theft is a dictionary and brute-force attack. The attackers try to identify a username and/or a password of a valid SIP account and use this to get access to someone else's VoIP account. For a dictionary attack the attackers send many words from a list of names and passwords to the VoIP server and identify possible accounts based on the response messages. For a brute-force attack, attackers send various combinations of characters or words to the VoIP server and also identify possible accounts based on the response message. Other variants are for example social engineering or vulnerabilities in the implementation of the VoIP server.
   - **Fraudulent Calls**: attackers call a victim with fraudulent intentions. The term "toll fraud" is used if a person or a group of people use paid services using another person's account, as described by Hoffstadt *et al.* [66]. In terms of SIP messages, the attacker first sends a REGISTER message containing the stolen credentials to the VoIP server. After the 200 OK response message from the server, the attacker can initiate calls by using INVITE messages. The costs of the calls have to be paid by the account owner.

     In the case of a fraudulent VoIP call, an attacker calls a victim with fraudulent intentions. In some cases attackers try to cause costs for the potential victim or distribute advertising news, e.g., the attackers use the hacked infrastructure to hide their own identity from the potential victims and immediately hang up and await a chargeable call-back from the victim.

     The fraudulent call attack is a two-stage process. After the identification of a suitable VoIP system, an attacker starts the first phase and tries to gain access to the system,

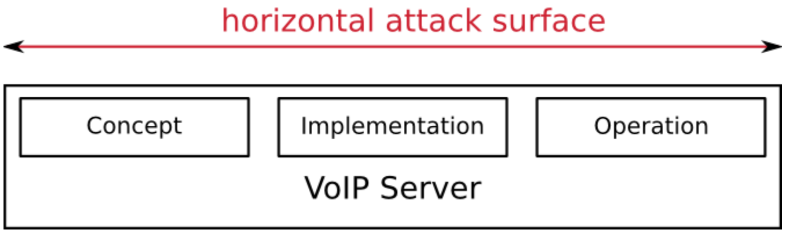


Figure 2.7: Horizontal attack surface of a VoIP server.

e.g., by brute-force attacks or social engineering. In the second phase the attacker connects to the system under attack and attempts to make calls.

5. **Physical access threats** are illegal or unauthorized physical access to VoIP devices or network components. [81]

6. **Interruption of service threats** are non-intentional problems that may nonetheless cause VoIP services to become unusable or inaccessible for users, e.g., loss of power due to inclement weather or resource exhaustion due to over-subscription. [81]

Stakeholders of VoIP Attacks

To better understand VoIP attacks and how to protect VoIP systems against these attacks we define the most important stakeholders of VoIP attacks. Further, we use the following definitions of the stakeholders:

- **Caller**: The caller initiates the call to the callee. The caller can be an attacker, e.g., when they attack a VoIP system with the intention of calling another person or earning money with premium numbers. However, the caller does not necessarily have to be an attacker, e.g., when they think they use a proper (third party) VoIP/phone service without malicious intentions, but the third party service uses a compromised infrastructure.

- **Callee**: The callee is called by the caller. The callee may use VoIP phones in an enclosed VoIP system, or a mobile phone or a PSTN number. In the case that the callee and the caller are the same person (e.g. someone using a softphone to call their own mobile phone), then the callee would also be involved in the attack.

- **Probing instance**: The probing instance tries to identify a vulnerable VoIP system for further attacks. The probing instance can also be a caller, e.g., to identify PSTN routes.

- **Attacker**: The attacker does not have to be the caller or the callee. He might also be the controlling instance of either, the caller, the callee or the probing instance. The attacker is a person with malicious intentions.
- **Owner of the VoIP system**: The owner of the VoIP system is the victim of the attack and often has to pay for the effects (e.g., pays the costs for the calls). They own a VoIP system containing security vulnerabilities, e.g., a misconfigured system or weak/default passwords.
- **Third party eavesdropper**: A third party can eavesdrop, analyze and interpret all traffic in a VoIP network, e.g., state-based surveillance.

The caller, the callee or the probing instance may use one or more hosts for their purposes. In this work a host is a physical or virtual environment with a unique IP address. In most cases the caller and the callee use two different hosts. The host of the caller is also known as the UAC and the host of the callee as the UAS.

2.3 Identification of Attacks using a Honeynet

For the selection and implementation of protection mechanisms it is important to know the details of existing real-world attacks. Various methods to identify attacks exist (e.g., traffic analysis, Intrusion Detection System (IDS) systems or firewalls). One method to identify real-world attacks is a honeynet. Honeynets have demonstrated their value as a security mechanism, primarily to learn about the tools, tactics, and motives of the attackers [68].

Honeypots and Honeynets

In this work we use the honeynet approach to improve our knowledge about current attacks against VoIP systems. Spitzner [141] defines a honeypot as: “A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource”. Provos and Holz [112] define a honeypot in more detail: “A honeypot is a closely monitored computing resource that we want to be probed, attacked, or compromised”.

Usually a honeypot has no legitimate activity and therefore any connection to it is most likely an unauthorized or malicious activity, e.g., probe, attack, or compromise. This means that all accesses to the honeypots in the honeynets are suspicious and represent attacks on the VoIP systems. Honeypots can be classified into *low-interaction* honeypots and *high-interaction* honeypots, whereby an interaction is defined as the activity between the attacker and the honeypot. [141]

Low-interaction honeypots are usually emulated services and/or operating systems. An attacker can only have as much activity as the emulated service provides. Therefore an emulated honeypot can be operated with minimal risks, because the attacker should never have access to the operating system. The advantage of a low-interaction honeypot is the simplicity. The disadvantage is that no new information about an attack can be gathered, because in the most cases the features of emulated services are limited. [141]

High-interaction honeypots do not use emulated services, they use real services and real operating systems. The main advantage is that it can gather much more information than with low-interaction honeypots, e.g., you can learn the behavior of the attackers or new attack vectors. However, the risk of a takeover by the attacker, and therefore the risk of an attack on other systems, increases. [141]

Honeypots can be used for research purposes or in production environments with the goal of identifying attacks. For research purposes the honeypot should gather enough information to identify the behavior of the attackers or showing the trends of attacks on special services. For production environments, the honeypot should help to protect an organization, e.g., detecting attacks. [141]

A honeynet is a network that contains one or more high-interaction honeypots to capture information about threats. [69]

> *Any interaction with a honeynet implies malicious or unauthorized activity.*
>
> (Honeynet Project [69])

In a honeynet all activities can be controlled and monitored, because it is a network with a single gateway. The primary advantage of a honeynet is their ability to gather extensive information about attacks. This information can be used, e.g., for analyzing attacks or for protecting production systems. [69]

Figure 2.8 shows an example architecture of a honeynet as designed by the Honeynet Project [69]. The architecture of a honeynet consists of a *honeywall* and one or more *honeypots*. The honeywall is a single gateway for the connections from and to the honeynet, i.e., all traffic has to go through the honeywall without being detected by the attacker. Usually the gateway is a

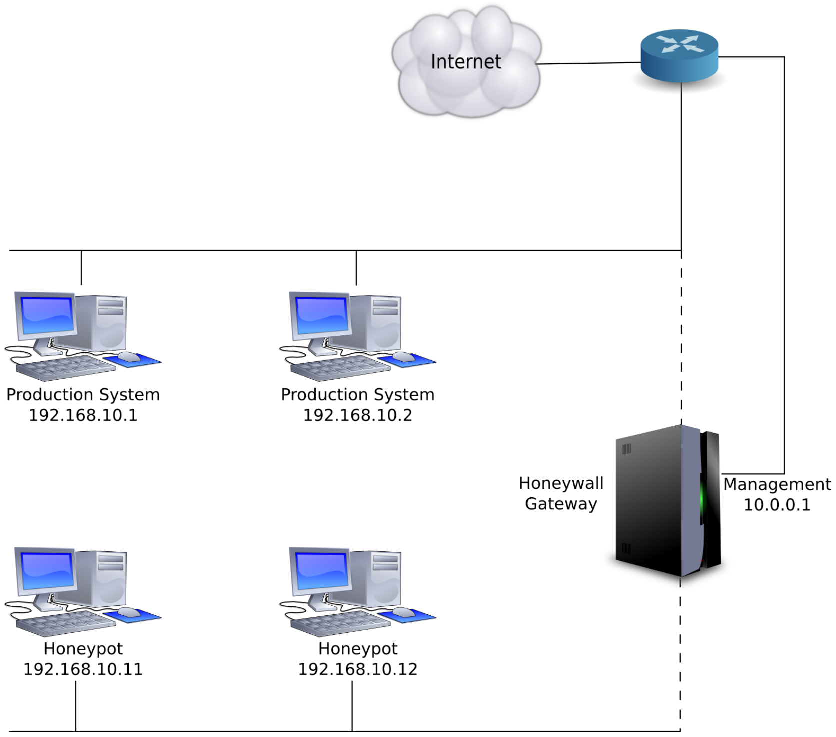


Figure 2.8: Example architecture of a honeynet [69].

layer 2 bridging device without any IP address. However, the honeywall can be accessed and configured with a separate management network. [69]

The key requirements for a honeywall are described as [69]:

- **Data control** helps to mitigate the risk of abuse to the honeynet by controlling the allowed connections. The challenge is implementing strong data control mechanisms versus detection of the honeynet by an attacker. Each attacker should be offered enough degrees of freedoms to get helpful information about attacks. But a higher degree of freedom leads to a higher risk of the honeynet being abused. The risk can only be minimized, but never entirely eliminated.
- **Data capture** monitors and logs all activities in the honeynet. The challenge is to capture as much data as possible without being detected by the attacker.
- **Data collection** combines all the collected data from the distributed honeynets.
- **Data analysis** helps to convert and analyze all the collected data.

In order to collect extensive information about known and unknown threats, the attacker needs enough freedom in the honeynet. But the risk for obtaining all this information is high. The risks of the operation of a honeynet can be divided into *harm, detection, disabling or violation*, as described in the Honeynet Project [69]:

- The honeynet can cause **harm** to non-honeynet systems. This can always happen, no matter what measures have been taken. Each organization must decide whether to accept this risk or not. For example, an attacker may break into a honeynet and carry out outbound attacks and successfully harm or compromise the intended victim.
- The honeynet can be **detected** by an attacker. If this happens the value of the honeynet is dramatically reduced. For example, attackers can ignore or bypass the honeynet, thus eliminating or decreasing its ability to capture information.
- An attacker can **disable** functionality of the honeynet, e.g., data control or data capture routines.
- **Violation** of the honeynet, e.g., attackers use the honeynet for criminal activities without attacking non-honeynet systems.

Honeynets are an appropriate approach for identifying attacks against VoIP systems and to learn about the tools, tactics, and motives of the attackers. However, the risks of operating the honeynet must be considered in the concept of a comprehensive VoIP honeynet solution.

2.4 Security of Communication Systems

This thesis focuses on a way to secure both the signaling and the media transmission of VoIP systems by applying and adapting state-of-the-art cryptography (such as Advanced Encryption Standard (AES)-Galois/Counter Mode (GCM) [94] and Fully Hashed Menezes-Qu-Vanstone (FHMQV) [126]). Methods and mechanisms provided by the science of cryptography are not the solution for all security problems of VoIP systems, but in most cases they are part of the solution [46]. The new mechanisms for VoIP security were chosen carefully after an in-depth analysis of the latest developments in the field of cryptography.

Historically, cryptography was the art of secret communication, and this art is almost as old as civilization itself. The main focus of classical cryptography was the problem of ensuring secrecy in transmitting messages. In the late 20th century, cryptography became a rigorous science. While the problem of secret communication (providing confidentiality) is still a major focus of modern cryptography, the science of cryptography concerns itself with a much broader array of topics today. [78]

Menezes *et al.* [98] provide the following definition of cryptography: “Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication.” Katz and Lindell [78] define cryptography as: “Cryptography is the scientific study of techniques for securing digital information, transactions, and distributed computations.” Goldreich [53, 54, 55] describes cryptography as the rigorous science that concerns itself with the design and analysis of abuse-resilient schemes and systems.

Cryptographic engineering, which is usually seen as separated from the science of cryptography, concerns itself with the application of cryptographic techniques and methods to real-world engineering problems. From an IT engineering point of view, cryptography itself is only one (often small) constituent of a secure system. However, it is usually a very critical one, e.g., for online banking or access restrictions. [46]

To improve the security of a system, all of its weaknesses have to be carefully taken into consideration. In most cases it is not sufficient to improve only one part of the system, such as the choice or implementation of the cryptographic algorithms.

> *A security system is only as strong as its weakest link.* (Ferguson *et al.* [46])

Important cryptographic tools are symmetric (shared-secret) ciphers, Message Authentication Code (MAC), authenticated encryption primitives, asymmetric (public-key) ciphers, digital signature schemes, and key exchange mechanisms, which are described in detail in [46, 30, 78, 98]. These techniques and methods are also the building blocks for the proposed VoIP security layer presented in Chapter 9.

Secrecy in transmitting messages is achieved by various methods of encryption and decryption (called ciphers). A fundamental assumption in cryptography was stated by Kerckhoffs in 19th century and is still an underlying principle of modern cryptography:

> *A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.* (Kerckhoffs [80])

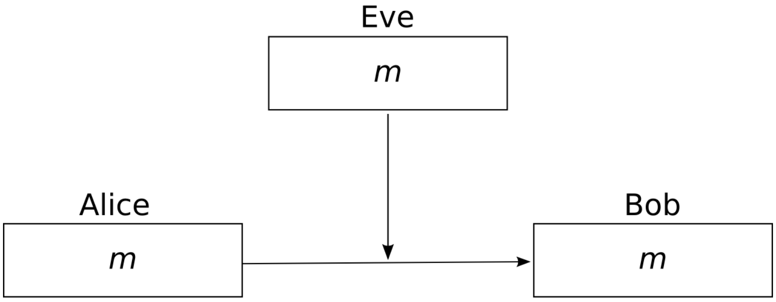


Figure 2.9: Communication without encryption based on Ferguson *et al.* [46].

Figure 2.9 shows an insecure communication channel between two communication partners (called Alice and Bob in this example) that is eavesdropped by a third, malicious party (called Eve).

Symmetric (Shared-Secret) Cipher

To prevent Eve from reading the content of the plaintext message m, Alice and Bob can use a symmetric cipher. The shared secret, the private key K, is shared between Alice and Bob (it is assumed that Alice and Bob have previously exchanged K, e.g., via a different secure channel [46]). Figure 2.10 shows Alice sending an encrypted message to Bob. Alice uses the encryption function $E(\cdot, \cdot)$, parameterized with the shared secret K, to encrypt the plaintext message m. She obtains the ciphertext c, which she subsequently sends to Bob. Bob decrypts the received ciphertext c using the decryption function $D(\cdot, \cdot)$, which he also parameterizes with the shared secret K. Even if Eve knows the encryption and decryption functions ($E(\cdot, \cdot)$, and $D(\cdot, \cdot)$), but is not in possession of the shared secret K, she can only see the ciphertext c, but cannot read the plaintext message m anymore just by wiretapping the communication channel. [46]

Symmetric ciphers can be distinguished between block ciphers (such as AES or 3DES) and stream ciphers (such as Vernam's one-time pad [100]). A block cipher uses plaintext of fixed length for the encryption and a stream cipher operates on streams of plaintext [30]. The following challenges have to be resolved to use symmetric cipher in practical applications [46]:

- The shared secret needs to be transmitted over a secure channel.
- Only the secrecy of the message is protected. An active adversary can still maliciously alter the encrypted message while in transmission without necessarily being detected.

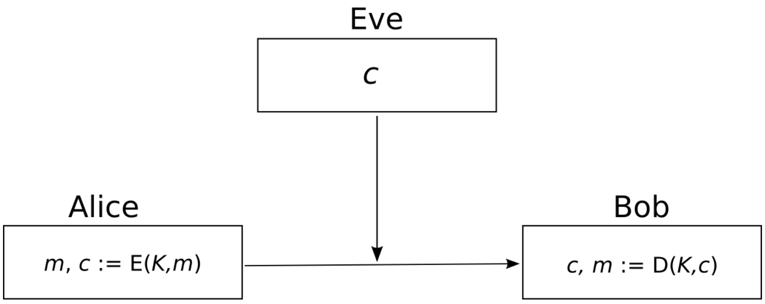


Figure 2.10: Communication with symmetric encryption based on Ferguson *et al.* [46].

Message Authentication Codes

A *MAC* is a cryptographic tool to detect tampering with a message and thus ensures *message integrity*. This is achieved by an additional tag that can only be computed by a party that is in possession of a secret key. A message authentication code consists of two functions: The signing function $S(K, m)$ takes as input the secret K together with a message m, and outputs a tag t. The verification function $V(K, m, t)$ takes as input the secret key K, a message m and a tag t. [30]

In practice, constructions for MAC are most commonly based either on block ciphers or on cryptographic hash functions. An example for a block cipher based MAC scheme is Cipher Block Chaining (CBC)-MAC. An example for MAC scheme based on cryptographic (collision-resistant) hash functions is the Hash-based Message Authentication Code (HMAC) as defined in RFC 2104 [86].

A MAC is needed if a message needs integrity protection. If a message needs both integrity protection and confidentiality protection, an authenticated encryption mechanism is often used.

Authenticated Encryption Schemes

An authenticated encryption scheme provides both confidentiality and integrity protection. Whereas a conventional symmetric cipher (such as AES in Counter Mode (CTR) or CBC mode) is designed to achieve confidentiality, an authenticated encryption scheme's goal is designed to achieve both confidentiality, as well as ciphertext integrity. This basically means confidentiality protection against an active adversary who tampers with the ciphertext. [12]

Schemes such as Offset Codebook Mode Version 1 (OCB1) mode encryption, Counter with CBC-MAC (CCM) mode encryption, or GCM encryption are prime examples of authenticated encryption schemes.

But neither an authenticated encryption scheme, nor a MAC scheme, protects against *replay attacks*. Ferguson *et al.* [46] describes a replay attack, as a recorded and later resent message by an attacker. Special measures must be taken to protect a communication protocol against replay attacks. Common techniques incorporate unique sequence numbers or timestamps in the integrity protected data.

Asymmetric (Public-Key) Ciphers

Symmetric ciphers require a shared secret known by the communicating parties. The problem of how such shared secrets should be exchanged, before they can be used to secure a transmission channel, has been left open so far. Of course, the shared secret can be exchanged on a second secure channel, but this is often impractical for most IT systems. Public-key cryptography provides a more practical solution to this problem.

The basic idea of public-key cryptography is that a key is divided into two parts: A public key P, which does not need to be kept secret, and a corresponding secret key S. Each communication party is in possession of a distinct key pair (P, S). The key pair has the property that, knowing only the public key P, it is computationally hard but not impossible to derive the corresponding secret key S. [30]

Figure 2.11 illustrates the concept behind an asymmetric (public-key) cipher: Bob and Alice each have their own distinct key pairs. Bob's key pair is (P_{bob}, S_{bob}). Alice wants to send a message m to Bob. Alice uses Bob's public key P_{bob} to encrypt the message m destined for Bob, and obtains the ciphertext c. Bob decrypt the received ciphertext c with his secret key S, and thus obtains m. [46]

Commonly used asymmetric ciphers are either based on the famous Rivest-Shamir-Adleman (RSA) [117] trapdoor one-way function, or on the hardness of the discrete logarithm (Dlog) problem (such as ElGamal [42] or elliptic curve groups [136]). [78, 30]

In practice, an asymmetric cipher is rarely used to encrypt the plaintext message directly. This has security reasons (e.g., vulnerable to chosen-plaintext attacks), as well as performance reasons (symmetric ciphers are notably faster). Instead, *hybrid encryption* is employed: encrypt the shared secret with an asymmetric cipher, and use a symmetric cipher after that. [131]

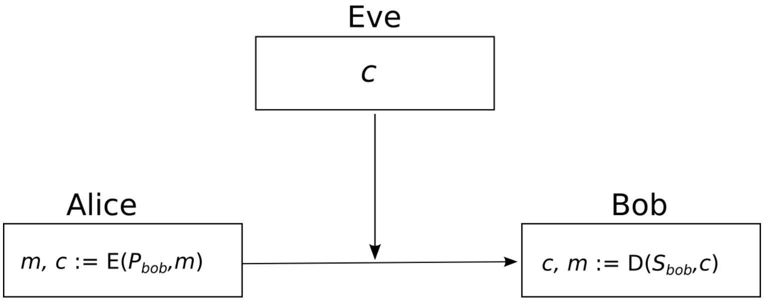


Figure 2.11: Communication with public-key encryption based on Ferguson *et al.* [46].

Although, with an asymmetric cipher, messages can be securely exchanged with just the knowledge of the recipient's public key, the problem of how to get hold of the right public key remains: when the public key is exchanged on an insecure channel, without further protection, an attacker might tamper with the transmission of the original public key and swap it with his own key. That way, the attacker can intercept (and possibly modify) all communication. Bishop [16] describes this attack as a MitM attack. The problem of distributing and verifying public keys is addressed with a *Public Key Infrastructure (PKI)*. The most important cryptographic tool for a PKI is a digital signature scheme, as discussed in the next subsection.

Digital Signature Schemes

Handwritten personal signatures on conventional documents are intended to guarantee authentication and non-repudiation [30]. Digital signature schemes have been invented in order to create a tool that mirrors these properties in IT systems. However, because it is easy to replace the whole message or arbitrary bytes during transmission in the digital world, it does not suffice just to append a signature to a digital message like signing a paper document. Rather, a digital signature needs to be a function of the message, with the property that forgeries are hard to make.

Similar to MAC for symmetric mechanisms, in a public-key setting, a digital signature guarantees the integrity of the signed message [30]. However, a digital signature can be verified by anyone who knows the public key of the signer. Moreover, only the party in possession of the secret private key can be the originator of a particular signature that verifies correctly with the corresponding public key.

Authenticated Key Exchange

The main advantage of key exchange mechanisms based on the Diffie-Hellman Key Agreement protocol [35] is that they provide *Perfect Forward Secrecy (PFS)*:

> *An authenticated key exchange protocol provides perfect forward secrecy if disclosure of long-term secret keying material does not compromise the secrecy of the exchanged keys from earlier runs.* (Diffie *et al.* [36])

That means, that encrypted communication from past sessions cannot be decrypted, even after the secret keys of the communication partners' asymmetric key pairs have been compromised. This fits also the design principle for cryptographic protocols, as described by Krawczyk [85]:

> *A good security system is not one that denies the possibility of failures but rather one designed to confine the adverse effects of such failures to the possible minimum.*
> (Krawczyk [85])

Protocols such as MQV [96, 90] and its successors HMQV [85], FHMQV [126], OAKE [159] fulfill this design principle by trying to reach an optimum of security when faced with leakage of various secret information.

2.5 Conclusion

This chapter considered the basics of IT security and VoIP systems, as well as the fundamentals of using honeynets to identify real-world attacks. The current security concerns of VoIP systems (based on the classification of VoIPSA) were also presented. The chapter also covered the use of cryptographic algorithms for securing communications. These basics of IT security for VoIP systems are the foundations of the following work. The next chapter will look at state-of-the-art VoIP security in more detail.

CHAPTER 3

State-of-the-Art of VoIP Security

Contents

3.1 Common Security Process

Information security is a dynamic process which has to be continuously monitored and controlled [130]. Many methods and procedures for IT security analysis are well described in various standards (e.g., ISO/IEC 27001 [82] or the standards 100-1, 100-2 and 100-3 from the Bundesamt für Sicherheit in der Informationstechnik (BSI) [20]) in order to get a reproducible set of actions and validation criteria.

A simplified security process is based on four main steps (as presented in Figure 3.1): *identification of threats and vulnerabilities*, *analysis and evaluation of risks*, *selection and implementation of countermeasures* and *examination of effectiveness*. Similar to other business processes or management processes the whole information security process is subject to a life cycle. A security life cycle is often defined similar to the “Plan-Do-Check-Act” (PDCA) model (as shown in Figure 3.2), which is also mentioned in ISO/IEC 27001 [82].

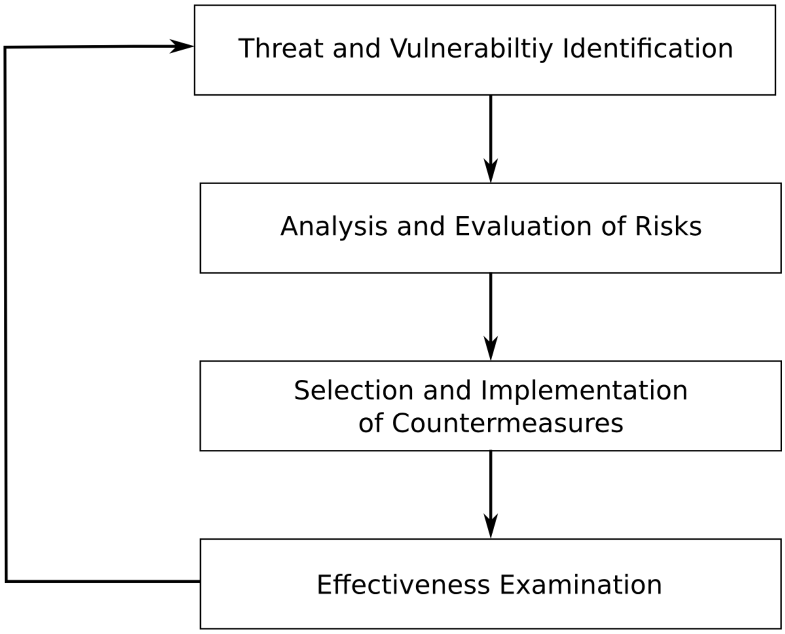

Threat and Vulnerabiltiy Identification
Analysis and Evaluation of Risks
Selection and Implementation of Countermeasures
Effectiveness Examination

Figure 3.1: Applied security process for protecting VoIP systems against current threats.

Schmidt [130] describes the phases of the security life cycle as:

1. **PLAN**: Identify the need for some action for an actual state. Possible methods are bottom-up or top-down. For bottom-up risk assessment is a suitable method. For the top-down approach auditing is a suitable method. If the need for action is identified, then the planning of suitable security measures is also contained in this phase. Various security measures are possible on different levels, but it is essential to find measures which are economically viable.

2. **DO**: Implementation of the planned security measures. The implementation costs and the acceptance of the measures have to be considered.

3. **CHECK**: This phase checks if the implemented measures fulfill not only the security goals but also the economic goals.

4. **ACT**: Optimization and improvement of regular operations with these security measures.

This work applies this streamlined security process (as seen in Figure 3.1) to the field of VoIP security. For the identification and evaluation of the risks a common risk assessment process was used, including a threat and vulnerability analysis. The risk depends on the likelihood of

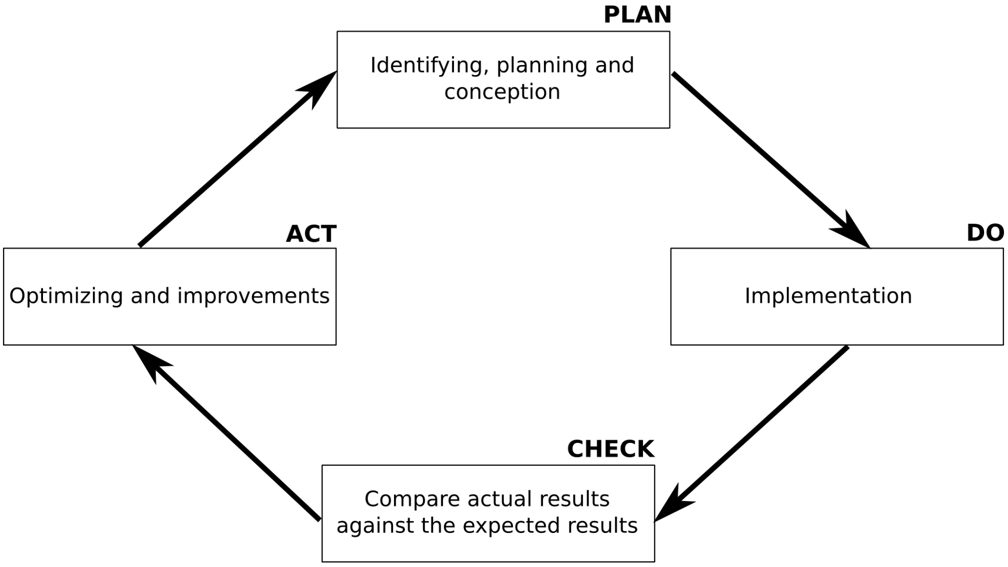

PLAN
Identifying, planning and conception
DO
Implementation
CHECK
Compare actual results against the expected results
ACT
Optimizing and improvements

Figure 3.2: Life cycle of a security process (PDCA model based on ISO/IEC 27001 [82]).

an attack's occurrence and the amount of damage which it can cause. A "tolerable risk" is the highest acceptable risk of a process or state. [37]

Schmidt [130] defines the risk formula as:

$$Risk = Consequence * Likelihood \tag{3.1}$$

The theoretical analysis of VoIP vulnerabilities (Section 3.2) and empirical analysis with the proposed VoIP honeynet approach (Chapter 5) help to identify threats and vulnerabilities of VoIP systems. By analyzing these previously identified threats and vulnerabilities (Chapter 6, 7 and 8) the risks to the system are assessed. Chapter 9 considers the selection and implementation of countermeasures against the main risks. The examination of the effectiveness of implemented countermeasures is the last step of the workflow and covered in Chapter 10. If the effectiveness of the countermeasures is not satisfactory or the remaining risk level is not acceptable the workflow starts again with the threat and vulnerability identification step. This process has to be carried out periodically to evaluate if the selected protection mechanisms of VoIP systems are still sufficient.

3.2 Security Aspects of VoIP Systems

Generally, VoIP is based on the IP stack and attacks on each layer of the IP stack may occur. The attack surface can be divided into horizontal and vertical attack surfaces, as described in Chapter 2.

A VoIP security stack could be compromised on three different layers: signaling, media transmission and key management [108]. Therefore, all three layers must be protected for secure VoIP communications. Many attacks on the application level of VoIP systems (e.g., DoS attacks or service abuse) are well described but not completely resolved. Keromytis [81] gave an overview of the status of VoIP security research, identifying the two specific problem areas of DoS and service abuse, where additional research effort should be focused. Analyzing and describing real-world attacks are needed to get more detailed and accurate information about these attacks.

An introduction to SIP, comprehensive threat analysis of SIP and especially the vulnerabilities regarding Spam over IP Telephony (SPIT) are published in [161, 39, 49, 148].

Often no cryptographic security mechanisms are used for performance reasons in VoIP systems to ensure quality of speech [137, 88]. This means that in typical VoIP communications the metadata may be easily obtainable and possibly the media content as well.

Securing the communication of VoIP systems using encryption has been addressed by multiple authors (e.g., Palmieri and Fiore [106], Gurbani and Kolesnikov [63], and Perez-Botero and Donoso [108]). It was identified that most current security mechanisms for VoIP do not offer protection against rogue proxy servers. Many security mechanisms (e.g., Transport Layer Security (TLS)) do not use end-to-end encryption, allowing intermediary proxies to have access to the unencrypted payload. To overcome this issue, Palmieri and Fiore [106] suggest introducing an additional encryption and authentication layer to SIP and RTP based on X.509 user certificates.

Secure Real-Time Transport Protocol (SRTP) (a profile of RTP) can provide confidentiality, message authentication, and replay protection to the RTP and RTCP traffic, as described in RFC 3711 [10]. White *et al.* [157] presented an approach for unmasking parts of an encrypted VoIP communication, where the interaction of variable bit-rate codecs and length-preserving stream ciphers leak information. Because the specification for SRTP does not alter the size of the original payload, the plaintext frame size is the same as the ciphertext frame size. This correlation is leveraged to model phonemes as sequences of lengths of encrypted packets.

Wright *et al.* [158] presented an approach to uncover spoken phrases in default SRTP encrypted VoIP communications. They recommend the use of padding as a mitigation technique.

Gurbani and Kolesnikov [63] compare security protocols which allow the establishment of a shared secret used by SRTP for media encryption. They looked at Session Description Protocol Security Descriptions (SDES) [4], Phil Zimmermann's Real-Time Transport Protocol (ZRTP) [160] and Datagram Transport Layer Security (DTLS)-SRTP [93], in terms of their security features, identifying SDES as the weakest.

- **SDES**: Is used to negotiate keys for media stream encryption [4]. The secret key is transported in the Session Description Protocol (SDP) attachment of a SIP message. Each SIP proxy can see the secret key and is able to decrypt the media data.
- **ZRTP**: Is a key-agreement protocol to negotiate keys for the encryption of the media data between two communicating parties [160]. One of the biggest disadvantages of ZRTP results from the use of the Diffie-Hellman key exchange, which does not offer protection against MitM attacks, and therefore must be extended, e.g., with Short Authentication String (SAS). However, SAS has the drawback of needing a verbal cross-check by the communicating parties.
- **DTLS-SRTP**: Similar to TLS, DTLS offers integrated key management, parameter negotiation, and secure data transfer to a datagram protocol (such as User Datagram Protocol (UDP)) [116, 93]. Fardan and Paterson [3] present a way to recover plaintext from a DTLS connection when CBC mode encryption is used.

Aghila and Chandirasekaran [1] present MitM attacks against the existing key exchange protocols namely Multimedia Internet KEYing (MIKEY) [6], ZRTP and SDES. Additionally, DTLS-SRTP and ZRTP come with *significant computation and communication costs* for the key-exchange or the use of PKI, which is the reason why it is not in widespread use [62].

Perez-Botero and Donoso [108] compared not only media keying protocols, but also suggest Secure/Multipurpose Internet Mail Extensions (S/MIME) to be ideal for use in VoIP environments, because it provides end-to-end confidentiality and leaves the SIP headers untouched. However, S/MIME is not yet widely supported in popular VoIP software and the overhead of using the PKI infrastructure may be too high. The idea of untouched headers will be reused in our proposed approach.

Web Real-Time Communication (WebRTC) enables real-time voice and video communication capabilities via simple JavaScript for web browsers [75]. Due to its web-integration, WebRTC is

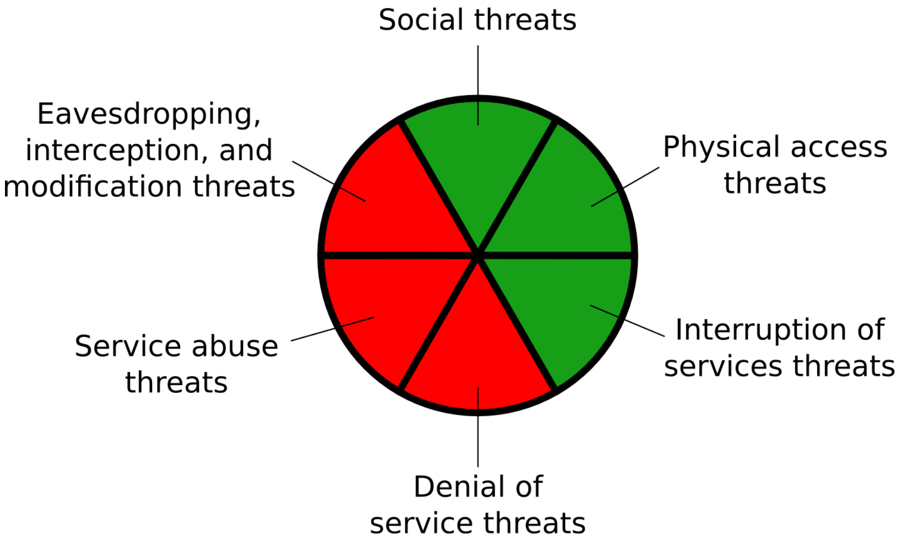

Social threats
Eavesdropping, interception, and modification threats
Physical access threats
Service abuse threats
Interruption of services threats
Denial of service threats

Figure 3.3: Identified threats to VoIP systems classified by taxonomy of VoIPSA.

an increasingly popular media exchange technology which uses Secure Socket Layer (SSL)/TLS for security and consequently inherits these security concerns (as discussed in Section 3.5).

Based on these findings, Figure 3.3 presents the categories (based the VoIPSA classification of from Chapter 2) for which a major threat (marked as red) was identified and for which protection mechanisms on an application level can help to establish secure VoIP calls. The analyses from Chapter 6 and 7 are used to gain detailed information about real-world attacks and to develop suitable countermeasures.

3.3 State-based Surveillance

The recent revelations about state organized compromising of networks (i.e., the National Security Agency (NSA) surveillance programs), changed the security principles of the Internet. As the Guardian [146] and Spiegel online [32] reported, various network components (e.g., routers or firewalls), cryptographic algorithms, end-devices, as well as service providers (e.g., for email or VoIP services) or even Internet providers have been compromised. Figure 3.4 depicts the opportunities (red circles) for the NSA to collect communication data. This means that both end-devices (e.g., computers, mobile phones or local WLAN networks) as well as global network components (e.g., routers or firewalls) can be compromised and the NSA is able to collect

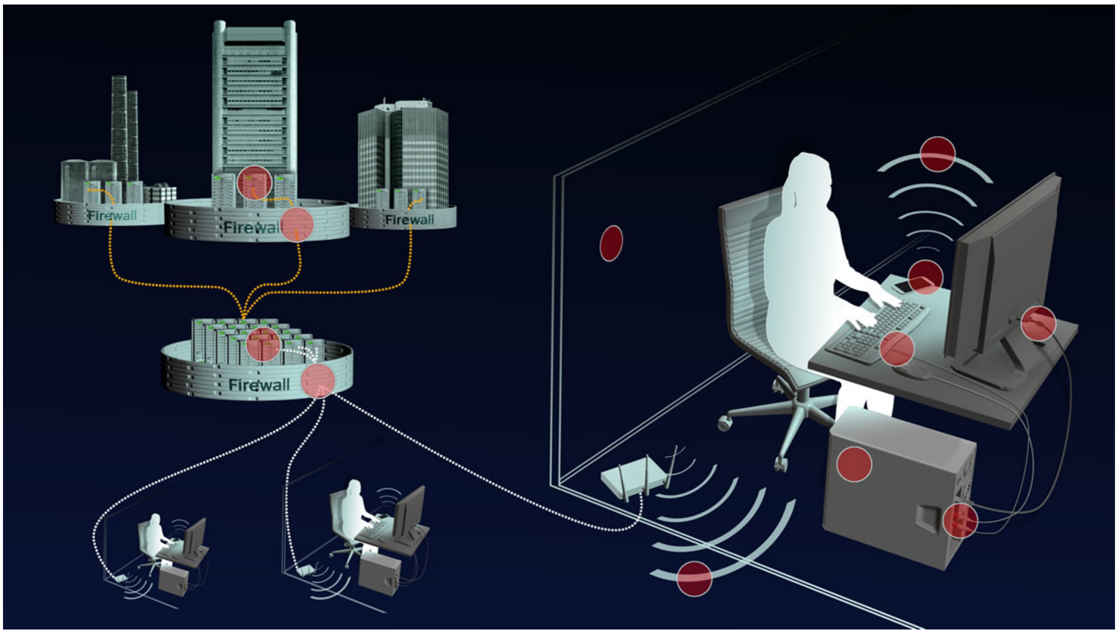


Figure 3.4: Spy catalog of the NSA as presented in Spiegel online [32].

almost all traffic of special targets (in most cases not for the masses) in the Internet [32]. But Snowden[1] in an online Q&A with the Guardian [145] had some good news:

> *Encryption works. Properly implemented strong crypto systems are one of the few things that you can rely on.* (Snowden in the Guardian [145])

Based on these revelations secure Internet communication systems have to consider strong end-to-end encryption to avoid interception and interpretation of data during transmission. Metadata also leaks a lot of valuable information about the participants and should also be hidden from eavesdroppers (e.g., by encryption). Still, end-to-end encryption only works if the client (which has access to unencrypted data) is not compromised. [147]

3.4 Methods for Capturing Attacker Behavior During Attacks

To protect systems from attacks it is necessary to understand the attacks as well as the attackers' behavior. If an attack is captured it can be analyzed in detail later to identify possible vulnera-

[1] http://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance

bilities of the system. One method to detect attacks in a network is to use an IDS. Scarfone and Mell [127] described an IDS and Intrusion Prevention System (IPS) as:

- **IDS**: "Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices." [127]
- **IPS**: "Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents." [127]

Scarfone and Mell [127] categorized an IDS or IPS as:

- **Network-Based**: Monitors the network traffic and analyzes the network (including the application protocol) to identify suspicious activity.
- **Wireless**: Monitors wireless network traffic and analyzes the traffic (including the wireless network protocol) to identify suspicious activity.
- **Network Behavior Analysis**: Monitors network traffic to identify attacks that generate unusual traffic flows (e.g., Distributed Denial of Service (DDoS) attacks or policy violations).
- **Host-Based**: Monitors the characteristics of a single host and the events occurring within that host to identify suspicious activity.

A challenge of IDS or IPS is to distinguish attacks from regular traffic. This may lead to a loss of attack information.

Another method to detect attacks is the use of a honeynet in which any traffic is malicious (as described in Chapter 2). Through the use of a honeynet, attacks against VoIP systems can be captured and analyzed. The distribution of the captured attacks shows the likelihood of their occurrence. Therefore, a honeynet seems to be an ideal method to identify threats and risks to VoIP systems. The conventional honeynet architecture does not have special mechanisms for efficiently analyzing VoIP calls. Based on the conventional architecture, additional enhancements to obtain more detailed and better information are needed.

Nassar *et al.* [103] described an architecture for a VoIP honeypot. They show the design and the implementation of a VoIP honeyphone agent and an inference algorithm to classify the various attacks. They do not capture attacks against VoIP servers (such as Asterisk Private Branch Exchange (PBX) or SIP Express Router), they only recognize attacks against a predefined UA in the production infrastructure.

Dionaea [83] is a low-interaction honeypot system with different modules for multiple network protocols. One of these modules is SIP for VoIP systems. Different VoIP phones can be simulated by using different user agent strings. Therefore, Dionaea could be a possible addition to a VoIP honeynet solution if there is a need for honeypots with low-interaction.

VoIP Honey [152] is an application which provides a set of tools for building a honeynet. The VoIP services such as Asterisk PBX or OpenSER are emulated and not real services. The authors recommend using VoIP Honey only for testing in a strictly controlled network environment without a direct Internet connection since the application is not yet stable. The VoIP honeynet concept presented in this thesis is more flexible, since it can include honeypots which are not emulated as well.

Do Carmo *et al.* [38] described a VoIP honeypot approach of (called Artemisa) to support security in VoIP domains. This VoIP honeypot registers itself as a softphone to VoIP registrars and collects information about incoming VoIP calls. In addition, different actions can be defined, e.g., block different IP addresses of callers categorized as attackers. Artemisa is just one possible honeypot which acts as a VoIP client. This approach provides similar functionality as a conventional honeynet, but it can only be used in existing VoIP domains as a UA and not in independent solutions or locations. VoIP servers like registrars or proxies cannot be used and attacks against these servers are not part of the analysis. Our solution also supports capturing and comprehensive analysis of attacks against all components of the VoIP system.

Valli and Al-Lawati [151] described simple methods for the detection of SIP-based attacks. This approach uses an IDS and simple emulated honeypots to detect attacks. Furthermore, there is no reporting system for dynamic and automatic analysis. Valli [150] described results of some selected events that were collected in their basic VoIP honeypot. Because the setup uses simple emulated honeypots, the analysis is not comprehensive enough to use this information to protect real-world systems.

A description of the scanning behavior of botnets is covered by Dainotti *et al.* [27]. Numerous websites feature information about VoIP attacks, e.g., SANS [125] shows the number of connections to port 5060 or Gauci's website on Sipviscious [51] gives details about recognized attacks sporadically, but reliable data for custom analysis or automatic generated reports are not available.

Various works in the field of honeynet technologies exist, but an overall solution for capturing, monitoring, analyzing and reporting of real-world VoIP attacks with sophisticated honeypots is missing. This is crucial for identifying threats and risks to VoIP systems and to get more information about real-world VoIP attacks. Therefore, we implemented our own honeynet solution, including a comprehensive VoIP analyzing engine, based on open-source software (as described in Chapter 5) for capturing attacks and identifying threats and risks to VoIP systems.

3.5 Cryptography in Communication Systems

To use cryptography as part of the security solution it is essential to use state-of-the-art and proven cryptography, because some cryptographic algorithms have been compromised, as the Snowden documents describe [146].

As Kerckhoffs [80] already mentioned, the cryptographic algorithms should be public knowledge and only the key must be private. Therefore, a reliable key exchange mechanism is essential for VoIP systems.

In 1976 Diffie and Hellman presented the first practical solution to the problem of how to exchange secret keys to bootstrap secure communication with a symmetric cipher. With their seminal work [35], they are considered (together with Merkle, whose work from 1974 did not find a publisher until 1978 [99]) to be the inventors of practical public key cryptography.

The basic Diffie-Hellmann scheme [35] is only resistant to passive adversaries, since there is no mechanism in place to protect the integrity of the exchanged values. So the scheme is vulnerable to a MitM attack. An active attacker (called Eve), could tamper with the communication and establish a secret with both Alice, pretending to be Bob, and with Bob, pretending to be Alice. Eve would then forward all communication between Alice and Bob, while intercepting, and possibly manipulating, the plaintext messages. [46]

Elliptic Curve Cryptography (ECC) [101] is one way to instantiate public-key cryptography protocols, for example implementing digital signatures and key agreement. The practical benefits of using elliptic curves are well-understood: they offer smaller key sizes [91] and more efficient implementations [14] at the same security level as other widely deployed schemes such as RSA, as Bos *et al.* [18] presented. NIST recommended key-sizes for similar strength of security for symmetric algorithms, hash functions, RSA and elliptic curve algorithms are presented in Table 3.1. Thus, ECC requires a key-size of only 256 bits to provide 128 bits of cryptography strength compared to RSA, which requires a key-size of 3072 bits for the same cryptography strength [9].

Bits of Security	Symmetric Algorithm	Hash Function	RSA Key Size	ECC Key Size
80	Triple DES (2 keys)	SHA-1	1024	160
112	Triple DES (3 keys)	SHA-224	2048	224
128	AES-128	SHA-256	3072	256
192	AES-192	SHA-384	7680	384
256	AES-256	SHA-512	15360	512

Table 3.1: Recommended key sizes for similar cryptography strength based on NIST [9].

Authenticated key-exchange protocols have matured through decades of academic evolution [36, 17, 85]. The chosen key-exchange protocol for the VoIP security layer presented in Chapter 9 is elliptic curve FHMQV [126]. FHMQV is a patent-free variant of the provably secure Hashed Menezes-Qu-Vanstone (HMQV) [85] protocol, with slightly stronger security guaranties. The HMQV protocol itself is a variant of the well-known Menezes-Qu-Vanstone (MQV) protocol [90], which is probably one of the most heavily analyzed cryptographic protocols.

FHMQV is based on carefully analyzed predecessor protocols and the security of the algorithm has been evaluated several times [97, 95, 126]. For this purpose, FHMQV provides stronger security guarantees than its predecessors and moreover is patent-free. Because of its implicit authentication, a very efficient handshake avoiding unnecessary messages can be obtained. This makes the handshake is a lot less cumbersome than the historically grown SSL handshake. Due to the simplicity of the design, FHMQV avoids unnecessary complexity and error-proneness during design and implementation. Through the authentication mechanism, FHMQV provides protection not only against passive eavesdropper, but also against active attackers such as MitM.

To securely transmit VoIP data, a secure channel is used for the implementation of the VoIP security layer. A *secure channel* for transmitting application data is a channel protocol that is both a *secure network authentication* protocol and a *secure network encryption* protocol, as Canetti and Krawczyk [22] and Nagao *et al.* [102] defined.

Authenticated encryption [12] is the preferred way for providing confidentiality against both passive, as well as active attackers (the congruous threat model in a networked setting) and bars risks from hand-knit constructions, such as MAC-then-encrypt [28, 84]. The AES-GCM modus [94] was chosen for building a secure channel in the VoIP security layer over the more elegant and efficient Offset Codebook Mode (OCB) modus [120, 87], because it is patent-free and has hardware support on recent Intel CPUs.

Ferguson [45] and Joux [76] identified weaknesses of the GCM mode, which have been taken into careful consideration for the implementation. The main weaknesses are: short authentication tag weakness, nonce reuse, plaintext is limited to 68.7 GB and security proof gets complicated with different nonce/IV sizes. Rogaway [119] sees the critiques of GCM as significant issues, but not as fatal.

Various implementations of FHMQV exist (e.g., Curve25515 [13] or M-511 [5]) and some are unsecure as Bernstein[2] and Lange[3] [15] presented. For the implementation of the VoIP security layer (Chapter 9) the elliptic curve M-511 [5] (formerly named Curve511187), recommended by Bernstein and Lange [15], will be used. For symmetric encryption AES-256 and for hash function SHA-512 was chosen. They offer proven high cryptographic strength for secure VoIP communications.

The design and implementation of cryptographic protocols is prone to innumerous pitfalls, as is evident from the history of SSL/TLS [154, 19, 8, 79, 40, 107, 3]. However, building secure protocols was far less thoroughly understood back when SSL was conceived than it is now. Many formalized notions and best-practice advice simply did not yet exist, and provable secure cryptographic primitives were in their infancy.

Due to the additional effort of encrypting signaling and media sessions, additional hardware resources are required. Work by Kulin *et al.* [88], and Reason and Messerschmitt [115] show the impact of encryption on various indicators, e.g., response times and throughput. In this context, Epiphaniou *et al.* [43] also considers the impact of cryptographic functions on user experience.

3.6 Conclusion

This chapter presented a short introduction to common security workflows, as well as the security problems and countermeasures of current VoIP systems. Current honeynet solutions for capturing attacks were also considered. Finally the chapter concluded with the state-of-the-art of cryptography in communication systems. In order to gain a better understanding of the attacks, the next chapter will look at the most common VoIP protocols in more detail.

[2] http://cr.yp.to/djb.html

[3] http://hyperelliptic.org/tanja/

Part I

VoIP Attack Collection – Automated Capturing Using a Honeynet Approach

CHAPTER 4

Introduction to VoIP Protocols for Capturing Attacks

Contents

4.1 Introduction

To be able to capture, analyze and understand attacks and attacker behavior, it is necessary to understand the details of IT systems and of the services offered (e.g., VoIP or Secure Shell (SSH)). For the selection of the right protection mechanisms the details of VoIP protocols are also important, because the protocol or parts of the protocol can also be the weakest link.

In general, the signaling and the media transmission phases of VoIP communications use different protocols. Signaling protocols are divided into Session Control Protocols and Media Control Protocols. The session control protocol is used for call initiation, call control and call termination. The media control protocol is used for the communication between media gateways. Some well-known signaling session control protocols are SIP, H.323 or InterAsterisk eXchange (IAX), and well-known media control protocols are Media Gateway Control Protocol (MGCP) or Megaco (H.248). [148, 44]

This work focuses on VoIP systems based on the signaling protocol SIP and the media transmission protocol RTP (both are briefly described in Chapter 2), because they are both widely used and well documented.

4.2 Session Initiation Protocol

The signaling protocol SIP is described in RFC 3261 [124] as "SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls." SIP is a plaintext protocol first standardized by the Internet Engineering Task Force (IETF) in RFC 2543 [65] in 1999 and replaced by RFC 3261 [124] in 2002.

SIP is a signaling protocol for bi-directional communications (not only for VoIP) and is similar to Hyper Text Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), which are also text-based. SIP communications use a request-response-protocol, i.e., the source sends a SIP request message and receives a SIP response message. SIP is an inherently stateful protocol and even uses the HTTP Digest Authentication [48] for user authentication. In its simplest form SIP uses the transport protocol UDP [109], but others are also possible, e.g., Transmission Control Protocol (TCP) [110] or Stream Control Transmission Protocol (SCTP) [143]. The signaling packets can be relayed through a number of SIP proxy servers, in contrast to the media protocol, which aims to exchange the packets directly between the endpoints. The exception to this is the use of media gateways to bridge different networks, e.g., SIP and PSTN, in which case the packets must pass through the gateway. [81, 124]

SIP Header

A valid SIP request message consists of a *Request-Line* and a *header*. The payload is not mandatory in a SIP message. The header requires the fields: *To*, *From*, *CSeq*, *Call-ID*, *Max-Forwards* and *Via*. The Request-Line requires the fields: *Method*, *Request-URI* and *SIP-Version*. [124]

SIP and SIPS Uniform Resource Indicators

A SIP or SIPS URI identifies a communication resource in the VoIP network (similar to HTTP or SMTP), as described in RFC 3261 [124]. SIP URIs are used in various places including the fields *To*, *From* and *Contact*. A SIPS URI means that an endpoint must be contacted in a secure manner, e.g. by using a transport security mechanism such as TLS [124]. Listing 4.1 shows the common format of the SIP URI as defined in RFC 3261 [124].

```
sip:user:password@host:port;uri-parameters?headers
```

Listing 4.1: SIP URI definition from RFC 3261

The tokens of the SIP URI have the following meaning [124]:

- **user**: Identifies a user of a VoIP domain.
- **password**: The password associated with the user.
- **host**: The IP address of the SIP service.
- **port**: The port number on which the service is listening on the host.
- **URI parameters**: The parameters have an impact on the handling of a request, e.g., the *transport* attribute which determines the used transport protocol, or the *user* attribute which indicates that the username contains a phone number.
- **headers**: The header fields determine the fields in the request message, e.g., a SIP URI can contain headers that are carried over from the URI to the SIP Request.

Listing 4.2 describes some examples of SIP URIs. The last line shows a SIP URI with the user "alice", password is "secretword", the host is "example.com", the URI parameter "method= REGISTER" which will indicate the method of the SIP request and the header field "to".

```
sip:alice@example.com
sip:alice:secretword@example.com;transport=tcp
sip:+1-123-456:secretword@example.com;user=phone
sip:alice:secretword@example.com;method=REGISTER?to=alice%40example↩
    .com
```

Listing 4.2: Examples of SIP URIs for communications without transport security mechanisms

SIP Request/Response Message

Each SIP message has a message type (also called a method), followed by the remaining header and a message body. A request message is a SIP message between two endpoints (from a UAC to a UAS) which is used to invoke a particular operation. A response message is a SIP message which is sent by the server to the client and indicates the status (including a reason phrase and sometimes additional information) of the request made by the client. [124]

The main SIP request message methods based on various RFCs are:

- **INVITE**: Initiate a new call session (RFC 3261 [124] and RFC 6026 [140]).
- **ACK**: A UAC received a final response to an INVITE message (RFC 3261 [124]).
- **OPTIONS**: Querying a user agent about the endpoint capabilities (RFC 3261 [124]).
- **REGISTER**: A client registers with a SIP Registrar Server (RFC 3261 [124]).
- **BYE**: A termination of the call is indicated (RFC 3261 [124]).
- **CANCEL**: Cancel an INVITE request (RFC 3261 [124]).
- **INFO**: Carrying of session related control information (RFC 6086 [67]).
- **MESSAGE**: Transport instant messages, such as chats (RFC 3428 [21]).
- **NOTIFY**: Notify the subscriber of a new event (RFC 6665 [118]).
- **PRACK**: Ensures reliable delivery of provisional responses (RFC 3262 [123]).
- **PUBLISH**: Publish an event state within the SIP events framework, for example “do not disturb” (RFC 3903 [105]).
- **REFER**: Used for call transfers, e.g., for conference calling (RFC 3515 [139]).
- **SUBSCRIBE**: Request information about the status of a service session (RFC 6665 [118]).
- **UPDATE**: Modify the state of a session without impact on the state of the dialog (RFC 3311 [122]).

The main SIP response messages are [124]:

- **Informational:** e.g., 100 (Trying) or 180 (Ringing) – indicate the request was received but not yet accepted.
- **Success:** 200 (OK) – indicate the request was received successfully and accepted. The information returned with the response depends on the method used in the request.

- **Redirection:** e.g., 301 (Moved Permanently) or 302 (Moved Temporarily) – a further action is required to complete the request.

- **Client Error:** e.g., 400 (Bad Request), 401 (Unauthorized), 404 (Not Found) or 407 (Proxy Authentication Required) – bad syntax found in the request or cannot be executed on this VoIP server.

- **Server Error:** e.g., 500 (Internal Server Error) or 503 (Service Unavailable) – the VoIP server failed to answer the request.

- **Global Failure:** e.g., 600 (Busy Everywhere) or 604 (Does not exist anywhere) – no VoIP server can answer the request.

Example of a SIP Message

An example of a SIP message, i.e., SIP INVITE message, is presented in Listing 4.3. User "bob" wants to establish a call session to user "alice". The empty line (at line number 14) is the separation between the SIP header and the SIP body. The body includes information about the media transmission via SDP. In the simplest form of SIP (without transport security mechanisms) all the messages are transmitted in plaintext and can be intercepted and interpreted easily by non-participating parties. It is also possible that a third-party modifies the messages and the callee does not recognize it.

```
INVITE sip:alice@192.168.10.21 SIP/2.0
Via: SIP/2.0/UDP 192.168.10.1:24445;rport;branch=z9hG4bK719462261
From: <sip:bob@192.168.10.21>;tag=638461942
To: <sip:alice@192.168.10.21>
Call-ID: 1024851935
CSeq: 20 INVITE
Contact: <sip:bob@192.168.10.21:24445>
Content-Type: application/sdp
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Max-Forwards: 70
User-Agent: LinphoneAndroid/2003 (eXosip2/3.6.0)
Subject: Phone call
Content-Length: 321

v=0
o=bob 1806 1527 IN IP4 192.168.10.1
s=Talk
c=IN IP4 192.168.10.1
b=AS:380
t=0 0
m=audio 7078 RTP/AVP 111 110 0 8 100 3 101
a=rtpmap:111 speex/16000
a=fmtp:111 vbr=on
a=rtpmap:110 speex/8000
a=fmtp:110 vbr=on
a=rtpmap:100 iLBC/8000
a=fmtp:100 mode=30
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11
```

Listing 4.3: Example of a SIP INVITE message

Session Description Protocol

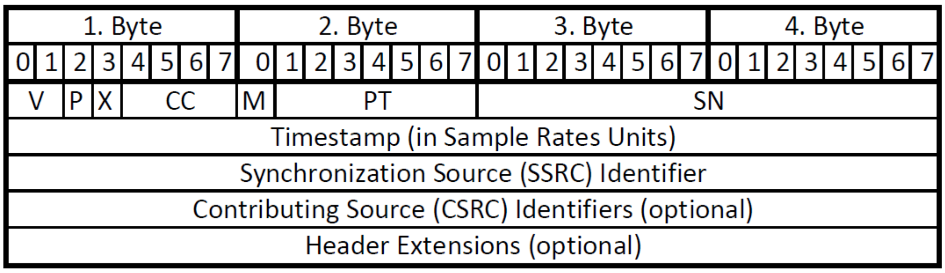
The SDP is defined by the IETF in RFC 4566 [64] and describes the streaming media initialization parameters between the communicating parties. For the analysis of VoIP attacks the SDP properties are important, because modifications can redirect the media stream to an attacker.

The SDP properties are represented in a text-based list of variables and their parameters and include the media type of transmission, the codec for the media type and the network port for RTP [148].

SDP is divided into three main sections, *session*, *timing*, and *media* descriptions. Each section can have a special selection of the following properties as described in RFC 4566 [64]:

- v= (current protocol version)
- o= (session originators name and session identifiers)
- s= (a textual session name)
- i= (a textual information about the session)
- u= (URI of further description)
- e= (email address of the initiator)
- p= (phone number of the initiator)
- c= (connection information)
- b= (proposed bandwidth limitations)
- t= (time the session is active)
- r= (specified the duration and intervals for any session repeats)
- z= (time zone adjustments)
- k= (simple mechanism for exchanging keys)
- a= (additional session attributes)

Usually, the additional session attributes are used to negotiate the mutually supported voice codecs between the endpoints. An example is seen in line 22 in Figure 4.3.



<table>
<tr><td colspan="8">1. Byte</td><td colspan="8">2. Byte</td><td colspan="8">3. Byte</td><td colspan="8">4. Byte</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">V</td><td>P</td><td>X</td><td colspan="4">CC</td><td>M</td><td colspan="7">PT</td><td colspan="16">SN</td></tr>
<tr><td colspan="32">Timestamp (in Sample Rates Units)</td></tr>
<tr><td colspan="32">Synchronization Source (SSRC) Identifier</td></tr>
<tr><td colspan="32">Contributing Source (CSRC) Identifiers (optional)</td></tr>
<tr><td colspan="32">Header Extensions (optional)</td></tr>
</table>

Figure 4.1: Structure of the RTP header as defined in RFC 3550 [135].

4.3 Real-Time Transport Protocol

The Audio-Video Transport Working Group of the IETF first developed and published RTP in 1996 as RFC 1889 [134], this was then replaced by RFC 3550 [135] in 2003. In RFC 3550 RTP was defined as a protocol for transmitting data with real-time characteristics (such as audio and video) over networks between the endpoints of a session [135]. Typically many media applications run RTP on top of UDP because, although they need near real-time delivery of packets, they can tolerate a certain amount of packet loss to achieve this goal.

RTP Header

Each RTP packet consists of the RTP header and the payload [135]. Figure 4.1 presents the structure of the RTP header as defined in RFC 3550 [135].

The elements of the RTP header are described by RFC 3550 [135] as:

- V – Version (2 bits): The version of RTP is described by this field.
- P – Padding (1 bit): Indicates if the packet contains one or more padding bits after the payload. Some encryption algorithms needed additional padding bits.
- X – Extension (1 bit): Indicates if the packet contains exactly one header extension.
- CC – CSRC count (4 bits): Defines the number of CSRC identifiers.
- M – Marker (1 bit): Packets can be marked for different interpretations (defined by profiles), such as frame boundaries.
- PT – Payload type (7 bits): Defines the type of the payload for further processing (defined by profiles).

- Sequence number (16 bits): This field is used to detect packet loss and to restore packet sequences by the receiver. For each RTP packet sent the sequence number is increased.
- Timestamp (32 bits): This field is used for synchronization and jitter calculation during a VoIP call.
- SSRC (32 bits): This field labels different streams and is used for multiplexing. The SSRC identifies the source of an RTP stream.
- CSRC (0 to 15 items, 32 bits each): This field identifies the sources of a combined stream (produced by an RTP mixer).
- Header Extensions: Indicates if a header extension (including the length) is appended.

The minimum header size is 12 bytes, while the list of Contributing Source (CSRC) identifiers is present only when inserted by a mixer. In RFC 3550 [135] a mixer is described as, "An intermediate system that receives RTP packets from one or more sources, possibly changes the data format, combines the packets in some manner and then forwards a new RTP packet. Since the timing among multiple input sources will not generally be synchronized, the mixer will make timing adjustments among the streams and generate its own timing for the combined stream. Thus, all data packets originating from a mixer will be identified as having the mixer as their synchronization source." The header size of RTP should be as small as possible to reduce the overhead, e.g., for encryption purposes. Cisco [26] describes some mechanisms to reduce the packet size with compression mechanisms. Another approach will be presented in Chapter 9 which compresses the overall packet to reduce the packet size.

RTP Payload

The payload of a RTP packet contains the data to be transported, for example audio samples or compressed video data [135]. In the case of VoIP the payload data consists of a voice codec.

Usually voice codecs are used to convert analog audio signals to digital encoded signals. The different codecs vary in the sound quality, the bandwidth required, the computational requirements, etc. The term codec is derived originally from *COder-DECoder*. [129]

The quality of voice in VoIP communications are influenced by the delay of IP packets, the delay differences jitter and packet loss. These QoS parameters are easy to measure, but do not describe the real voice quality of the communication, i.e., it is not clear if the communication was understandable for both parties. [47]

Codec	Codec Bit Rate in kbps	Effective Bit Rate in kbps	MOS
ITU G.711	64	87.2	4.1
ITU G.722	64	87.2	4.13
ITU G.726	32	55.2	3.85
ITU G.728	16	31.5	3.61
ITU G.729	8	31.2	3.92
GSM	13.2	29.2	3.7
iLBC	15	27.7	4

Table 4.1: Bit rate and Mean Opinion Score (MOS) values of main voice codecs [26, 47].

However, the quality of a call is subjective and is measured using Mean Opinion Score (MOS). Cisco [26] describes MOS as: "MOS is a system of grading the voice quality of telephone connections. With MOS, a wide range of listeners judge the quality of a voice sample on a scale of one (bad) to five (excellent). The scores are averaged to provide the MOS for the codec." Table 4.1 shows the main voice codecs and their MOS values. The International Telecommunication Union (ITU) G.711 codec has a similar voice quality to PSTN, for most of the other codecs from Table 4.1 the voice quality is worse than PSTN and should only be used if the bandwidth is limited, e.g., in mobile networks.

RTCP

The RTCP (also defined in RFC 3550 [135]) is used to control the RTP parameters between the communication endpoints and to get feedback on the quality of the data distribution [7]. RTCP is used to monitor data transmission, so that it can be scaled for large multicast networks without unnecessary overheads [135]. Usually implementations use the port number from RTP + 1 for RTCP messages.

4.4 Conclusion

This chapter provides details of the main protocols used in VoIP systems. Key elements of the signaling protocol SIP (e.g., SIP URI, different SIP messages and the SDP protocol) were described, as well as key elements of the media transmission protocol RTP (including an overview of possible voice codecs) and the control protocol RTCP. The details of the VoIP protocols are used in the following chapters for identification, analyzing and evaluating real-world VoIP attacks, as well as for the implementation of the protection mechanisms.

CHAPTER 5

Design and Implementation of a VoIP Honeynet for Capturing and Analyzing VoIP Attacks

Contents

5.1 Introduction

The identification of current threats and vulnerabilities of VoIP systems is an important step towards improving the security of the system. Therefore, a systematic approach to capturing data and identifying real-world attacks is needed to expand our knowledge beyond the known vulnerabilities discussed in Section 3.2.

This thesis uses a honeynet approach to capture real-world VoIP attacks and to identify threats and vulnerabilities. This chapter describes the design and implementation of the honeynet approach, which was fully implemented with open-source software including a PSTN gateway to enable calls to a PSTN system and a highly customizable analyzing engine for VoIP attacks.

5.2 Concept and Design of the VoIP Honeynet

The proposed solution is a complete infrastructure to identify threats and vulnerabilities of VoIP systems, gain details of VoIP attacks and trace the behavior of the attackers. It consists of a VoIP honeynet to collect data and an analyzing engine to analyze the captured attacks.

Concept of the VoIP Honeynet

The overall goal is to collect as much data about attacks on a VoIP infrastructure as possible, to determine threats and vulnerabilities of VoIP systems. Therefore, the data collection should be conducted on several layers, e.g., recording calls in order to detect fraud or collecting data packets to get information about attacks at the protocol-level. The intention was to only capture the data of explicit attacks (such as brute-force attacks or misuse of the system) and not the data from regular calls.

Figure 5.1 shows a sequence diagram of the proposed approach to capture attacks based on a honeynet. The diagram shows the attacker, the honeywall, the services of the honeywall (data capture, data control, notification) and the honeypots. An attacker attempts to attack a honeypot in the honeynet. The honeywall captures the packets, verifies if the packet is allowed to be forwarded to the honeypots (data control) and classifies the packet with a signature-based IDS. If the IDS recognizes an attack a notification will be send out. If the packets are allowed to be forwarded (based on IP firewall rules), they will be forwarded to the honeypot. Afterwards the response message from the honeypots will also be captured, controlled and a notification will be sent before the attacker receives the reply.

Each attack which cannot be classified by the IDS has to be analyzed manually in order to identify new or previously unknown attacks on the VoIP honeypots. If a new attack pattern is identified the IDS rules are adapted in order to recognize it automatically the next time.

The proposed approach is extensible to a VoIP honeynet with an uplink to a PSTN system, in order to capture fraudulent calls to PSTN systems too.

The honeypots provide VoIP systems which attract attackers. The concept considers that various different VoIP systems are better than only one kind, because this approach may attract different and probably more attackers.

The analyzing engine is responsible for preparing the captured data and carrying out customizable analysis on the data. Additional sources (e.g., external sources such as Domain Name

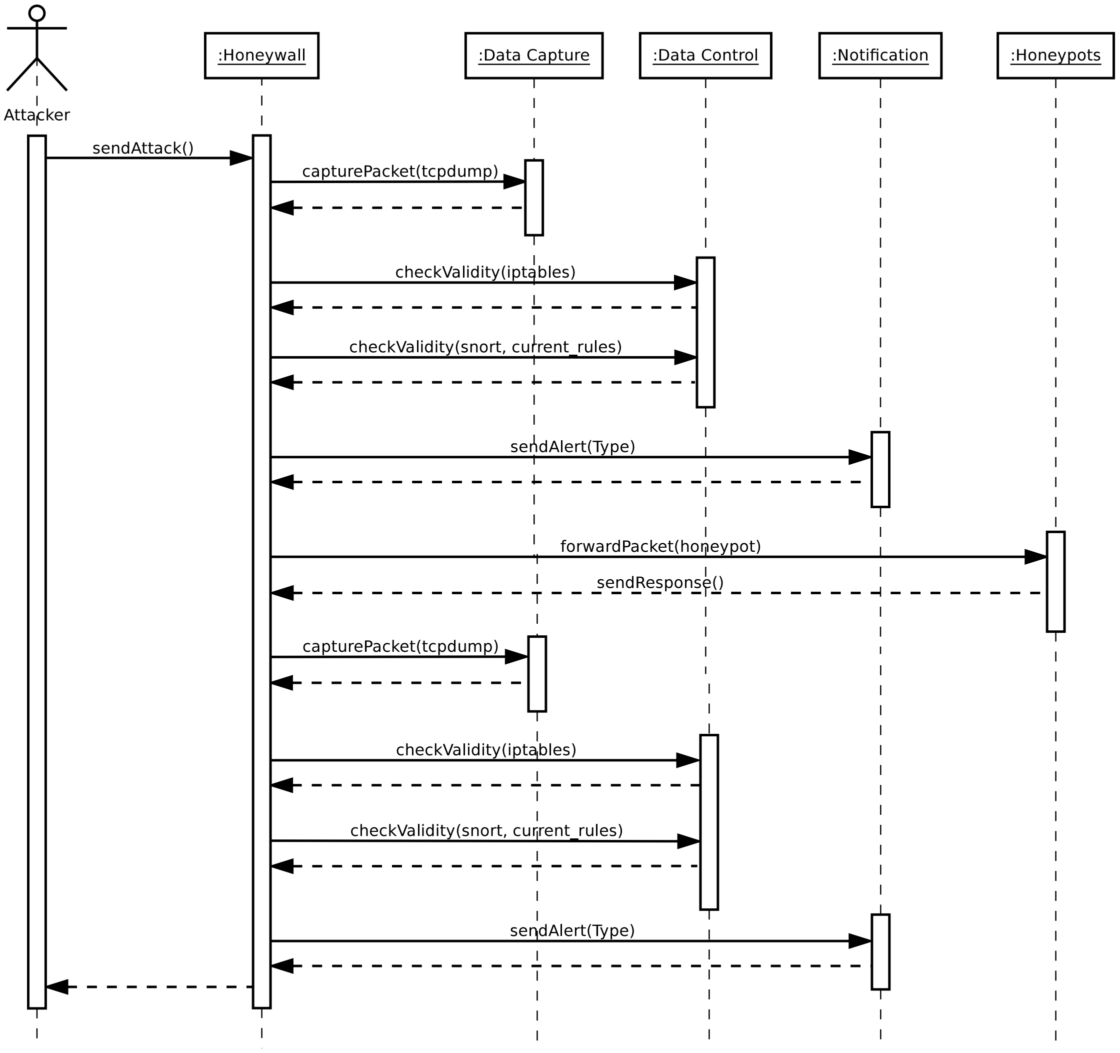


Figure 5.1: Sequence diagram of the proposed approach for capturing VoIP attacks by a honeynet.

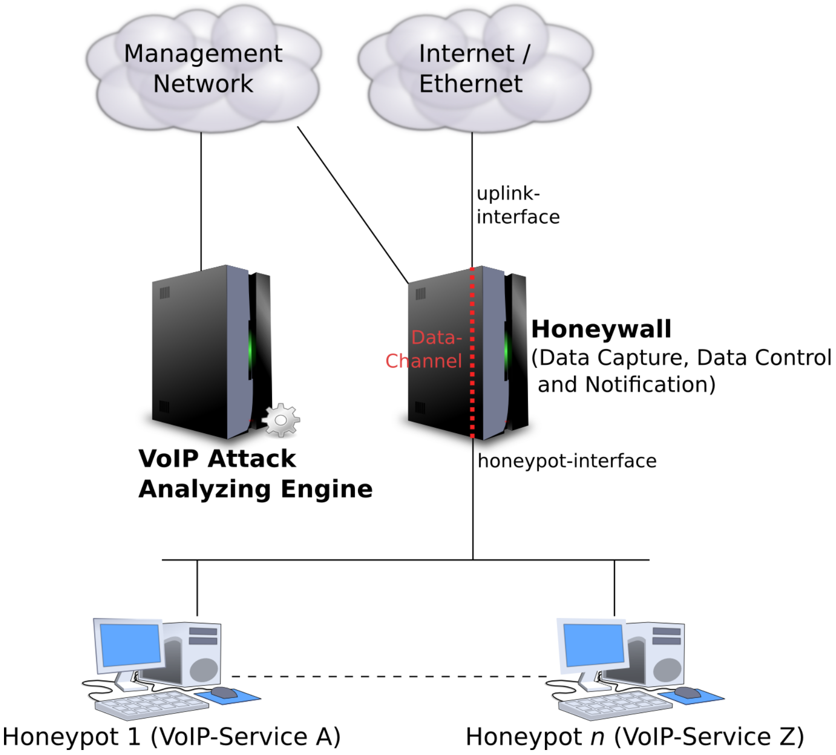

Management Network
Internet / Ethernet
uplink-interface
Data-Channel
Honeywall
(Data Capture, Data Control and Notification)
VoIP Attack Analyzing Engine
honeypot-interface
Honeypot 1 (VoIP-Service A)
Honeypot *n* (VoIP-Service Z)

Figure 5.2: Proposed architecture of a VoIP specific honeynet to identify threats and vulnerabilities.

System (DNS) or public phone books) can be used to carry out more detailed evaluations. To avoid delaying the response, the analyzing engine is a separate process and does not affect the regular packet flow. If the response times are too long, an attacker could identify the honeynet and stop the attack or make bluff attacks.

Design of a VoIP specific Honeynet

The basic structure of the proposed VoIP honeynet and the analyzing engine is shown in Figure 5.2. The honeywall is used as a centralized bidirectional data channel from the Internet to the honeypots. The design uses high-interaction honeypots in order to gain more details about the attacks. Each honeypot has a unique IP address and is accessible through the data channel. The VoIP specific honeynet has no connection to PSTN systems and can be used for identification of VoIP specific attacks.

The centralized data channel is a layer two bridge from the honeypots' interfaces and the uplink interface at the honeywall. All traffic to and from the honeypots goes through this bridge. The bridge is essential for the honeywall functionalities *data capture* and *data control*.

The data capture functionality of the honeywall uses PCAP[1] files to store the captured packets. To get additional information about VoIP attacks a Network Intrusion Detection System (NIDS) was deployed on the honeywall. The NIDS reads all incoming packets and tries to detect malicious traffic with rules or known signatures. The NIDS generates an alert message on detecting malicious traffic and stores it in a database. This database is later used for evaluation and analysis purposes.

For the data control functionality a firewall was deployed in order to protect the honeywall and the honeypots against unwanted attacks (i.e., attacks against services other than VoIP). All unwanted packets to and from the honeynet are blocked by this firewall and all wanted packets (e.g., VoIP packets) are also examined by a Network Intrusion Prevention System (NIPS). The NIPS listens on the layer two bridge and decides for each incoming packet the firewall accepted whether it is allowed to access the honeypots. It also decides for each outgoing packet whether it should be forwarded from the honeynet to the Internet. The NIPS uses the same rules and known signatures for detection of malicious traffic as the NIDS uses. However, both NIDS and NIPS alone only detect known attacks, unknown attacks may remain undetected. Therefore the analyzing engine uses both PCAP files and NIDS/NIPS information.

It was decided to use a dedicated management interface to control and monitor the honeywall and the data traffic to and from the honeypots. This has the advantage that the traffic on this interface is not captured for any analysis or evaluation, in contrast to the uplink interface.

Our honeywall supports monitoring attacks in real-time. Each alert from the NIDS can be followed in real-time and simple evaluations can be done very quickly. To carry out complex analysis or automatically generate reports about the captured data, a dedicated analyzing engine was developed. The analyzing engine is a separate component of the solution and fetches the captured data from the honeywall and imports the transformed data in to a database. Based on this data the analyzing engine caries out evaluations and analysis of the VoIP attacks. Templates for the queries and the output format must be defined to automatically create reports. The predefined queries and templates for automated analysis are stored in the analyzing engine.

The data from the honeywall are transmitted periodically via the management network to the analyzing engine, which prepares and stores the data and carries out queries to analyze the collected VoIP attacks. Without the analyzing engine the honeynet would only collect raw data,

[1] http://www.tcpdump.org/ or http://www.winpcap.org/

but this does not provide information about the VoIP attacks and the threats cannot be easily analyzed and evaluated. The analyzing engine is specially designed for VoIP to get as much information as possible about these attacks.

The honeynet solution was designed to notify the system operators, if an attack is recognized by the honeywall or an error occurs. The solution monitors traffic in real-time and sends notifications shortly after detection. The notifications are customizable and should be adapted to personal requirements, e.g., the number of notifications, which type of notification should be sent or at which time the notification should be delivered.

PSTN Extension of a VoIP Honeynet

In order to enable calls from a VoIP account to a number in the PSTN system, a special gateway is needed, as described in Chapter 2. The PSTN uplink can be an Integrated Services Digital Network (ISDN) modem, a data modem or a third party VoIP provider. We decided to use a third party VoIP provider with support for a prepaid solution to allow better cost control. To activate the uplink interface and to allow calls to PSTN endpoints, credit must first be bought from the provider. The integrated VoIP provider sends a notification if a customized limit is reached, in order to top up the credit in time.

Figure 5.3 shows an example of the steps for capturing calls to the PSTN system with our extended honeynet approach. First ①, the attacker tries to get access to a VoIP account on the honeypot for later use. In the next step ② a caller (who does not need to be the attacker) tries to call a phone number in the PSTN system. The honeywall captures all the VoIP packets, including the packets from the attacker, with the *data capture* functionality, sends an alert and forwards the packets to the honeypot ③. The honeypot itself has a network route to a PSTN gateway through the honeywall to support calls outside the infrastructure. The honeywall also controls the PSTN gateway to mitigate further risks (such as unintended calls) with the *data control* functionality. The honeywall allows configurations of predefined blacklists to block such calls, e.g., calls to law enforcement agencies or hospitals. In the fourth step ④ the honeypot forwards the call (signaling as well as media data) through the honeywall (data capture, data control and notification) to the callee in a PSTN system. This architecture should help to confirm the assumption that toll fraud attacks exist in the real-world and there is a business case for the attackers to earn money using them.

For the attackers the VoIP system of the honeypot seems to be a real VoIP gateway to make calls to PSTN endpoints. To carry out an analysis of the signaling and media data from toll fraud

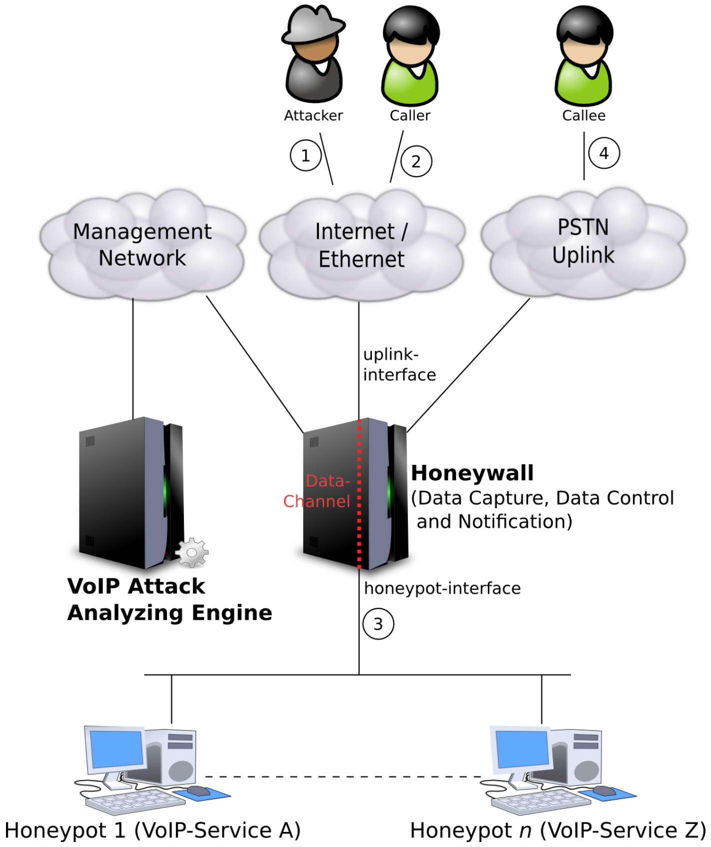

Attacker
Caller
Callee
1
2
4
Management Network
Internet / Ethernet
PSTN Uplink
uplink-interface
Data-Channel
Honeywall
(Data Capture, Data Control and Notification)
VoIP Attack Analyzing Engine
honeypot-interface
3
Honeypot 1 (VoIP-Service A)
Honeypot *n* (VoIP-Service Z)

Figure 5.3: Proposed architecture of a PSTN specific VoIP honeynet to identify threats and vulnerabilities.

attacks, the analyzing engine was extended with a feature to replay the media data, so that calls could be classified based on their content.

5.3 Implementation of the VoIP Honeynet

Based on the basic honeynet design of Spitzner [142] a honeynet was implemented for VoIP systems. Existing honeywalls like Honeywall CDROM [70] or honeypots like honeyd [111] did not fit our requirements, such as expandability of honeypots, easily extensible monitoring, controlling mechanisms and automatic and customizable analysis. Therefore, it was decided to implement a honeynet approach based on open-source software.

Data Capture

The data capture functionality is used to capture all traffic to and from the honeypots for later monitoring and analyzing activities. The VoIP honeynet uses tcpdump[2] to capture the traffic on the centralized data channel and uses the IDS Snort[3] to classify attacks and to get real-time alerts about recognized attacks. Snort is an open-source network intrusion detection and prevention system developed by Sourcefire which performs actions based on rule sets, e.g., writing log entries or sending alerts. The snort rules can be updated periodically by Sourcefire to provide alerts about newly discovered VoIP attacks. However, if there are mistakes in the configuration or a previously unknown attack is carried out, attackers can perform some unmonitored activities and false negatives can occur. To solve this problem a combination of tcpdump and Snort is used to detect malicious traffic.

Snort uses different rules to detect attacks and trigger alerts to notify the administrator of suspicious behavior in the honeynet. The VoIP honeynet uses the standard rules of Snort and customized rules to recognize patterns of attacks, especially for attacks against VoIP, e.g., SIP specific attacks. Snort can use different output channels out of the box, e.g., various databases or log files. Our VoIP honeynet uses a MySQL database[4] for storing alert messages.

Tcpdump captures the whole traffic on the layer two bridge in a PCAP file on the file system. These files are later used to evaluate and analyze VoIP attacks and are also the data basis for the reporting system. When the PCAP file reaches a predefined size of captured traffic a new PCAP file is created in order to ease the handling of the files. The size is an empirically measured average per day and can be adapted to the needs of the organization hosting the honeynet.

The management interface of the honeywall is monitored separately and each SSH connection is logged.

Data Control

To mitigate the risk of abuse to the honeynet the connections must be controlled. The data control functionality regulates which inbound and outbound data are allowed in the VoIP honeynet. If an attacker is able to get full root access to the honeypots, it could be possible to attack someone else in the internal or external network.

[2] http://www.tcpdump.org/

[3] http://www.snort.org/

[4] http://www.mysql.com/

Since we do not want attackers to attack external systems (e.g., banking systems), data control of the outgoing honeynet traffic is necessary. In our setup the VoIP honeynet provides only VoIP services for the external network, i.e., the Internet. Moreover, to avoid unnecessary traffic to the honeynet a control mechanism for incoming packets is used. It was decided to use iptables and Snort to handle data flow control. Snort analyzes all packets on the layer two bridge and decides for every packet if it will be forwarded to the honeypots or not. The VoIP honeynet is designed for SIP services and forwards only Internet Control Message Protocol (ICMP) packets, SIP packets on the standard port 5060/UDP and RTP packets to the honeypots. The second default SIP port on 5061 is not enabled, since we do not support TLS in our VoIP honeynet at this time.

Outgoing connections from the honeypots are restricted to the same conditions as incoming traffic for SIP and RTP. However, new ICMP requests are forbidden. The honeywall forwards all ICMP, TCP and UDP traffic to an iptables queue and the Snort NIPS functionality reads and analyzes packets from the queue and forwards VoIP traffic to the honeypots or to the system accessing the VoIP resources. All other traffic will be logged and dropped by the honeywall.

On the management interface only SSH connections to the honeynet are allowed. All other incoming connections will be dropped by iptables on the honeywall.

Honeynet Maintenance Capabilities

To perform updates or install new software on the components of the honeynet, a connection to an external network (such as the Internet) is required. This traffic should not be included in the evaluation of VoIP attacks. Therefore, a network route to the management interface of the physical host was created on the honeypots and the default route was removed. All traffic to the external network takes the new route and maintenance capabilities can be performed. This interface is only available via the physical host and is disabled for the honeypots in production.

The honeypots have no management interfaces of their own. To provide a connection from the honeypots to an external network, the physical host must provide routing functionality with IP masquerading to send and receive the packets. To switch back to production operation the default route on the honeypots must be enabled again. Figure 5.4 shows the default route (red line) for production purposes on the left side and for maintenance on the right side.

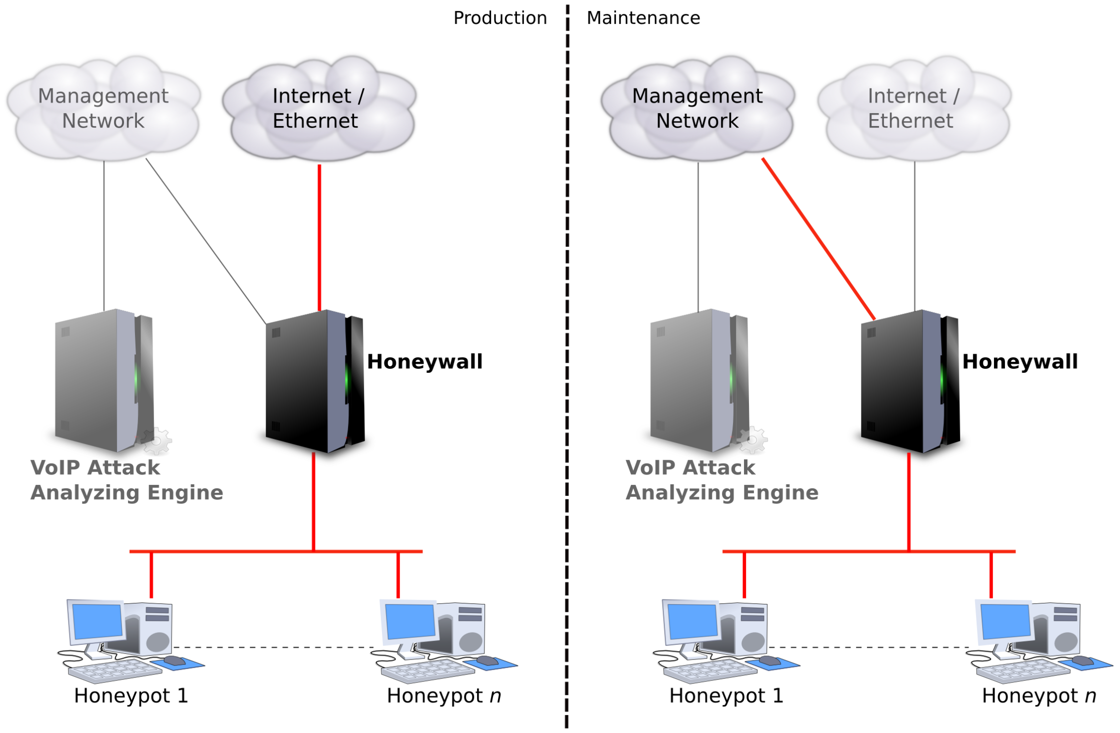


Figure 5.4: Network routes in the honeynet for production and maintenance purposes.

VoIP Honeypots

The proposed solution uses only high-interaction honeypots to get in-depth information about real-world VoIP attacks and to make it more difficult to identify the honeynet. Emulated services which are used in low-interaction honeypots are easily detectable by the attacker as Raffetseder *et al.* [114] described.

The introduced solution uses the Asterisk PBX[5] and a VoIP server based on SIP (called *sipListener*), which was specifically implemented for the proposed solution.

The in-house implemented VoIP server, which operates as a SIP proxy server and a UA, has the primary goal of easily recording VoIP communications (such as SPIT messages) and other data about calls. Therefore, all incoming SIP calls are accepted and logged to the file system. The VoIP server processes the RTP stream and transforms and stores it as an audio file (e.g., in WAV or mp3 format). To be able to save more calls each registration to the VoIP server is accepted and each call is answered by the proxy. This means that any username and password is accepted.

[5] http://www.asterisk.org/

All calls are answered by the VoIP server and anything the caller says is recorded. However, sipListener does not play any message in response.

Asterisk is a widely used VoIP system, which is easily connected to external systems and is therefore a very popular target for attackers. This means that an Asterisk honeypot can be used for identifying attacks against VoIP systems with or without a PSTN gateway. On the other hand, our in-house implemented VoIP server is completely unknown to attackers. The honeynet is not restricted to these two honeypot systems, other VoIP systems could be deployed as honeypots as well.

Notification System

To get just-in-time information from the honeynet we decided to enable a notification system. Information about the honeynet status (e.g., honeypots or specific services are offline) and notifications if an attack occurs in the honeynet are processed and delivered to predefined recipients. Which notifications are sent to the recipients can be configured, otherwise the system stores the information. In our VoIP honeynet an email was sent when an alert occurred.

The proposed VoIP honeynet has different components: the honeywall, the honeypots, the PSTN gateway and the VoIP attack analyzing engine. Each component will be monitored separately and an alert will be triggered if abnormal behavior is detected. The honeywall and honeypots periodically perform self-health checks of their systems, to verify if all services are running or the system is out of capacity. For example, if a service is not running as desired an automated process is started to restart the service and log entries describing the faulty state will be created, or if disk space is low a notification will be sent. The honeywall also monitors the management interface and an alert will be raised if someone connects to the honeywall via ssh or an invalid password authentication is detected.

With the web based tool Basic Analysis and Security Engine (BASE)[6] all attacks on the VoIP honeypots can be observed in real-time using the honeywall. Each alert from Snort creates a new database entry and can be seen immediately within BASE. Simple evaluations and reports are also possible with BASE.

The honeynet has an additional notification system, which presents the origin of an attack in a world map in real-time (see Figure 5.5). A country is marked as blue if the source of an attack originates from this country. The blue color is darker if more attacks occur from this country.

[6] http://base.professionallyevil.com/

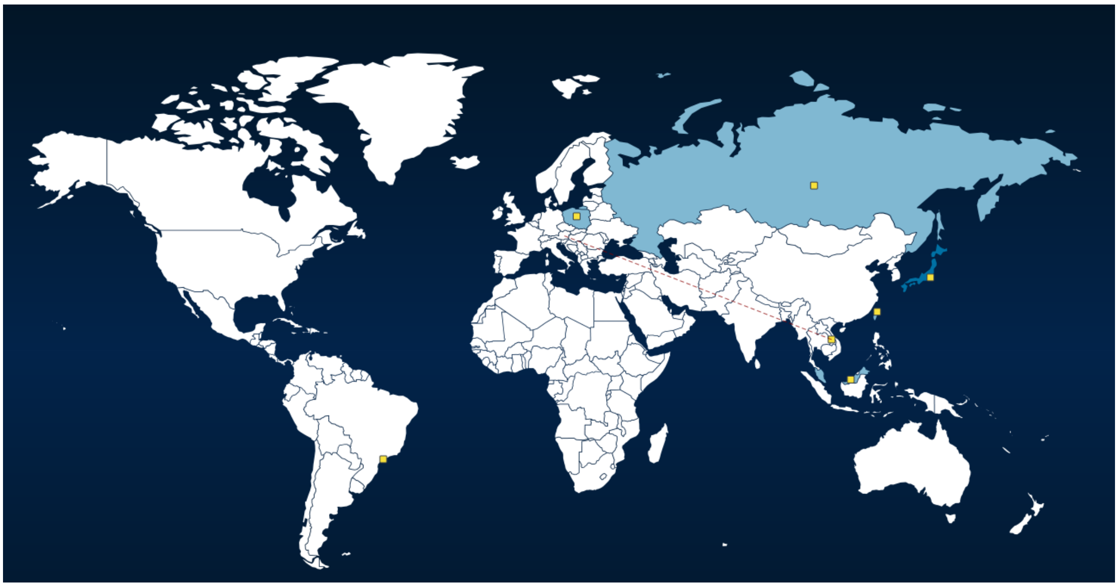


Figure 5.5: Real-time notification of collected attacks in the honeynet based on the ideas from the honeymap project [155].

The dashed line shows the combination of the attacker with the position of the target. This is helpful when multiple honeynets in different countries are in operation.

5.4 Analyzing and Evaluation of Captured VoIP Data

To easily analyze and evaluate the identified threats to VoIP systems, we decided to implement our own analyzing engine. Figure 5.6 presents the architecture of the VoIP attack analyzing engine. With this engine all data from the honeywall and the honeypots can be semi-automatically analyzed to gain information about the attacks and the attackers. In addition to the captured data from the honeynet (PCAP and Snort database), the analyzing engine uses third-party sources (e.g., the whois directory service, phone books, price tables of various VoIP services or UA databases) to gain even more information about the attacks and the attackers.

The PCAP data source for the VoIP attack analyzing engine includes the timestamp of the data collection, the properties of the IP stack (such as source and destination IP address), the SIP properties (such as SIP UA or SIP To address) and the RTP properties (such as media type) as well as the media stream. The Snort database contains information such as attack patterns or classification of attacks, but this information depends on the Snort rules and can be different in various deployments.

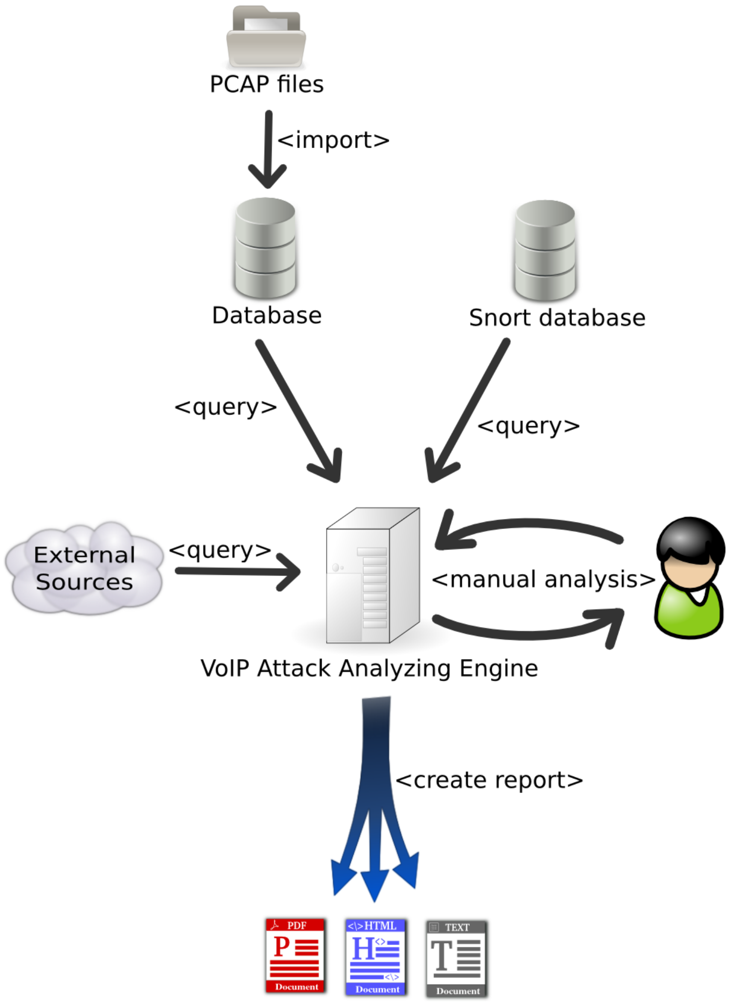


Figure 5.6: Architecture of the VoIP attack analyzing engine.

The analyzing engine supports manual media data analysis to identify similar noises, unidirectional communication and human conversations to determine the attacker's intentions, especially for fraudulent calls.

The proposed approach for analyzing and evaluating VoIP attacks can be easily extended with additional third-party sources or additional analyzing methods to gain more details about the captured attacks. Therefore, this approach is also suitable for future attacks against VoIP systems.

5.5 Conclusion

The proposed honeynet approach is an extensible and flexible solution for capturing and analyzing attacks against VoIP systems. Our approach can be used for VoIP specific systems as well as for VoIP systems with an uplink to PSTN systems, but also other VoIP extensions are possible. The ability to substitute the VoIP honeypots covers a wide range of different VoIP vulnerabilities, as different attacks for different VoIP systems exist.

The implemented VoIP attack analyzing engine uses various data sources to enable comprehensive analysis to gain details of VoIP attacks. Attacks against VoIP specific systems as well as VoIP/PSTN systems can be analyzed and evaluated. The option to create reports automatically may help to improve the security of VoIP systems continuously, because new attacks can be detected in each iteration.

The presented approach provides a comprehensive means of gaining details about attacks on real-world VoIP systems. The analysis and the evaluation of these threats and vulnerabilities will be covered in the next chapters.

Part II

Attack Analysis – Investigating VoIP Attacks

CHAPTER 6

Analysis and Evaluation of VoIP Specific Attacks

Contents

6.1 Introduction

The VoIP specific honeynet is necessary to understand more about VoIP attacks and to identify the attack surface of current VoIP systems. Using the VoIP specific honeynet, data about VoIP attacks were collected over a long period of time and based on this data a security analysis and evaluation were carried out.

The results from the analysis and evaluation of the captured data present valuable details about current VoIP attacks. These attacks represent real-world threats to VoIP systems and are also the basis for the design of secure VoIP communication.

In contrast to other work (e.g., by Werapun *et al.* [156] or Al-Allouni *et al.* [2]) which is often based on the analysis of available implementations, protocol specifications and known attacks, this approach determines if such attacks exist in the Internet and also provides an overview of the pattern of current real-world attacks. Different phases of VoIP attacks were captured, analyzed and evaluated. The VoIP attack analyzing engine also allowed the analysis of the behavior of the attacks in order to recognize patterns or signatures.

6.2 Basic Setup of a VoIP Specific Honeynet

The concept of the VoIP honeynet, as described in Chapter 5, can be used to obtain information about all kinds of VoIP attacks. The attacks do not have to be classified, unclassified attacks are also recorded and can be analyzed later with the analyzing engine.

By running multiple VoIP honeynets in different networks for over several months, a large number of attacks on VoIP systems could be collected.

Configuration and Operation in Separated Networks

Two different VoIP specific honeynets were deployed in separate networks in order to gain information about the correlation between attacks in different network ranges.

The first location used infrastructure at the Vienna University of Technology and, therefore, has a basic firewall, i.e., ICMP packets were not routed to the honeypots. The second location was in a public network in Austria without network-based protection mechanisms.

The honeynet at Vienna University of Technology operated eight honeypots (see Figure 6.1) over a duration of 4 years and 5 months. The honeynet in a public network in Austria operated two honeypots (see Figure 6.2) over a duration of 3 years and 9 months. The honeynet at Vienna University of Technology used one physical host for the honeywall and one physical host for the virtualized honeypots. The honeynet in the public infrastructure in Austria used only one physical host for the honeywall and the honeypots. In both solutions a large number of VoIP attacks were detected, analyzed, correlated and evaluated with the VoIP attack analyzing engine.

The honeypots in both honeynets used the Asterisk PBX and the in-house implemented VoIP server (*sipListener*) as VoIP services. The biggest impact for the analysis of the honeynet at Vienna University of Technology is that ICMP is blocked by a global firewall. ICMP is usually used to check the availability of hosts by sending ping requests. Attackers often use these in the first step, as Skoudis and Liston [138] described. Many attacks on the Vienna University of Technology honeynet were not carried out, since ICMP was blocked and the attackers did not continue with the attack.

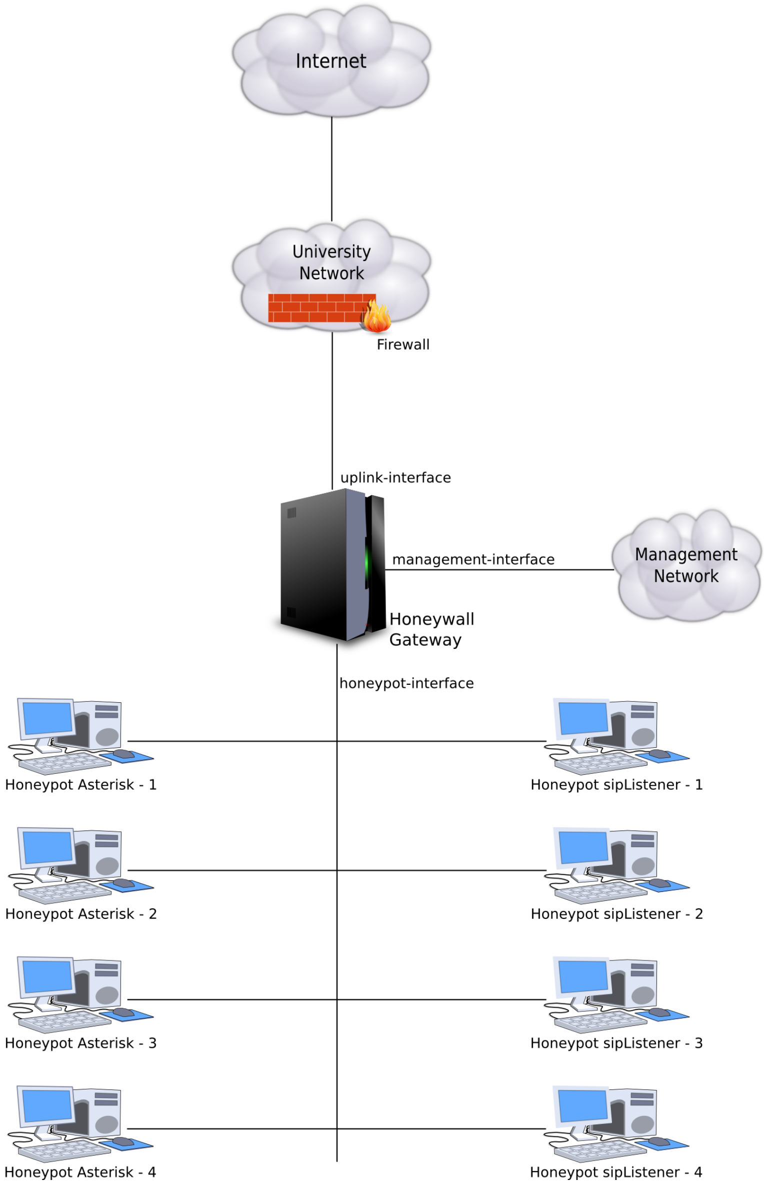


Figure 6.1: VoIP specific honeynet architecture at the Vienna University of Technology.

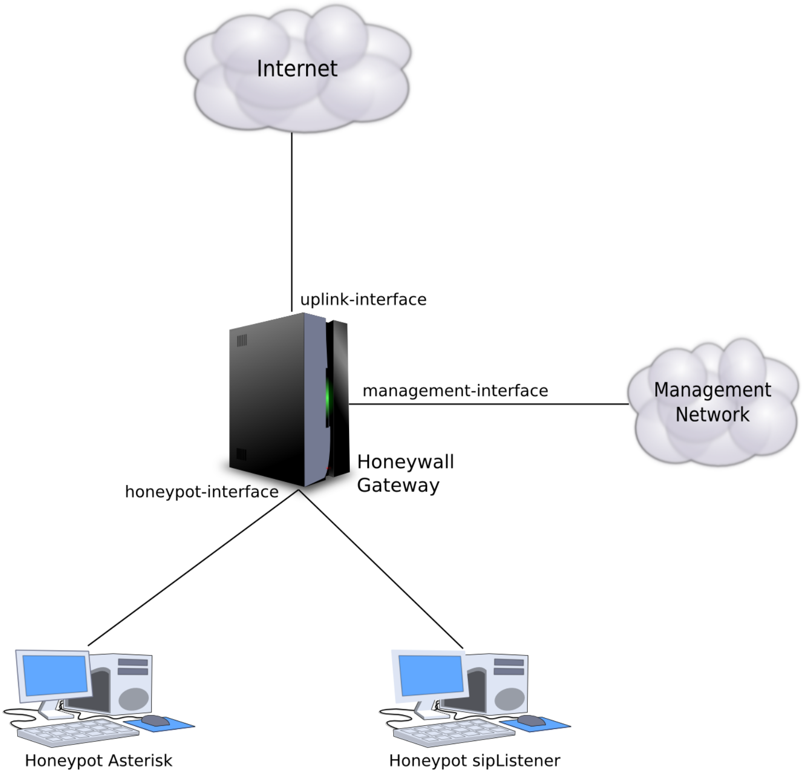


Figure 6.2: VoIP specific honeynet architecture in a public network in Austria.

Both VoIP honeynets used similar architectures, VoIP services and configurations to make correlations and analysis on both systems easier. They differed in the locations and the IP addresses of the honeynets. So as not to arouse suspicion, the IP addresses were from different ranges. Neither of the honeynets had a connection to the PSTN. Calls to PSTN were recorded but not forwarded and no voice response was sent.

Collected Data

The VoIP honeynet at the Vienna University of Technology has been in operation since August 2009. For the analysis in this work, the honeynet collected data up to and including December 2013. In this period the honeywall recorded 56,775,275 SIP messages.

The VoIP honeynet in a public network in Austria has been in operation since April 2010. This honeynet collected data as long as the honeynet at Vienna University of Technology and recorded 66,487,081 SIP messages.

Both honeynets recorded a large number of VoIP attacks and only a small number of other IP-based attacks, because only VoIP services were available. All other services (such as SSH, File Transfer Protocol (FTP) or HTTP) were captured but then blocked by the honeywall. No IP addresses or SIP URIs were published for either of the honeynets to see if an unpublished system would be attacked or not. Therefore, the attackers did not have any information that SIP services were running on the IP addresses used without scanning the ports of the host running the VoIP services.

6.3 The Most Disruptive Security Attacks on VoIP Specific Systems

The collected data have to be analyzed to get meaningful information about VoIP attacks and to use these information to implement secure VoIP systems. Therefore, as the first step the attacks collected in the honeynets were classified into different attack groups (based on the VoIPSA classification in Chapter 2). These attack groups are the basis for further analysis. A detailed discussion which focuses on conspicuous attacks is presented in the last section.

Analysis of Classified Attacks

In both VoIP honeynets various numbers of classified attacks were detected. Table 6.1 shows the detailed result of the analysis. Honeynet A represents the honeynet at the Vienna University of Technology and Honeynet B the honeynet in the public network.

We observed that SIP REGISTER messages, with the goal of identity theft, were the most common messages in both honeynets. This kind of attack includes a large number of dictionary and brute-force attacks. In most cases the attacker tried to find a valid username by brute-force or some default user of the VoIP system. This is possible because VoIP systems disclose information about whether a user is valid or not. Afterwards the password is attempted with random combinations or dictionary attacks.

The second most common messages in both honeynets were malformed message attacks. They occurred more often in the honeynet in the public network than in the Vienna University of Technology honeynet. In most cases the protocol specification had been violated, e.g., necessary fields were missing or illegal characters were used. This attack could cause DoS or disclose information about the VoIP system.

Type of Attack	Honeynet A	Honeynet B
Identity Theft	98.88%	97.59%
Malformed Messages	0.59%	1.72%
Fraudulent Calls	0.53%	0.69%

Table 6.1: Relative number of observed attacks on SIP systems in different VoIP specific honeynets.

The third most common messages in both honeynets were SIP INVITE messages, with the goal of making fraudulent calls. In most cases the attackers tried to call PSTN numbers. SIP calls to other VoIP users were in the minority.

Although three main attacks were identified, identity theft was by far the most common attack in both honeynets and presents the greatest threat.

Details of VoIP Attacks

Based on the captured data, various statistical analyses were carried out to get more details about the VoIP attacks. The intensity or the frequency of the attacks captured by the honeynet represents the likelihood of the occurrence in real-world VoIP systems, which highlights the main risks to VoIP systems (as used in Chapter 8).

The analysis of the number of SIP INVITE messages shows various peaks in both honeynets. Figure 6.3 shows the number of SIP INVITE messages on both honeynets over time. The honeynet at the Vienna University of Technology had a peak in the second quarter of 2011. The honeynet in the public network in Austria had a peak in the second quarter of 2013. Most captured calls tried to call an external PSTN number, but the call was not established successfully. Only in few cases did an attacker try to call another VoIP user.

Attackers gain the most profit from a successful attack if they can use a valid account in a VoIP system to make international calls for free. One possible intention of the attacker is to cause costs for the victim and/or increase their own profit by calling their own chargeable (premium rate) numbers. Another possible intention is a DDoS attack on an external SIP entity and the honeynet is only a part of a larger attack. For example one number was collected several times, which was identified as the phone number of the Foods Standards Agency[1] in Great Britain.

[1] http://www.food.gov.uk/

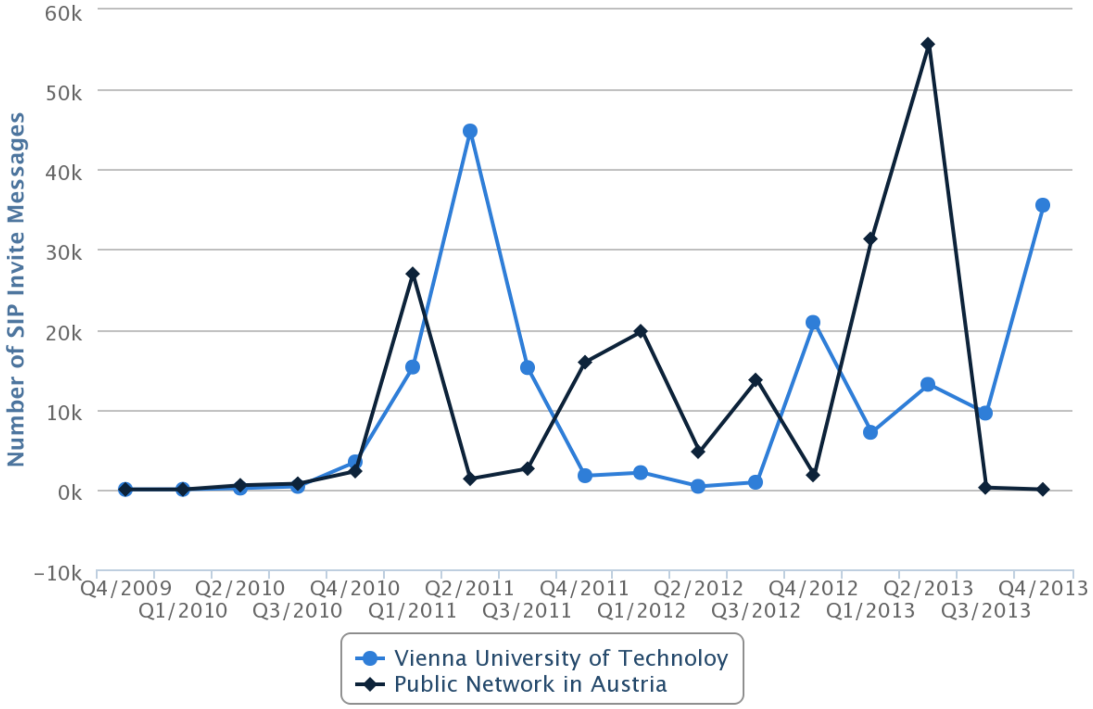


Figure 6.3: Number of SIP INVITE messages in the VoIP specific honeynets.

Another interesting point is the large number of unique IP addresses in both honeynets. It was expected that many IP addresses would be the same in both honeynets. However, this was not the case, as only one-fifth of the recorded IP addresses overlapped with each other in both honeynets. All other IP addresses are different. This does not mean that the attackers were different, as they could be the same person using different IP addresses and the intention of the attacks might be the same.

Although the IP addresses were different, the patterns of the attacks were very similar. In the first months the attackers tried to get an account with dictionary or brute-force attacks and after a successful attempt, the availability of the SIP system was checked with SIP OPTION messages.

Some attackers (i.e., IP addresses) were detected repeatedly over several months. A detailed analysis of the VoIP attacks shows that not all possible SIP attacks occur with the same frequency over the months in which an attacker was repeatedly recognized. The attackers perform different VoIP attacks in the first months than at a later point in time, e.g., malformed messages or brute-force attacks. In the public honeynet the attacks that occurred most in the first months were attacks to get a valid registration or malformed message attacks (named as *Others*), as shown

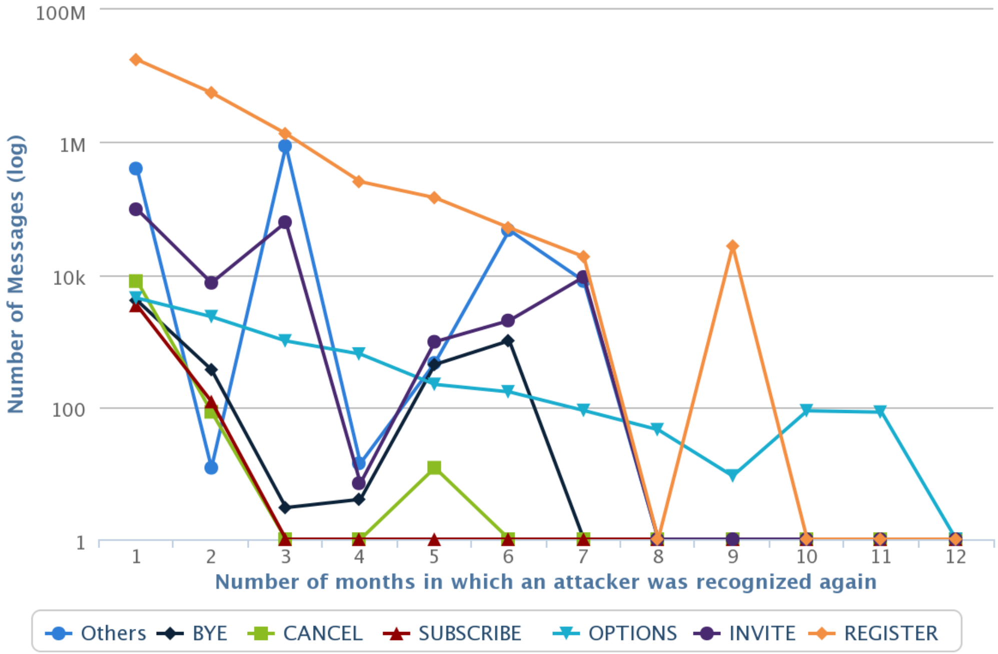


Figure 6.4: Number of SIP messages and their course in the public VoIP specific honeynet grouped by SIP methods and the number of months in which an attacker was recognized again.

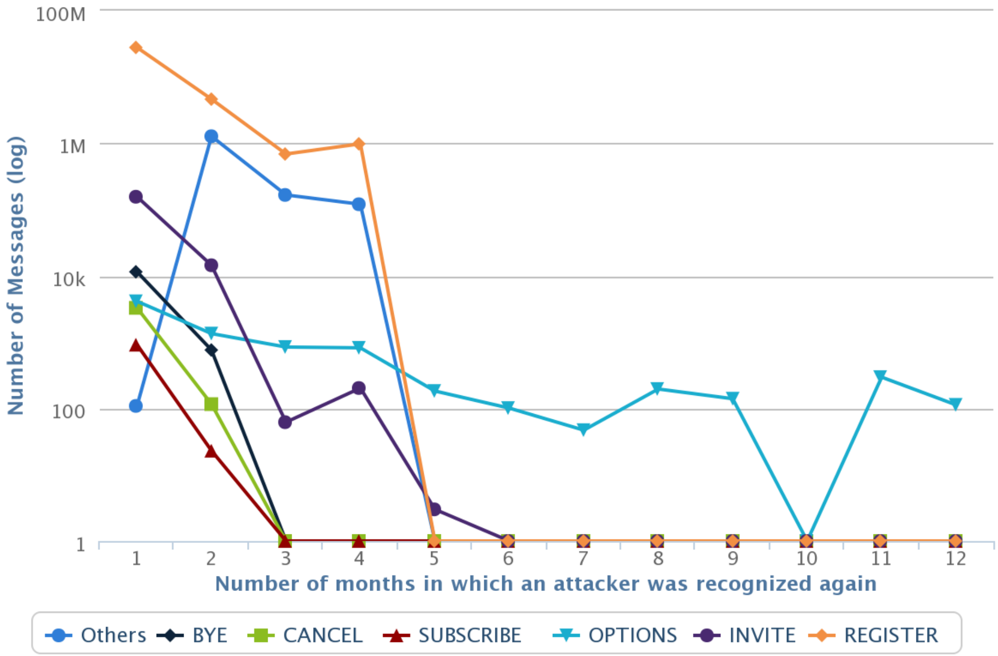


Figure 6.5: Number of SIP messages and their course in the VoIP specific honeynet at the Vienna University of Technology grouped by SIP methods and the number of months in which an attacker was recognized again.

in Figure 6.4. In the honeynet at the Vienna University of Technology register attacks and fraudulent calls were the most common ones in the first months, as shown in Figure 6.5. For a clearer view the vertical axis is shown in a logarithmic scale in both figures. After the first month fraudulent calls (INVITE messages) and malformed messages (named as *Others*) changed place.

In both honeynets after a few months the number of different attacks decreased while the SIP OPTION messages remained over a long period (i.e., 29 months in the public honeynet and 42 months in the honeynet at the Vienna University of Technology). These messages are often used to verify the validity of a stolen SIP account or to verify if a VoIP system is still running.

Figure 6.6 shows the number of SIP messages grouped by SIP methods in the honeynet at Vienna University of Technology. Most messages are by far SIP REGISTER messages. Figure 6.7 shows the number of SIP messages without SIP REGISTER messages grouped by SIP methods in the honeynet at Vienna University of Technology. The second most common SIP message types were malformed messages (named as *Others*) and SIP INVITE messages.

Figure 6.8 presents a similar result for the honeynet in a public network in Austria. SIP REGISTER messages are by far the most frequent messages. Figure 6.9 shows the number of SIP messages without SIP REGISTER messages grouped by SIP methods in the public network honeynet. Malformed messages (named *Others*) are the second most common SIP message type.

Analyzing Attacker Behavior

To get more information about the attackers and their behavior further analyses were carried out. The analysis of the locations of the attackers represents the distribution of the attack origins. Each identified source IP address of VoIP attacks was mapped with GeoIP [52] to get the country where the system was used by the attacker. Table 6.2 shows an extract of interesting countries ordered alphabetically. The honeynet at Vienna University of Technology is represented as Honeynet A and the honeynet at a public network in Austria is represented as Honeynet B. In the honeynet at the Vienna University of Technology IP addresses associated with Germany sent the most SIP messages to the honeynet. In the honeynet in a public network in Austria IP addresses associated with the United States sent the most SIP messages to the honeynet.

Another interesting result was obtained by the analysis of the number of unique source IP addresses in the honeynets. Figure 6.10 shows a huge peak of unique IP addresses during the fourth quarter of 2010 in the Vienna University of Technology honeynet. However, the honeynet in the public network has no peak during the same time. This shows that the attackers of both hon-

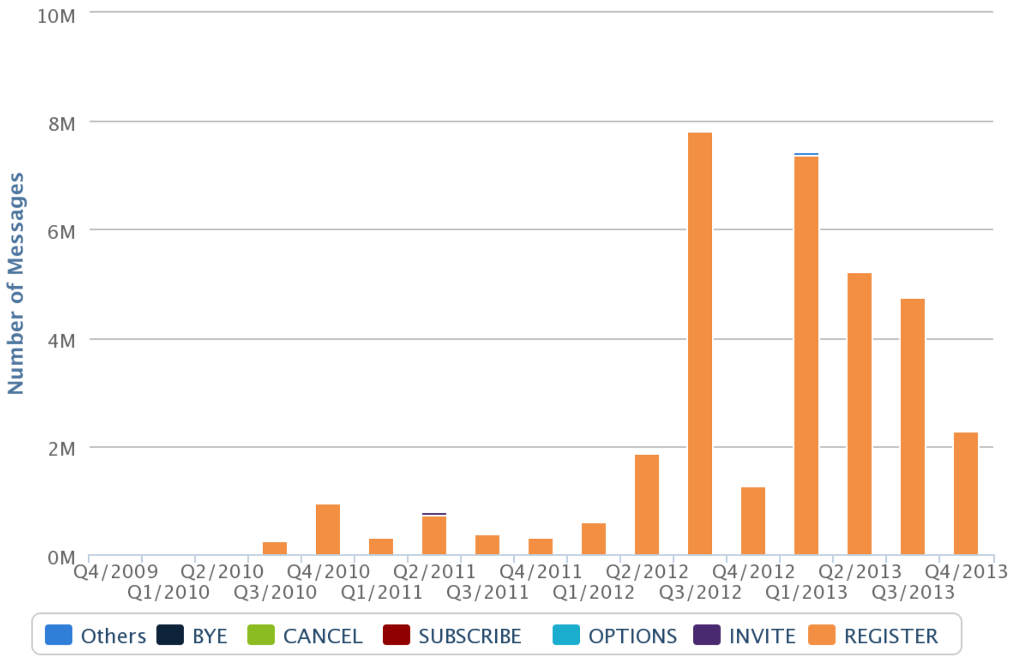


Figure 6.6: Number of SIP messages in the VoIP specific honeynet at the Vienna University of Technology grouped by SIP methods over time.

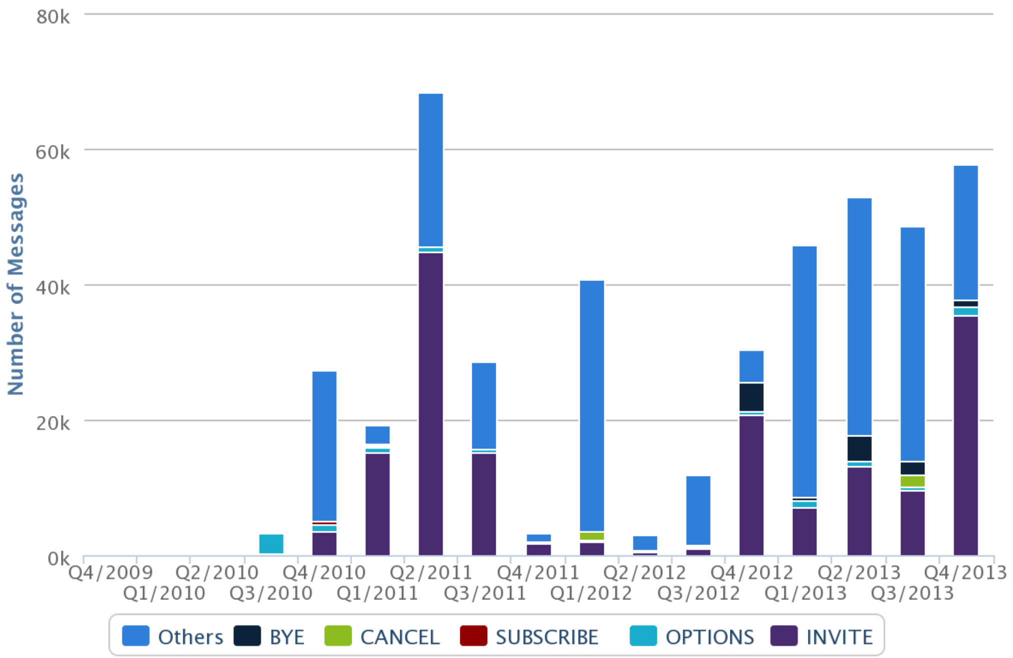


Figure 6.7: Number of SIP messages (without SIP REGISTER messages) in the VoIP specific honeynet at the Vienna University of Technology grouped by SIP methods over time.

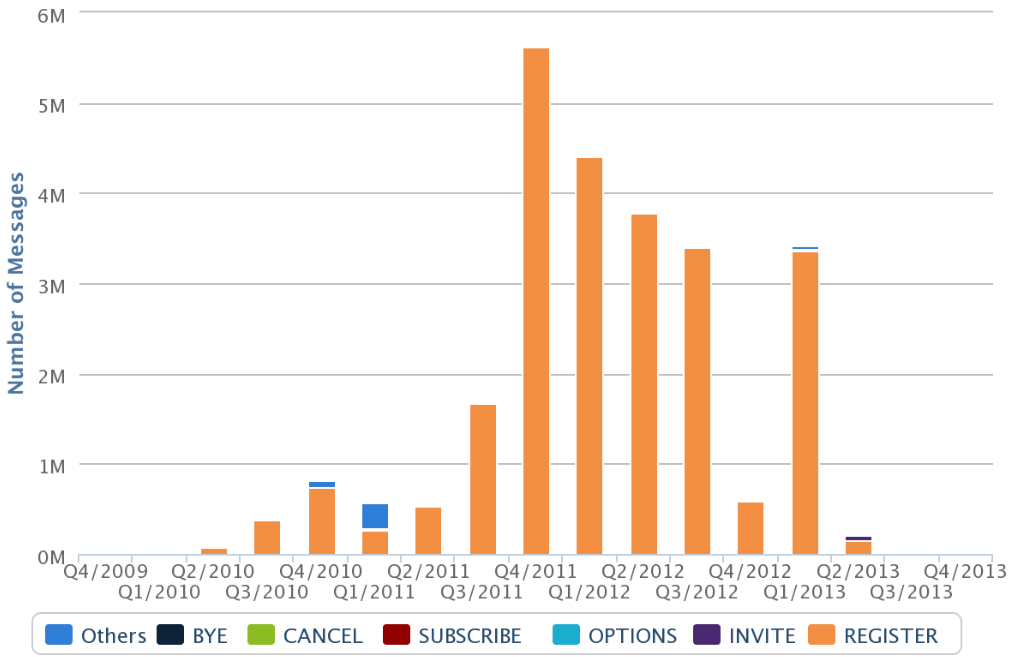


Figure 6.8: Number of SIP messages in the VoIP specific honeynet at a public network in Austria grouped by SIP methods over time.

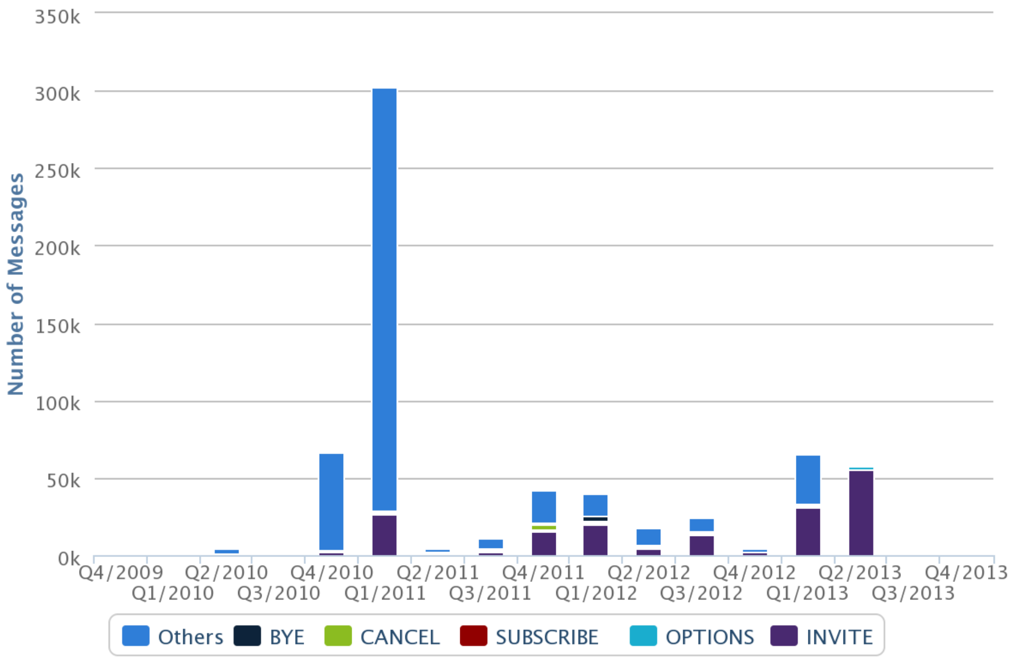


Figure 6.9: Number of SIP messages (without SIP REGISTER messages) in the VoIP specific honeynet at a public network in Austria grouped by SIP methods over time.

Country	Honeynet A	Honeynet B
Australia	16,649	32,080
Austria	20,874	659
Canada	1,466,475	752,048
China	1,049,535	3,833,890
France	211,508	1,776,913
Germany	20,935,839	9,726,482
India	174,439	141,017
Japan	101,125	101,006
South Africa	229,357	20,175
Taiwan	92,868	58,460
United Kingdom	589,572	489,514
United States	10,779,925	12,042,734

Table 6.2: Number of unique source IP addresses grouped by countries in both VoIP specific honeynets.

eynets are most likely not the same, because in most cases they also have different IP addresses. This indicates that the honeynet was attacked by a botnet or the attackers exchanged information about the targets.

The analysis shows different attackers in the honeynet at Vienna University of Technology and the public network honeynet. The SANS [125] systems also recognized some peaks of unique source IP addresses in VoIP attacks.

The analysis of used UAs (see Table 6.3) of the clients accessing the honeypots shows a large number of known tools that are used by attackers of SIP systems, e.g., `friendly-scanner` or `sundayddr`. The large number of messages from `friendly-scanner` can be explained by the high number of identity theft attacks, some known SIP scanners use this UA entry. The UA `sundayddr` rarely occurred and is included in the category *others*. The high number of *empty* UAs seems to indicate that many well organized attackers performed attacks, and not only attackers who use standard tools (e.g., scanning tools) without modifying them.

Table 6.4 shows the number of days a source IP address was recognized by the honeynet. The majority of hosts access the honeynets for approximately one month, afterwards the IP address is not seen again or the attacker's IP address changes.

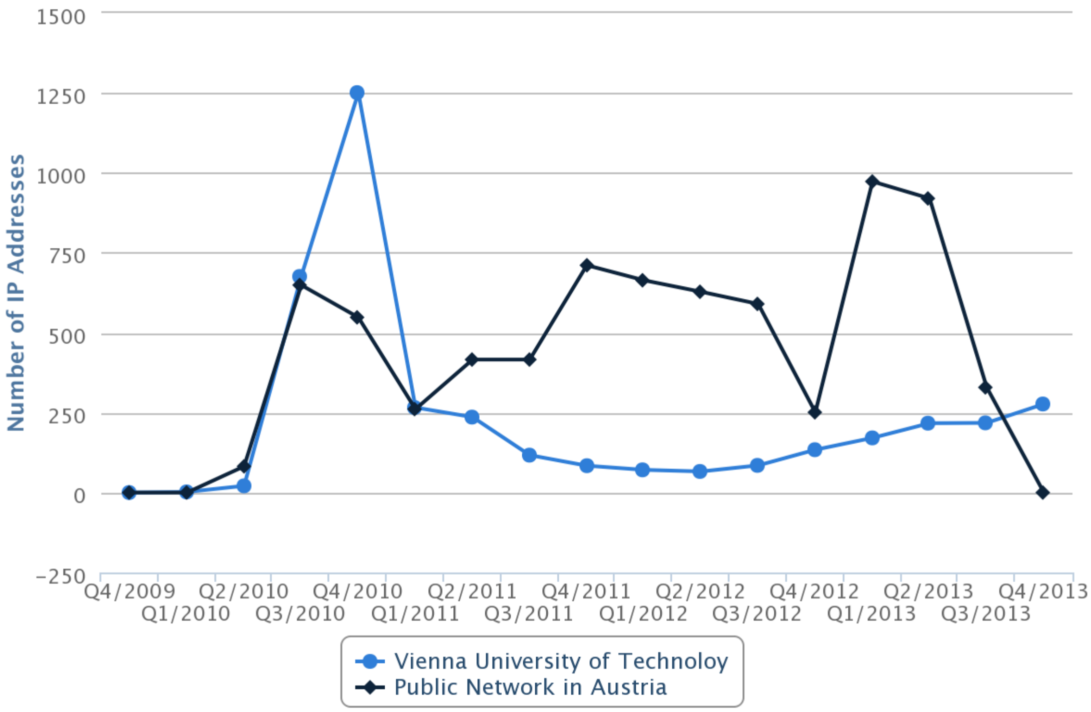


Figure 6.10: Number of unique source IP addresses in the VoIP specific honeynets.

SIP User Agent	Honeynet A	SIP User Agent	Honeynet B
friendly-scanner	97.80%	friendly-scanner	97.24%
empty	0.94%	*empty*	1.88%
VaxSIPUserAgent/3.1	0.68%	nike	0.33%
others	0.28%	*others*	0.55%

Table 6.3: Relative number of different SIP user agents in messages to the VoIP specific honeynets.

Type	Honeynet A	Honeynet B
Median	30	30
Average	39	38

Table 6.4: Number of days a source IP address was recognized by the VoIP specific honeynets.

Validation of the Collected Data

To validate if the honeynets collected representative data over a long period of time, various analyses were carried out, e.g., attackers access the system in an unexpected way.

Since the first attack occurred new IP addresses were identified in both honeynets every month. After the first VoIP attack, the number of attacks increased. It looks like organized attackers because, after the honeynets were recognized, the information was presumably distributed to other attackers.

Table 6.5 shows the number of recurring and unknown IP addresses per month for an evaluation period from January 2012 to December 2013. After an in-depth analysis of the collected IP addresses, we identified that the VoIP attacks were not arbitrary, but organized. The attacks were performed by recurring and unknown IP addresses in a similar way to avoid a complete blacklisting of known IP addresses and thus blocking access to the VoIP systems.

Figure 6.11 shows the number of VoIP messages in the VoIP specific honeynets. In the third quarter of 2011 the honeynet in the public network in Austria had a huge peak of collected VoIP data. After that various up- and downturns can be seen in the results. The honeynet at Vienna University of Technology had four times more honeypots as the honeynet in the public network. However, the amount of data collected was not four times more. This discrepancy is due to the blocking of ICMP in the network of Vienna University of Technology (as described Chapter 5).

The distribution of the access to the honeypots in the honeynet at Vienna University of Technology is presented in Figure 6.12. Figure 6.13 shows the distribution of access to the honeypots in the honeynet at a public network in Austria. Both analysis show a slight imbalance in the distribution of the honeypots. In both honeynets the Asterisk honeypots were accessed more often than the sipListener honeypots, which indicates that the attackers choose their victims selectively (e.g., based on the user agent string).

6.4 Conclusion

The analysis of the data collected from both VoIP specific honeynets gave details about VoIP attacks and the attackers' behavior over a longer period of time. Both honeynets were attacked many times from different source countries. In the honeynet at the Vienna University of Technology most attacks were associated with an IP address from Germany. In the honeynet at a public network in Austria most attacks had the United States as the country of origin. The classification of the attacks showed that the majority of attacks in real-world VoIP systems were identity theft

	Honeynet A			Honeynet B		
Date	Sum	Recurring	Unknown	Sum	Recurring	Unknown
2012-01	34	11	23	207	38	169
2012-02	21	9	12	202	41	161
2012-03	16	9	7	255	42	213
2012-04	22	6	16	234	41	193
2012-05	18	5	13	199	39	160
2012-06	26	6	20	195	39	156
2012-07	28	8	20	139	21	118
2012-08	28	4	24	218	33	185
2012-09	29	5	24	233	51	182
2012-10	40	7	33	251	36	215
2012-11	44	9	35	-	-	-
2012-12	50	8	42	-	-	-
2013-01	61	11	50	-	-	-
2013-02	49	11	38	419	15	404
2013-03	61	9	52	554	86	468
2013-04	73	16	57	362	87	275
2013-05	79	12	67	297	78	219
2013-06	65	7	58	261	93	168
2013-07	72	13	59	269	124	145
2013-08	60	10	50	300	215	85
2013-09	86	13	73	-	-	-
2013-10	61	14	47	-	-	-
2013-11	122	13	109	-	-	-
2013-12	94	21	73	-	-	-

Table 6.5: Recurring and unknown IP addresses in the VoIP specific honeynets from January 2012 to December 2013.

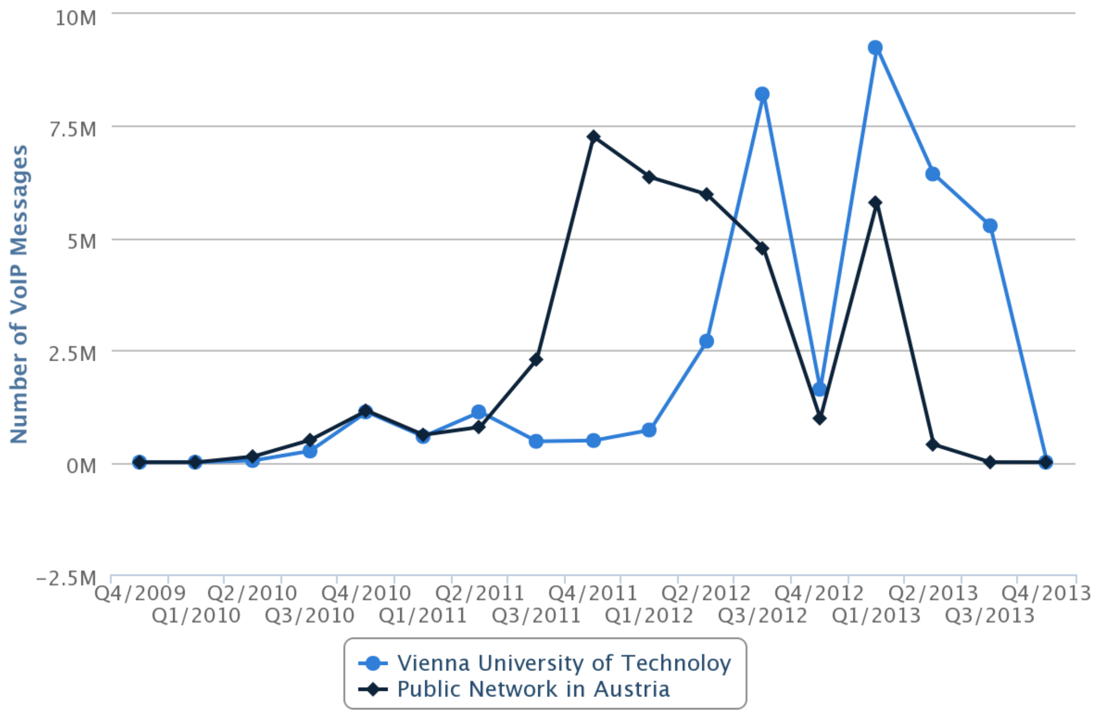


Figure 6.11: Number of VoIP messages in both VoIP specific honeynets.

attacks. An in-depth analysis of the messages showed that the most attacks were carried out by a standard tool called `friendly-scanner`.

To secure VoIP systems using simple security mechanisms is not enough. For example, simple black list protection will not work efficiently, because of the large number of different IP addresses used for the attacks. Moreover, attackers constantly change their behavior and their attack patterns.

Identity theft, unintended SIP messages and malformed messages are the main problems for the security of VoIP systems. The operation of the honeynet has shown that interception of the VoIP messages (signaling as well as media transmission) might be a problem if rogue proxies are involved. These security problems have to be considered in the design of a secure VoIP solution.

The insights into VoIP specific attacks gained through the analysis are the basis for securing VoIP systems. The next chapter will examine the PSTN based VoIP attacks further expand our knowledge of VoIP attacks.

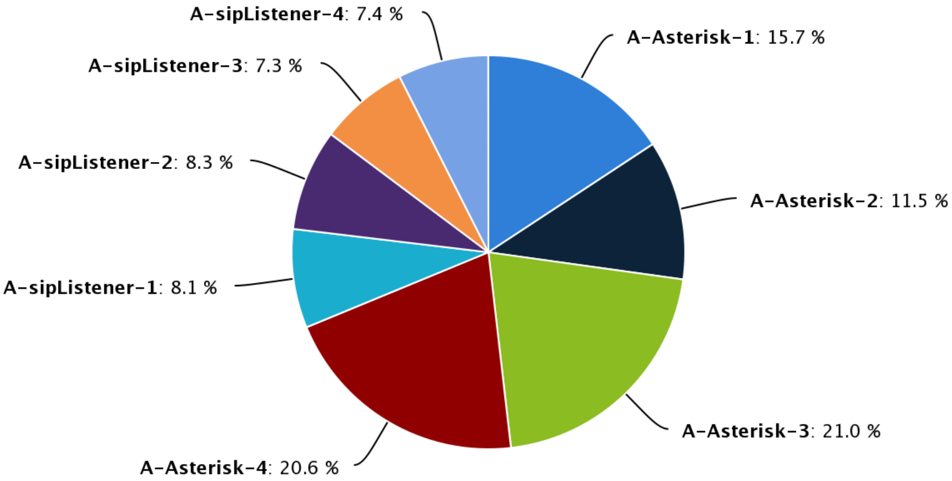


Figure 6.12: Distribution of access to the honeypots in the VoIP specific honeynet at Vienna University of Technology.

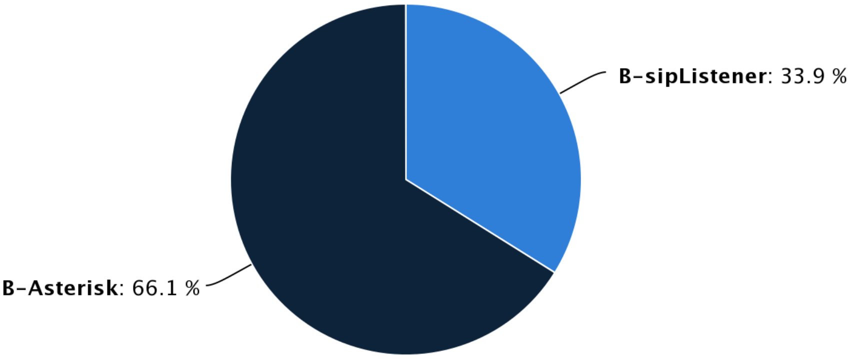


Figure 6.13: Access to the honeypots in the VoIP specific honeynet at a public network in Austria.

CHAPTER 7

Analysis and Evaluation of PSTN Specific VoIP Attacks

Contents

7.1 Introduction

The analysis and evaluation of threats and vulnerabilities of VoIP systems in Chapter 6 showed us that VoIP security is a critical issue because a lot of attacks against VoIP specific systems took place. To gain additional information about threats, especially toll fraud attacks or fraudulent calls, an extended honeynet with an uplink to PSTN systems was implemented and operated.

Data about VoIP attacks (in particular fraudulent calls to PSTN systems) was captured with this extended honeynet approach. This data was semi-automatically analyzed and evaluated to find out the major threats to VoIP systems which have a PSTN connection.

The analysis and evaluation of toll fraud attacks give us insights into the behavior of attackers and also provide key information for the design of secure VoIP communication.

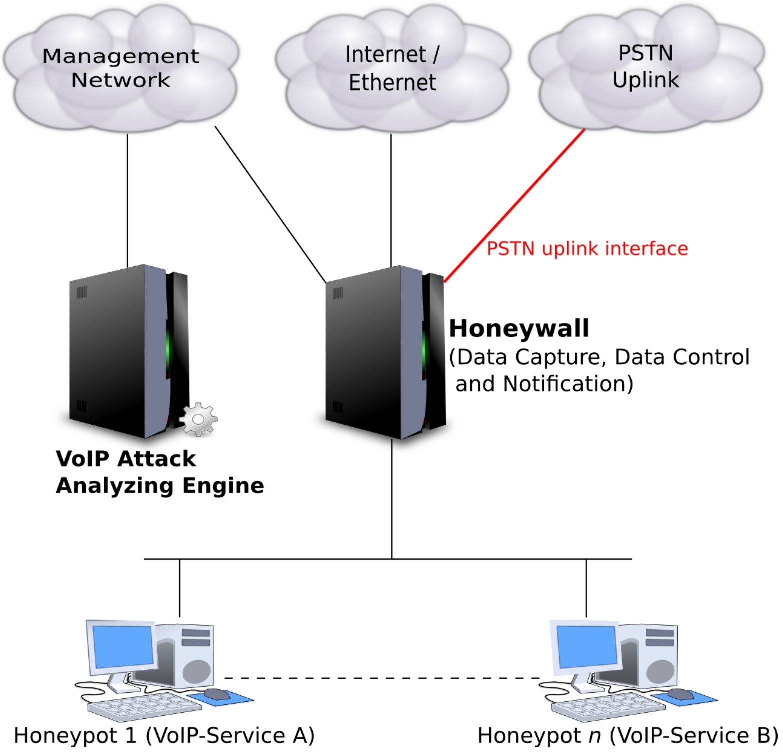

Management Network
Internet / Ethernet
PSTN Uplink
PSTN uplink interface
Honeywall
(Data Capture, Data Control and Notification)
VoIP Attack Analyzing Engine
Honeypot 1 (VoIP-Service A)
Honeypot *n* (VoIP-Service B)

Figure 7.1: PSTN uplink of the proposed VoIP honeynet architecture.

In contrast to other work (such as by Hoffstadt *et al.* [66]) which often gives an overview about fraudulent calls, we present the extent of fraudulent calls on real-world VoIP systems, analyze the attacks and finally try to determine the real attackers behind fraudulent calls.

7.2 Basic Setup of a PSTN Based VoIP Honeynet

For capturing the data of fraudulent calls we use the proposed honeynet approach with one honeypot and a PSTN gateway, as described in Chapter 5. The PSTN gateway connects a VoIP system with a PSTN system, as shown in Figure 7.1. This uplink can be activated and disabled by the honeywall to control the costs for calls to the PSTN system.

As a VoIP honeypot we used an Asterisk SIP server with four default users who had weak passwords to increase the probability of toll fraud attacks. If a caller tries to call a callee outside the local VoIP domain of the honeynet, the call is forwarded to the VoIP/PSTN provider via the PSTN uplink interface, who routes the call to the specific PSTN endpoint.

The VoIP attack analyzing engine can be customized to carry out various analyses, e.g., about the country of the callee's phone number, to show details of the signaling data or the possibility to classify the content of the calls. These features help to identify the intention of the attackers, e.g., regular calls, SPIT messages or unidirectional communications.

7.3 Analysis of Fraudulent VoIP Calls in the Honeynet

The packets captured by the VoIP honeynet were analyzed to get meaningful information about fraudulent calls. Using the results of the analysis we tried to identify the attack surface of VoIP/PSTN systems. Therefore, we analyzed various factors such as the local distribution of the hosts which initiate the calls, the local distribution of the calls to PSTN destination endpoints and the attacker behavior.

An example of a SIP address for calls to the PSTN system is listed in Listing 7.1. The IP address of the honeypot and some numbers of the PSTN extension are anonymized.

```
sip:0040736******@127.0.0.1 SIP/2.0
```

Listing 7.1: Example of a SIP address of a fraudulent call to a PSTN number

Description of the Captured PSTN Specific VoIP Data

The VoIP honeynet has been in operation since August 2011. For the analysis in this work the honeynet collected data up to and including March 2013. In this period the honeywall recorded `35,592,736` SIP messages. Figure 7.2 shows the number of captured SIP messages in the honeynet. The peak in March 2012 can be explained by the large number of SIP call attempts after the uplink interface was activated. In comparison to the other uplink periods, in March 2012 the number of calls was much higher but the length of the calls was shorter. Between September and December 2012, maintenance was performed on the infrastructure which explains the lack of activity during that time.

In the honeynet there were different periods with an activated uplink interface, because an active uplink interface produces costs for the operator of the honeynet. To control the costs we used a prepaid third party to connect the VoIP system with the PSTN network. Table 7.1 shows the activation periods of the uplink interface in the honeynet. The number of days with an activated uplink interface is seen in column "Active". The number of days until the first call in the period

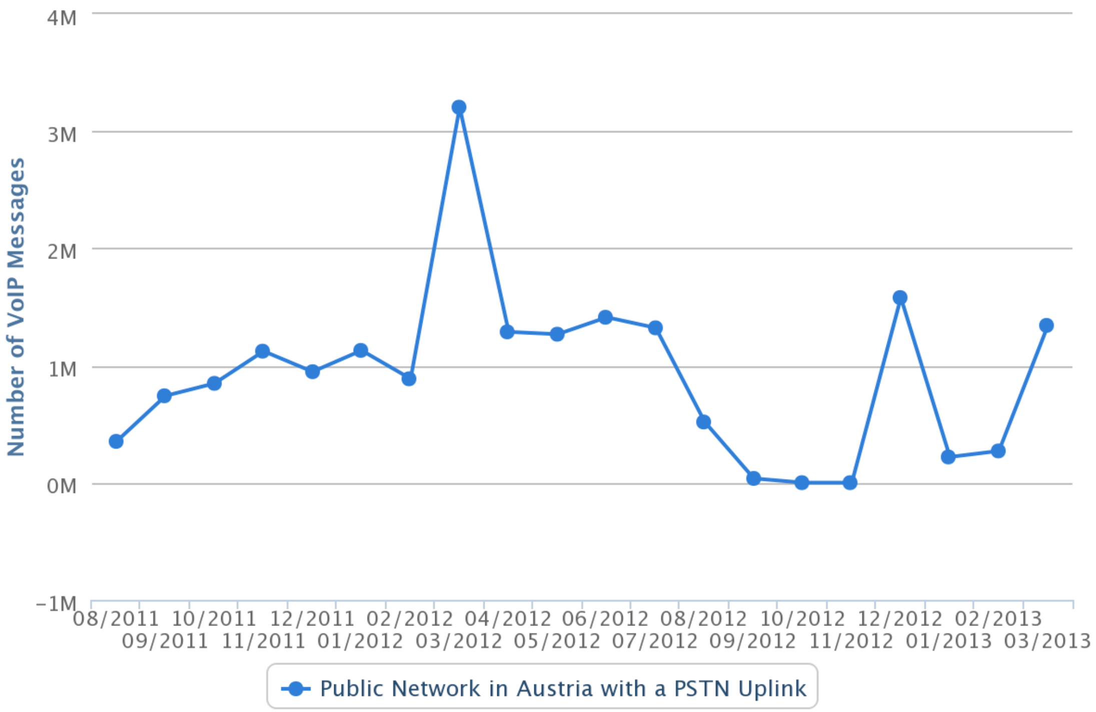


Figure 7.2: Number of captured SIP messages in the PSTN based VoIP honeynet.

	Activation period	Euro	Active	First Call	Empty
1.	2011-08-01 – 2011-08-19	10	19	19	1
2.	2011-10-17 – 2011-10-18	20	2	2	1
3.	2011-10-18 – 2011-10-20	20	3	1	2
4.	2012-03-28 – 2012-04-01	100	5	3	2
5.	2012-04-02 – 2012-04-06	100	5	1	4
6.	2012-12-06 – 2012-12-16	800	11	2	9
7.	2013-01-17 – 2013-01-28	100	12	7	5

Table 7.1: Periods of the activated uplink interface in the PSTN based VoIP honeynet.

is seen in column "First Call". Column "Empty" presents the number of days until the credit was used up. It can be seen that the activation periods are very short. Even without actively promoting our SIP honeynet service and IP address, it generally does not take long until the first attacker identifies the activated uplink interface. After that, credit is consumed very quickly. The time to detect the open uplink by the attacker and the duration until the credit is used up are in most cases only a few days, as can be seen in Table 7.1.

Analyzing the Signaling Data of Fraudulent Calls

To get information about the attackers' behavior for toll fraud attacks, the captured attacks were investigated in detail. All captured calls in the evaluation period tried to call an external number in a PSTN system, in most cases an international number or a premium number. No calls and no attempts to other VoIP users in the local domain were detected.

It appears that attackers gain the most profit from a successful attack if they can use a valid account in the VoIP system which allows them to make international calls for free. Such an attack can be a real business case for criminals, because one weak account password is enough to route all the VoIP traffic via this account. In all captured cases the method used for identity theft was brute-force attacks on accounts and passwords. Exploiting applications were not seen in the VoIP honeynet. In a successful case the attackers routed all PSTN calls of their own VoIP systems through the compromised system. The cost is not incurred by the caller, but the owner of the VoIP account, who has to pay the costs for these calls. We found, that a couple of IP addresses scanned our SIP service and tried to register with a known account from an earlier attack.

Figure 7.3 shows the recurring and the unknown hosts per month in our honeynet. There is only a small number of recurring hosts (who had accessed the honeynet already) and each other host was unknown. The peak in December 2012 can be explained by the activated uplink interface to PSTN and the higher uplink credit (800 Euro) than in the previous periods. It would appear that an attacker controls different hosts, because after one successful call to a PSTN endpoint, the number of calls from different hosts increases dramatically. After the uplink interface was disabled (because all the credit was consumed) the number of calls decreased again.

To get a better understanding of the attackers of toll fraud attacks we analyzed the home countries of the callers. The result of the analysis is shown in Figure 7.4. Each identified source IP address for toll fraud attacks was mapped with GeoIP [52] to get the countries of origin of the systems used by the caller or the probing instance. Most attacks were coming from Egypt, which had six times more callers than the Netherlands. And the Netherlands still had more callers than all remaining countries combined. The third most common address was an anonymous proxy, therefore we could not identify the country of origin.

The basic information about toll fraud attacks is presented in a compact form in Table 7.2. The first row shows the total number of attempts to establish a phone call to any destination. The second row shows the number of successfully established phone calls to PSTN endpoints. The third row shows the number of different hosts which attempted to establish a call. The fourth

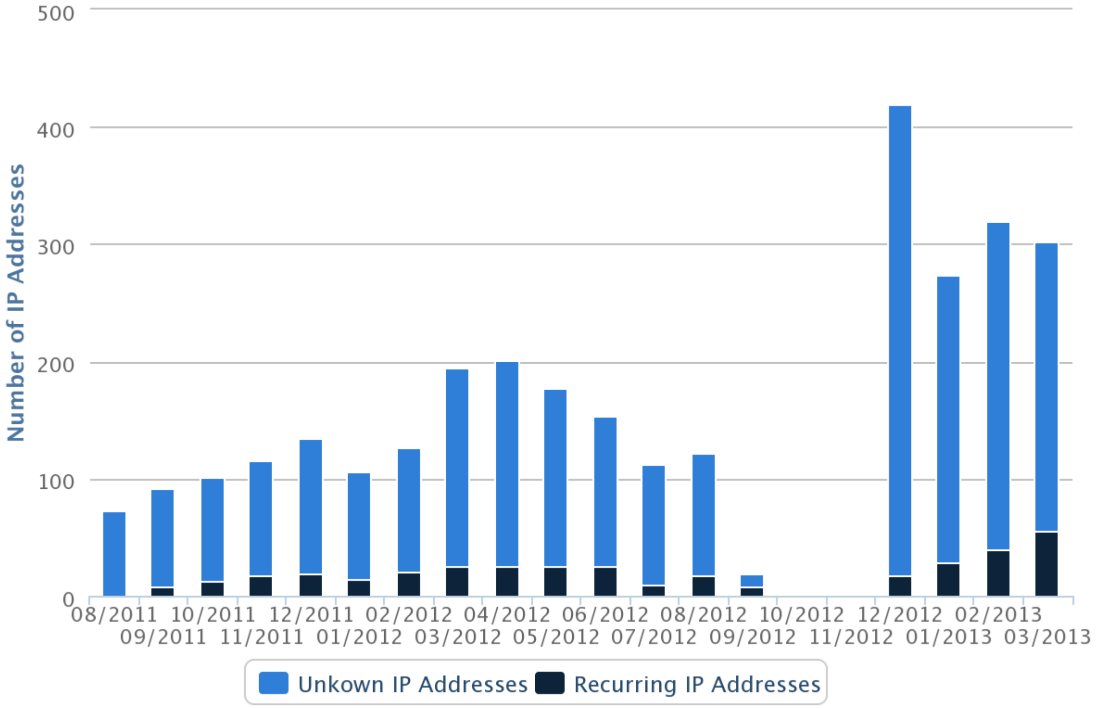


Figure 7.3: Number of unique source IP addresses in the PSTN based VoIP honeynet with the intent to call PSTN numbers.

	Number of	Number
1.	INVITE messages	72,362
2.	Established phone calls	4,311
3.	Different IP addresses	1,081
4.	Different IP addresses only established phone calls	170

Table 7.2: PSTN based VoIP honeynet statistic of fraudulent calls.

row shows the number of different hosts where a phone call was actually established. The large number of total attempts is due to probing whether a call to the PSTN system is possible or not, even when the prepaid account is empty.

The analysis of the calling codes of the destination phone numbers shows the countries which received the most calls, as can be seen in Figure 7.5. In comparison with the results of the analysis of the countries of origin of the systems used by the caller, these results show that most

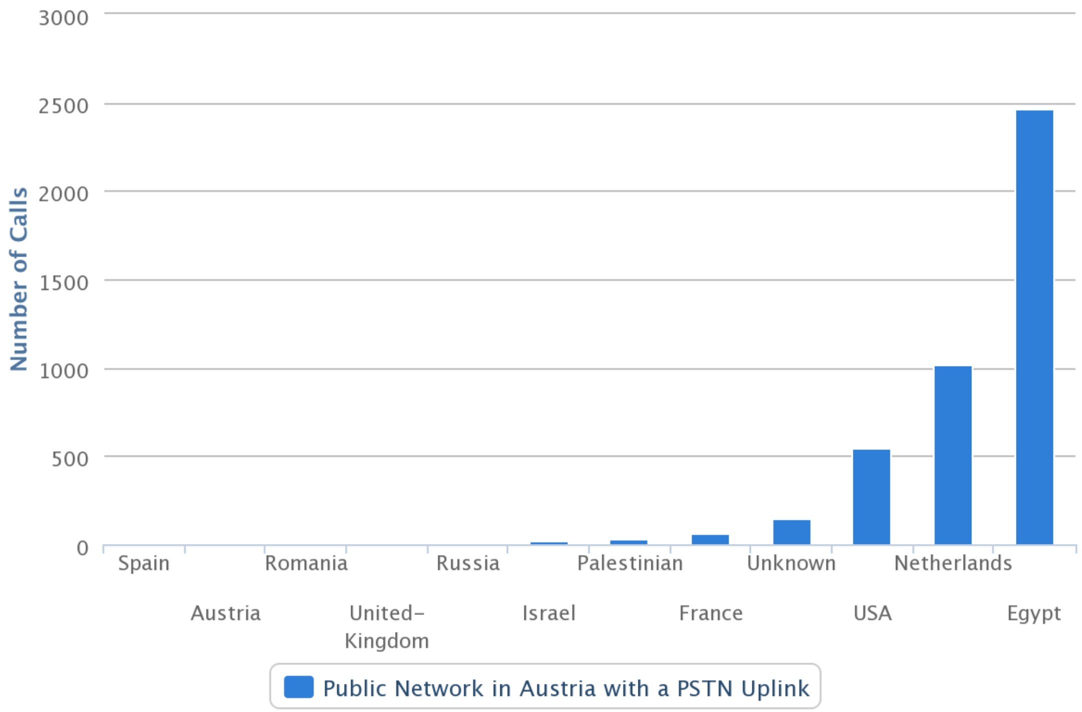


Figure 7.4: Number of established PSTN calls grouped by countries of origin in the PSTN based VoIP honeynet.

calls are international calls. Ethiopia and American Samoa are the most common destinations of the calls.

The costs of toll fraud in an activated uplink period look like an exponential curve. When the probing instance of an attacker recognized that calls to the PSTN were possible, the attacker started to use the compromised service. In many cases the number of callers increased after the activation of the uplink interface. It seems like a propagation in a community, well-organized attackers or only one attacker, e.g., as a controller of a botnet.

None of the attacks were performed with only one static IP address. In most cases the IP addresses changed frequently. Only one IP address occurred very often. This IP address was associated with a host in the Netherlands. A port scan identified several open ports and it appeared the caller was using a Mac system, which was probably a compromised system.

To identify the attackers' tools we analyzed the distribution of the SIP UA used for the established phone calls and the register attacks, as can be seen in Table 7.3. The large number of `VaxSIPUserAgent/3.0` suggests that a large number of attackers were using the same tool,

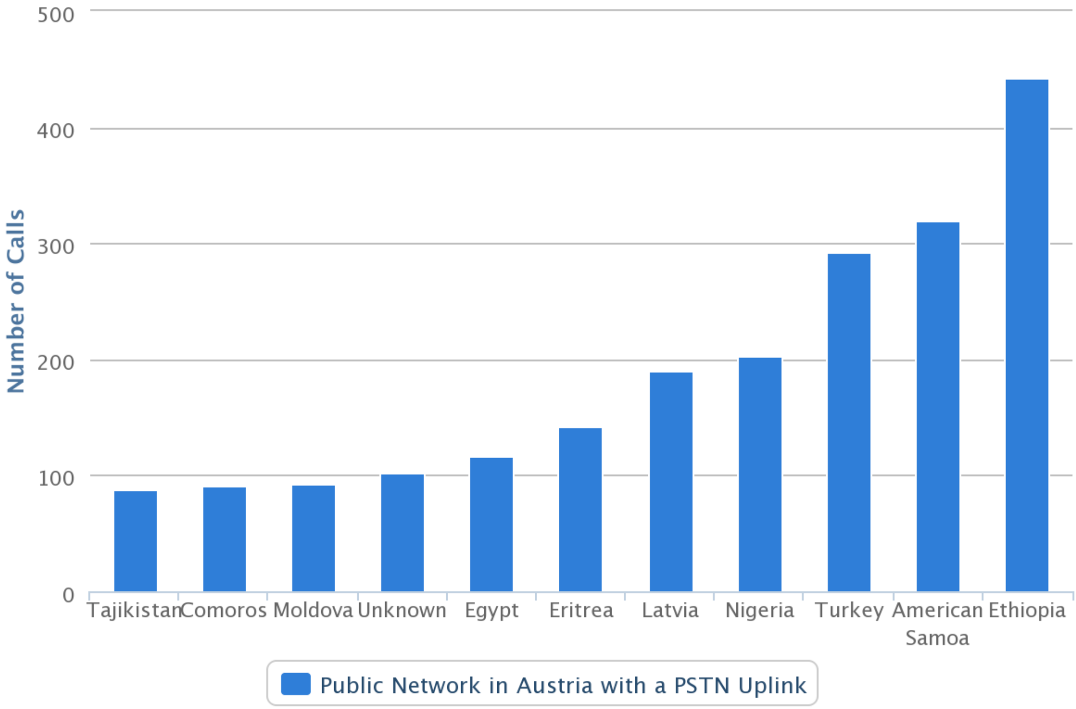


Figure 7.5: Top destination calling codes grouped by countries in the PSTN based VoIP honeynet.

or the attacks were controlled by a few masters, like a botnet. We identified the same structure in the SIP headers of all the SIP messages with `VaxSIPUserAgent/3.0` user agent, which suggests that the UA string is not manipulated. `VaxSIPUserAgent/3.0` refers to a SIP SDK which provides tools and components for quickly adding SIP-based telephony features to web pages and software applications. It is available for Windows and Mac systems. We guess the caller is infected with malicious software and the attacker uses the caller to forward the calls to our honeynet. Many caller IP addresses were behind a router with weak or default passwords in a small home network. They often had open TCP/UDP ports (such as FTP, SSH or HTTP) which indicates that the caller had not invested any effort in concealing their identity. The UA most often used for register attacks is clearly the UA `friendly-scanner`.

Analyzing the Media Data of Fraudulent Calls

The VoIP honeynet records all phone calls to PSTN to get even more information about the attacker, their behavior and their intentions. As already mentioned, the intention was to cap-

User agent for toll fraud attacks	Number	User agent for register attacks	Number
VaxSIPUserAgent/3.0	1,554	friendly-scanner	11,017,276
eyeBeam release 3010n	841	nike	117,565
X-Lite release 1011s	596	MizuPhoneFree/2.1.0	80,403
X-Lite release 1104o	262	X-Lite release 1104o	35,749
Linksys/PAP2T-3.1.15(LS)	238	*empty*	29,425
MizuPhoneFree/2.1.0	153	X-PRO build 1101	22,892
MizuPhoneFree 2.1.3.123	102	eyeBeam release 3010n	11,352
eyeBeam release 3006o	98	X-Lite release 1011s	5,537
X-Lite release 1002tx	81	MizuPhoneFree 2.2.0.6	4,578
Nuvois release 11010f	80	MizuPhoneFree 2.1.3.123	3,828
eyeBeam release 3007n	68	eyeBeam release 3006o	1,859
X-PRO build 1101	51	VaxSIPUserAgent/3.1	1,624
SIPPER for PhonerLite	48	MizuPhone 2.2.0.6	1,005
eyeBeam release 1100z	30	SIPPER for PhonerLite	919
MicroSIP (c) 2010	13	Linksys/PAP2T-3.1.15(LS)	800
MizuPhoneFree 2.2.0.6	13	eyeBeam release 3007n	783
NCH Software Express Talk 4.26	11	VaxSIPUserAgent/3.0	510
Zoiper rev.11137	11	Nuvois release 11010f	465
empty	10	X-Lite release 1002tx	236
others	51	*others*	1037

Table 7.3: Distribution of the SIP user agents for established calls and register attacks in the PSTN based VoIP honeynet.

ture only calls from attackers and not from regular users. The caller and the callee can speak undisturbed and should not recognize that the call is being recorded. The calls can be stored in popular audio file formats, like wav or mp3, and can be replayed for analysis purposes.

Most calls were identified as human conversations without any suspicious characteristics. However, we also identified calls from different hosts and different countries to various PSTN endpoints with the same content language. For example, a host from Egypt called a number in Hungary and they spoke in Arabic. Such constellations were often found during our analysis. It appears that the attacker is a call shop (either physical or in the Internet) which offers cheap calls to North African countries, because the spoken language of most of the calls was Arabic. These call shops do not pay the costs for the PSTN call, because rather than using their own infrastructure, they misuse the hacked infrastructure of other providers to establish their calls. After the analysis and identification of regular calls through the honeynet solution all the captured calls were deleted.

Before our analysis it was presumed that SPIT would be a major motivation for the calls. However, after the voice analysis we can now exclude SPIT as the purpose for the calls, because we never identified an advertising call.

7.4 Uncovering the Attacker Model

The results of the detailed analysis from this chapter give us a broad knowledge of VoIP toll fraud attacks and the various intentions of the attackers. We analyzed some of the attacks in detail to get more information about the probable business model.

The attackers use different hosts for toll fraud attacks, but the patterns of the attacks are very similar. The host of the caller has no open SIP port on the standard port. This suggests that the host only forwards the calls and expects no incoming calls, because otherwise there would have been an opened SIP port 5060 to accept incoming calls. Blacklisting IP addresses would not work against these kind of attacks, because the callers change their IP addresses frequently, as can be seen in Figure 7.3.

By analyzing the phone calls we saw that various phone calls to different PSTN destinations were made from the same location. We identified similar background noises in different calls, such as the same baby crying or background conversations. It therefore seems that the attacker calls different numbers, accepts the calls and leaves the connection open for a long time without any conversation in order to earn money. This fits a business case whereby attackers earn money by calling their own premium rate numbers using stolen VoIP accounts.

After analyzing the voice patterns of the communications between the callers and callees, it was established that in most cases the purpose of the calls was normal conversations and not communications between attackers. However, we also identified many short calls to different countries all over the world in a very short time span. This suggests that the goal of the attackers was to identify the countries to which PSTN calls were possible. In most cases, these calls are unidirectional communications, e.g., the destinations are advertising numbers or mobile boxes. This attack is a very aggressive one, because a large number of calls will be tried in a short time on one VoIP system.

We can divide the attacks in two phases, the probing phase to identify VoIP systems with an uplink to PSTN and a misuse phase to generate money for the attackers. Figure 7.6 presents the supposed architecture of the identified attacker model and highlights the attacker (red), the target (green) and the regular users of a VoIP system.

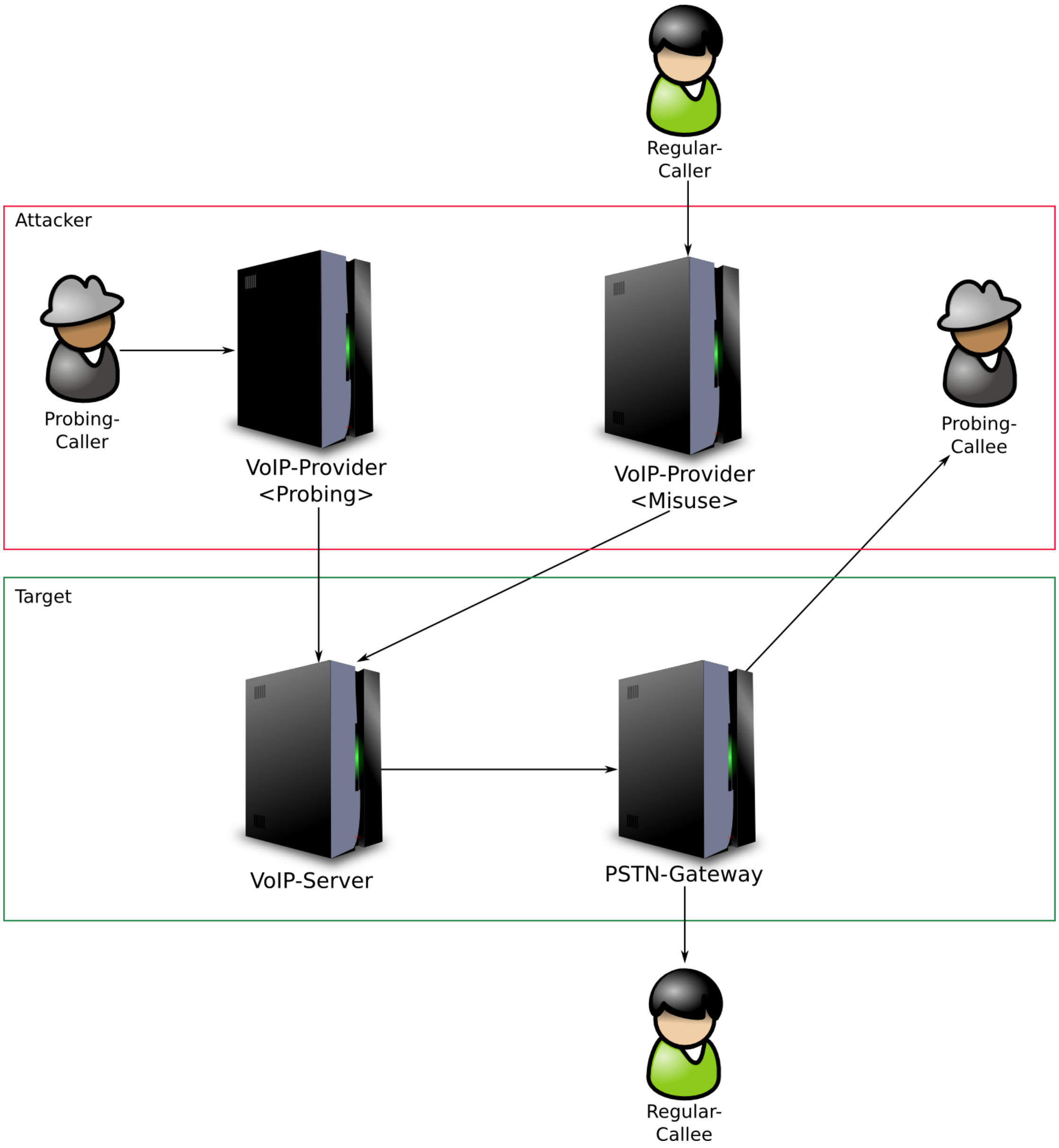


Figure 7.6: Supposed architecture of the identified attacker model of fraudulent calls.

Probing Phase

In the probing phase of an attack the probing instances are searching for VoIP systems with weak accounts. If an attacker finds a valid account they use this information to route the calls through an account on this VoIP system with special forwarding hosts. This assumption has been confirmed in our analysis, because we detected IP addresses without REGISTER attacks and we also detected IP addresses without INVITE attacks. Of course, it could be the same attacker with dynamic IP addresses, but the low number of these occurrences does not seem to indicate that.

Once the attackers have compromised an account, they periodically try to validate this account. They try to authenticate on the VoIP system and they try to call a few PSTN numbers to validate if the uplink is activated.

It would appear that these attacks were automated for two reasons. First, the system was being probed regularly, even when there had not been any credit available for a while. This means that the attackers detected the activation of the uplink shortly after it occurred and started to misuse the system. Secondly the majority of attacks had the same user agent, often an indication of a distributed bot system.

Misuse Phase

The attackers misused our infrastructure to forward VoIP calls to PSTN numbers. The analysis shows that only PSTN calls are forwarded to our honeypot, other VoIP users were never contacted. It can clearly be seen that the number of forwarding hosts rises after the identification of vulnerable SIP systems with a PSTN connection. Unfortunately, the source IP addresses of the forwarding systems change often and quickly, so blocking an IP address is not a viable defense.

Figure 7.7 shows the expected operating systems of the most frequently occurring source IP addresses of SIP connections. Most of the hosts, which connected to our honeypot were behind a NAT network with probably only a few clients, which were unknown to us as they were behind a NAT. Therefore, we do not know the exact operating system of the clients, but the most common user agent `VaxSIPUserAgent` is only supported for Windows and Mac. It is suspected that a bot system is running as a client to forward the calls. In most cases no way back through the NAT system to the client could be found, because no route to a VoIP server was defined in the NAT system. The attacking system, i.e., one of the hosts behind the NAT, only forwards the call from another VoIP system. It is necessary that the client initiates the communication between

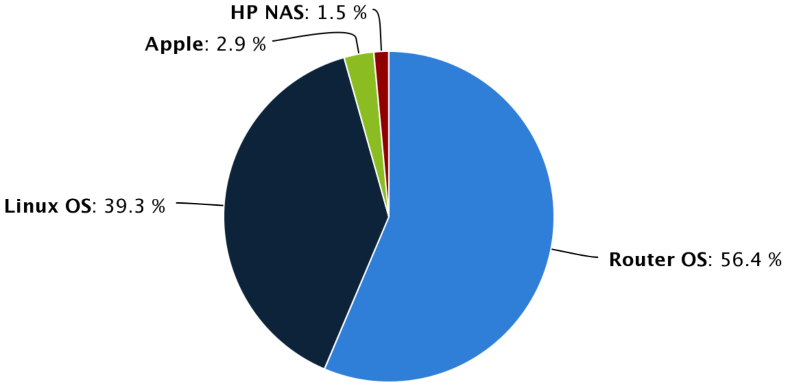


Figure 7.7: Surmised operating system of the most frequented hosts in the PSTN based VoIP honeynet.

the attacker's base station and the victims, e.g., the honeynet. This helps the attackers to protect their own systems.

7.5 Conclusion

The PSTN specific VoIP honeynet architecture provided a lot of data to analyze and evaluate fraudulent calls originating from all over the world.

The analysis of toll fraud attacks has shown that VoIP security is critical and toll fraud is an important aspect of VoIP security, because a huge number of toll fraud attacks were recognized by the honeynet. Toll fraud can be very expensive for the owner of a compromised VoIP account, because in most cases calls to PSTN systems incur interconnection costs.

Analysis of the signaling data reveals that most fraudulent call attacks are used to make free international calls. Signaling data also reveals that most fraudulent call attacks originate from Egypt and the Netherlands. Most of the attackers use `VaxSIPUserAgent/3.0 SDK` as UA, which suggests that a botnet is in control of the fraudulent call attacks. The IP addresses of the attacking hosts changed frequently. Finally, most of the calls were in Arabic.

The details of VoIP attacks, especially toll fraud attacks, have to be considered in the design of a secure VoIP solution, which will be discussed in the next chapter.

Part III

Hardening VoIP Systems – Countering Real-World Attacks

CHAPTER 8

Design Criteria for a Secure VoIP Solution

Contents

8.1 Introduction

The theoretical and empirical analysis in conjunction with the evaluation of the threats and vulnerabilities of VoIP systems showed us that various attacks exist and that attackers try to misuse the system in different ways. Additionally, the different VoIP providers in a call setup can intercept the communication (like our VoIP honeynet) and with the recent revelations about state-organized mass-surveillance of the Internet communication, new protection mechanisms for VoIP communications are required. Otherwise VoIP providers or third parties (such as intelligence services) can detect who is talking to whom, when, how long, as well as record the conversation of the VoIP call.

This chapter proposes design criteria for secure VoIP communication which reduce the main identified security risks to an acceptable level.

Based on the security process (as described in Chapter 3) Section 8.2 covers the identification of threats and risks. The analysis and evaluation of the identified threats and risks are described in Section 8.3. The selection and design of countermeasures are presented in Section 8.4.

8.2 Identified Threats and Risks to VoIP Systems

When VoIP was introduced, the main focus was on functionality and QoS of the systems rather than security [44]. Since then, this has changed because VoIP has gained widespread acceptance and significance, and crucially because of the revelations of mass-surveillance and various privacy violations, such as authorities capturing and analyzing communication data in the Internet [146, 32].

Based on the revelations of the NSA surveillance programs we describe the level of risks to Internet communications (based on actors) and the required protection mechanisms to counter them as:

1. **Sophisticated intelligence services with unlimited financing and reach:** Dedicated highly secure tools for hardware and software are needed and the operational communication procedures have to be tailored in order to communicate securely.
2. **Sophisticated intelligence services with limited financing and reach:** Special tools only for hardware and software are sufficient to protect communication in the Internet.
3. **Other intelligence services / state actors:** Some standard tools and widespread use of special tools are sufficient to protect against these actors.
4. **Organized crime and law enforcement:** Standard tools and selective use of special tools helps to protect against this risk.
5. **Individuals or groups and "hackers":** Standard tools used wisely are enough.

The proposed countermeasures focus on the protection of the application level of VoIP systems and not against all possible methods available to sophisticated intelligence services (such as compromising hardware). In our approach it is assumed that the endpoints (e.g., mobile devices or Personal Computers (PCs)) are not compromised and no additional specialized interception mechanisms are used, e.g., shoulder surfing or surveillance through microphones, cameras, people/staff. Various protection mechanisms are described in [29, 23, 74].

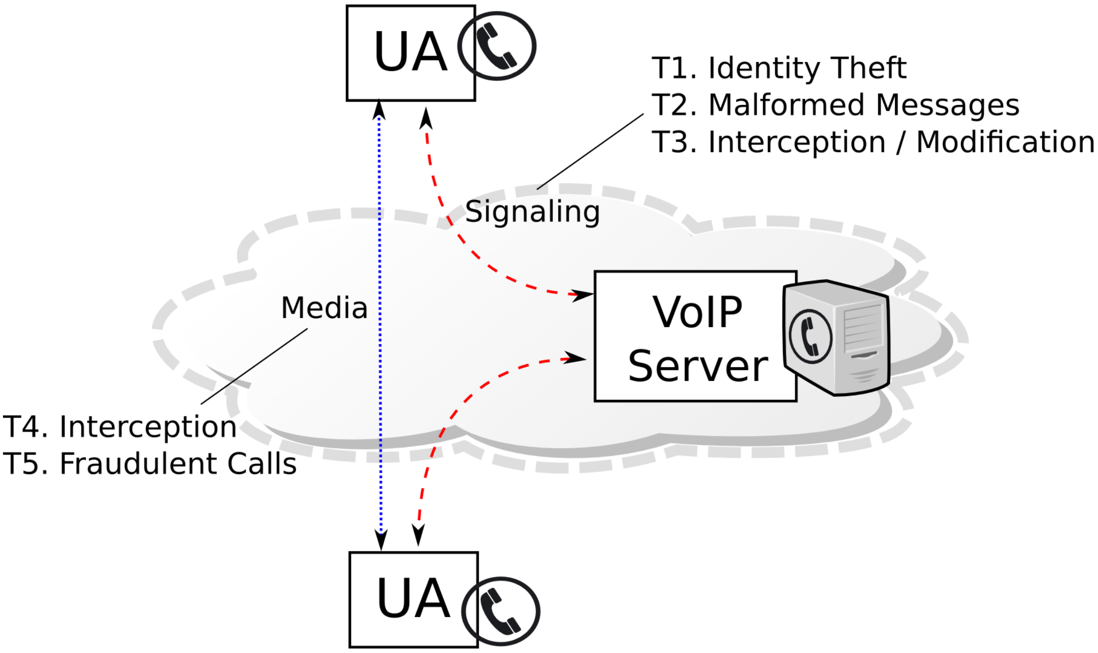


Figure 8.1: Main identified threats to VoIP systems by our analysis.

This also means that simple protection mechanisms (e.g., strong passwords against identity theft) against the identified attacks on VoIP systems are often not sufficient, or only covers the lowest risk level. For a higher risk level a strong and overall protection concept (for signaling as well as media transmission) is needed.

Figure 8.1 shows the main threat classes of VoIP systems, as identified in Chapter 3, 6 and 7. The threats are modeled in threat classes because many threats have the same origin or the same structure. Through the use of the three honeynet instances a lot of information about VoIP attacks was gained. For the signaling part the main problems are identity theft, malformed messages and interception (maybe modification) of SIP messages. Identity theft is critical because the attackers try to get a valid account from another person (e.g., by brute-force). However, a strong authentication mechanism is needed to avoid identity theft from the lowest risk level (based on actors) as well as the upper levels. Table 8.1 summarizes the identified threats to VoIP systems, and indicates which threats were also identified by our honeynet in the real-world (marked as T).

Usually in a probing attack for identity theft, the VoIP server gives unnecessary information to an attacker (e.g., UA string of the server or information about existing, but not necessarily registered users in the VoIP system). Therefore, a VoIP server should only answer (on an application level)

Threat-Class	Risk	Description
Threat-1	Service abuse – identity theft (T1)	Identity theft of existing VoIP accounts.
Threat-2	Malformed messages (T2)	SIP messages with incorrect syntax were sent to the VoIP server by an untrusted component.
Threat-3	Eavesdropping and interception (T3, T4)	Interception of the signaling as well as the media data is possible by a rogue VoIP provider as well as organizations.
Threat-4	Service abuse – fraudulent calls (T5)	Attacker calls a victim with fraudulent intentions through a hacked infrastructure. (Take advantage of identity theft)
Threat-5	Modification (T3)	Each message can be modified by an untrusted component without detection by the communicating parties.
Threat-6	DoS/Service abuse – Malicious messages	Unexpected SIP messages were sent to the VoIP server by an untrusted component.

Table 8.1: Identified threat classes to VoIP systems by our honeynet approach.

if the communicating partner is part of an established trust relationship. The trust relationship should also decrease the number of malformed messages, because untrusted messages should not be processed. As we have seen with our honeynets, the interception and also the modification of SIP messages is very easy and has to be protected against.

Media data interception (privacy violation) is a major problem too. The operation of our honeynet solution showed us how easy it is for a MitM (e.g., an untrusted provider) to capture data, without arousing the suspicion of the communicating parties. A key deduction of the analysis was that some attackers (maybe VoIP providers) are acting outside the law by hacking accounts and using them for their customers to call PSTN numbers.

Without end-to-end encryption of the VoIP data, the threat of eavesdropping or modification of packets by a compromised network or rogue proxies is still alive. VoIP systems can be protected against interception from “simple hackers” by the use of SIPS instead of SIP, but not for a higher level of actors. The disadvantages of SIPS, various key-exchange protocols and TLS security were already discussed in Chapter 3.

Occurrence likelihood	Possible damage		
	Normal	High	Very High
Possible	Low	Medium	High
Probable	Low	Medium	High
Very Probable	Medium	High	High

Table 8.2: Risk classification based of the occurrence likelihood and the possible damage.

8.3 Evaluation of the Identified Threats

The identified threats and vulnerabilities were evaluated using their potential damage and the likelihood of their occurrence. The level of risk represents the possible harm and the probability of occurrence.

In the risk assessment process the BSI defines three protection requirement categories (based on possible damages) which are later used for classifying the items that are in need of protection (described in standard 100-2 [20]):

- **Normal** protection requirements: “The impact of any loss or damage is limited and calculable.” (BSI 100-2 [20])
- **High** protection requirements: “The impact of any loss or damage may be considerable.” (BSI 100-2 [20])
- **Very High** protection requirements: “The impact of any loss or damage can attain catastrophic proportions which could threaten the very survival of the agency/company.” (BSI 100-2 [20])

The occurrence likelihood of the threats and risks to VoIP systems are categorized as:

- **Possible**: Sometimes detected by the VoIP honeynet.
- **Probable**: Often detected by the VoIP honeynet.
- **Very Probable**: Very often detected by the VoIP honeynet.

To assess the risks, the occurrence likelihood is put in relation to the possible damages (as seen in Table 8.2). The risk categories are defined as “Low”, “Medium” or “High”.

Threat-Class	Description	Occurrence likelihood	Possible damage	Level of risk
Threat-1	An attacker can pretend to be someone else for malicious activities	Very Probable	Very High	High
Threat-2	An attacker can disturb the regular use of the VoIP system	Possible	High	Medium
Threat-3	Every eavesdropper knows who is talking to whom and also the communication payload	Probable	Very High	High
Threat-4	An attacker can use the hacked infrastructure to forward chargeable calls	Possible	Very High	High
Threat-5	Each communication can be redirected to other people without detection by the caller	Possible	High	Medium
Threat-6	An attacker can disturb the regular use of the VoIP system	Possible	Normal	Low

Table 8.3: Evaluation and risk classification of the identified threat classes to VoIP systems.

Each identified threat was analyzed and evaluated to define the level of risk based on the presented protection requirements, the occurrence likelihood requirements and the risk classification. Table 8.3 presents the level of risk for all identified threat classes to VoIP systems according to our analysis. The risk level of Threat-1 is classified as high, because most of the attacks identified in the honeynets were identity theft attacks and the possible damage can be very high. Threat-2 has a risk level of medium, because the possible damage is high for the VoIP system (e.g., DoS or application failure), but the occurrence in the honeynet was medium and no malformed message attacks with the intent of DoS were documented. Threat-3 has risk level high, because this attack is hard to recognize and in the worst case each communication can be intercepted without recognition. Therefore, the possible damage is very high. The analysis from Chapter 7 showed that fraudulent calls are still a problem and can cause major financial damage for the operators of a VoIP system. Therefore, the risk level for Threat-4 is classified as high. The risk level of Threat-5 is medium, because the possible damage of a modified message without detection is high, but this attack was not often identified by the honeynets. Threat-6 has a risk level of low, because DoS attacks were recognized in the most cases immediately and the occurrence in the honeynet was medium.

8.4 Implications for Building a Secure VoIP Solution

The previously identified VoIP security risks, threaten confidentiality, integrity, availability, authenticity/trustworthiness and privacy of VoIP communication, and require new thinking in designing secure communication systems. Therefore, only the accepted and authenticated participants should be able to process the data so that no third-party can intercept and interpret the communication data. A secure VoIP system should also provide anonymous and non-traceable communications, but security measures should not limit usability.

Current and future VoIP solutions have to protect users from these risks, and as well as deal with new security issues and flaws which effect sensitive messaging and communication systems. The aim is to explicitly give data sovereignty back to the communicating partners and constrain every user in the system to a minimal set of essential tasks and functionalities needed to meet their responsibilities, thereby reducing the scope for malicious behavior.

These mechanisms do not help against a compromised end-point device or a trusted but malicious user of the system.

The proposed design criteria for secure VoIP communication cover all currently identified risks to VoIP systems. Table 8.4 shows the trust model underlying the design of a security layer for closed VoIP systems. A closed VoIP system means that only users known to the system can use the system to make international VoIP calls (not only in closed network infrastructures). That means the communication should be possible within private systems (e.g., a company or an organization), but calls to PSTN networks do not have to be supported. The trust is only granted to the UA and the VoIP Server, which makes it necessary to create a secure path between these components irrespective of the underlying network architecture.

To be able to continue using the existing VoIP applications but nevertheless increase the security level, our approach is based on a conventional VoIP architecture which fulfills the following conditions:

- Separate protocols for signaling and media operation are used (e.g., SIP and RTP).
- For signaling purposes, an UA may not directly communicate with other UAs. The usage of signaling or proxy servers is compulsory.
- For transferring media data, the UA should communicate as directly as possible with peer UAs. If necessary, media relay servers can be used.

Component	Trusted	Comment
VoIP Server	X	For this approach the VoIP server components are seen as trustworthy.
UA	X	For this approach the UA is seen as trustworthy.
Provider for Data Communication	-	Data communication providers are not considered trustworthy. They might be compromised or forced to give out data.
External VoIP Provider	-	3rd party VoIP providers are not considered trustworthy. They might be compromised or victim of abuse etc.
Other Institutions	-	Institutions outside the system cannot be seen as trustworthy, as they might compromise the communication.

Table 8.4: Trust model underlying the design of a secure VoIP system.

To implement protection mechanisms for secure VoIP communication based on existing signaling and media protocols the following high-level requirement classes must be met.

- **REQ-1**: A VoIP server should only be accessible after a secure channel [22, 102] has been established.
  - **REQ-1.1**: A VoIP server should only exchange media data in a confidential way, with the content not being accessible to third-parties.
  - **REQ-1.2**: Furthermore, not only media content, but also the signaling data must be adequately protected so as not to reveal more information than necessary.
  - **REQ-1.3**: The secure VoIP solution should also guarantee interception-safety and protection against caller and callee tracking, and identification via signaling data or other metadata through untrusted components.
  - **REQ-1.4**: Decryption of past calls should not be possible, even if the whole communication (including signaling and media data) has been intercepted.
  - **REQ-1.5**: The cryptographic mechanisms to protect VoIP data have to be well researched and should avoid already identified weaknesses (e.g., for ZRTP, SRTP, SIPS, TLS/SSL as discussed in Chapter 3).
  - **REQ-1.6**: Only participating parties can process the content of the media data, therefore the data should be end-to-end encrypted. This is different to VoIP solutions which build on a Virtual Private Network (VPN) tunnel for example, where a central

VoIP server is used as a contact point and the data inside the tunnel to this server might still be transferred unencrypted. In our case, each conversation between a UA and also the conversation with the VoIP server is encrypted by a dedicated key.

  - **REQ-1.7**: The secure channel establishment should also be usable on mobile devices.
  - **REQ-1.8**: The protection mechanism has to consider the bandwidth consumption of the security layer, because some networks have limited resources. Therefore, the overhead should be minimized so that it is usable in slow communication networks.
  - **REQ-1.9**: The data encryption can only be compromised if the client or the server itself is compromised.

- **REQ-2**: The access to the VoIP server should be protected by key-based authentication.
  - **REQ-2.1**: The cryptographic mechanisms have to be state-of-the-art and not compromised.
  - **REQ-2.2**: The authentication mechanisms have to protect against MitM attacks.
  - **REQ-2.3**: The key-based authentication should also be usable on mobile devices.
  - **REQ-2.4**: The mechanism can only be compromised if the client or the server itself is compromised.
- **REQ-3**: To protect the server from DoS and brute-force attacks the clients have to show proof of work to the server.
  - **REQ-3.1**: The client-based proof of work should use cryptographic mechanisms (similar to the Hashcash system [89]).
  - **REQ-3.2**: VoIP data should only be processed by the VoIP server after a secure channel has been established.
  - **REQ-3.3**: The proof of work should also be usable on mobile devices.
- **REQ-4**: The source of fraudulent calls is identity theft. Therefore, strong authentication will also reduce fraudulent call attacks.
  - **REQ-4.1**: Other VoIP providers also have to authenticate each other via strong authentication to build a trust relationship.
  - **REQ-4.2**: Each call establishment has to be accountable to a trusted participant.
- **REQ-5**: Each VoIP message should be protected against modification.

	REQ-1	REQ-2	REQ-3	REQ-4	REQ-5
Threat-1	-	X	-	-	-
Threat-2	-	-	X	-	-
Threat-3	X	-	-	-	-
Threat-4	-	-	-	X	-
Threat-5	-	-	-	-	X
Threat-6	-	-	X	-	-

Table 8.5: Traceability matrix to correlate threat classes and requirement classes.

- **REQ-5.1**: The cryptographic mechanisms to ensure integrity have to be state-of-the-art and not compromised.
- **REQ-5.2**: The protection mechanism should also be usable on mobile devices.

Table 8.5 presents a traceability matrix to show the correlation of the proposed requirement classes to the identified threat classes. The coverage of the requirements shows that all main threat classes are treated by the countermeasures. The evaluation of the effectiveness of the proposed countermeasures and the residual risk assessment are presented in Chapter 10.

8.5 Conclusion

The identified risks to VoIP systems and the rising number of VoIP attacks make it necessary to provide additional security mechanisms to ensure not only confidentiality, integrity and availability but also authenticity/trustworthiness and privacy.

The defined design principles cover the main identified risks to VoIP systems, and should help to reduce the risks of the threat classes to an acceptable level. The principles were derived by a common risk management process including risk identification and risk analysis/evaluation methods.

Based on these design requirements the next chapter introduces an additional security layer for VoIP systems to fulfill these security criteria.

CHAPTER 9

Design and Implementation of a Transparent Security Layer to Enable Anonymous VoIP Calls

Contents

9.1 Introduction

The identified threats and vulnerabilities, and the risks from Section 8.2 require a robust and secure solution to protect VoIP calls.

We propose an approach to meet the requirements and design principals from Section 8.4 by establishing a non-invasive security layer between the network and the application layer of the IP stack.

The aim was to develop a system by combining existing cryptographic primitives taking their advantages and limitations into consideration. It was not to design new untested algorithms or protocols, because the error susceptibility should be as low as possible (especially for cryptographic algorithms). The approach described in this work takes the complete VoIP communication into account, therefore signaling as well as the communication data itself are part of the security considerations. In conventional VoIP systems insufficient security mechanisms (e.g., only protecting the media transmission) are often used (as described in Chapter 3).

The proposed approach for a secure VoIP communication solution takes extensive security and privacy precautions for users by adding a non-invasive security layer (called VoIP Security Layer (VSL)), to conventional VoIP systems. In addition to the encrypted communication data, communicating partners cannot be identified by third-parties analyzing the captured VoIP data, as signaling data and metadata are encrypted as well.

For this approach to work, closed VoIP systems extended with this integrated layer have to be used. The security measures of this approach are limited to any data transmitted between clients, and the central systems (e.g., the VoIP server). The security of the client itself is still dependent on the user, which has created its own fields of research (e.g., [25, 50, 121]).

The VSL also takes account of limiting factors in the VoIP systems, such as limited resources and low bandwidth communication networks, by avoiding unnecessary overheads (e.g., the packet size or number of requests in the handshake). We implemented a proof of concept to show its feasibility on mobile devices.

This approach helps to decrease the identified risks to VoIP systems described in Chapter 8 to an acceptable level.

9.2 Common Secure VoIP Communication Architecture

We use the concept of an additional non-invasive security layer, located between VoIP application layer protocols and network transport protocols, to protect common VoIP systems. This allows independent operation of VoIP protocols such as SIP or RTP, but still enables comprehensive coverage of the communication data generated and received by the VoIP agents.

The VSL uses a simplified SSL/TLS protocol without unnecessary legacy functions, in order to save resources and to keep the security protocol as simple as possible. Additional to the dedicated encryption between the UAs and the VoIP server, the media traffic will be end-to-end encrypted for the caller and the callee. Therefore, not even the VoIP server or any other third party can interpret the media data. This allows the protection of closed communication systems, e.g., companies or projects where multiple companies are involved.

A basic description of the communication flow is presented in Figure 9.1. The UA and the VoIP server have to establish a secure channel before the UA is able to communicate with the VoIP

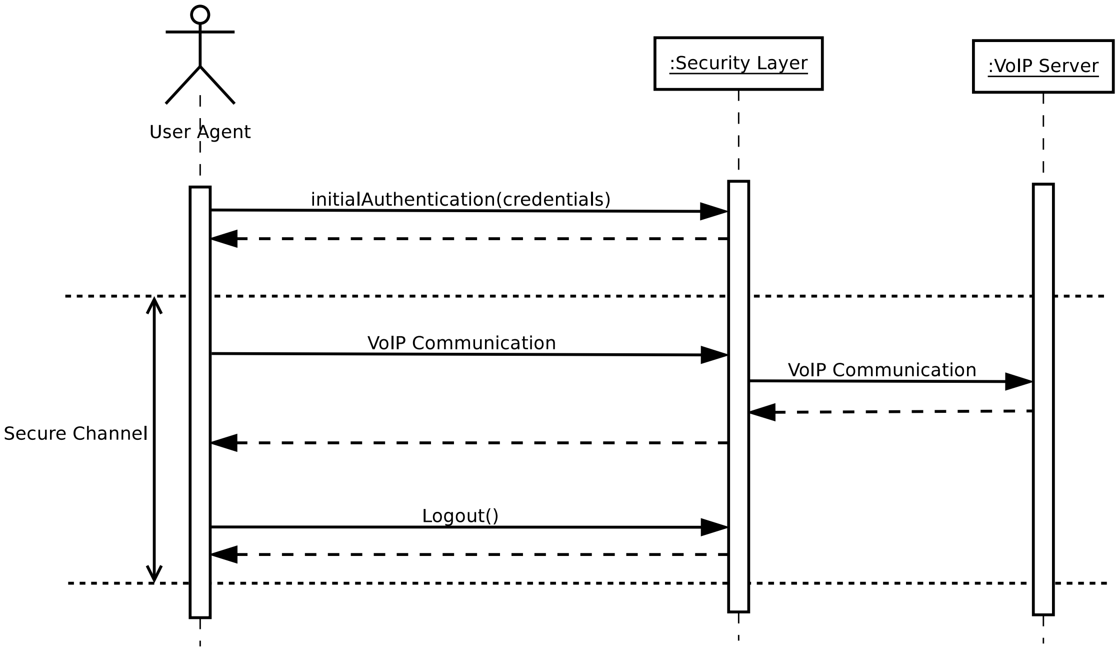


Figure 9.1: Sequence diagram of the proposed security layer for VoIP communication.

server. If the secure channel setup fails, no packets will be forwarded to the VoIP server. The secure channel setup includes a DoS protection through the use of cryptographic client puzzles, as described by Juels and Brainard [77].

However, when using cryptography in the context of VoIP systems, we need a mechanism for *authenticating* the communicating parties, high performance *cryptography algorithms* for the voice and signaling data, and a *key exchange mechanism* for simple distribution of cryptography keys. Therefore, each encryption key used is negotiated by a key exchange mechanism that also ensures the authenticity of the communicating partners and the integrity of the data exchanged.

Consequently, for each connection a dedicated symmetric encryption key is used, which is only known by the communicating partners involved. For signaling, the dedicated key is only known by the UA and the VoIP server. For the media stream, the shared key is only known by the caller and the callee. For a UA to participate in the secure communication, the VSL is required. Without the VSL, participation is not possible because the authenticity of the parties cannot be guaranteed and the VoIP system does not interpret any VoIP messages from a non-authenticated party.

Our proposed approach provides PFS [36] for the signaling communication between the UA and the VoIP server as well as the direct media communication between the UAs. This means that

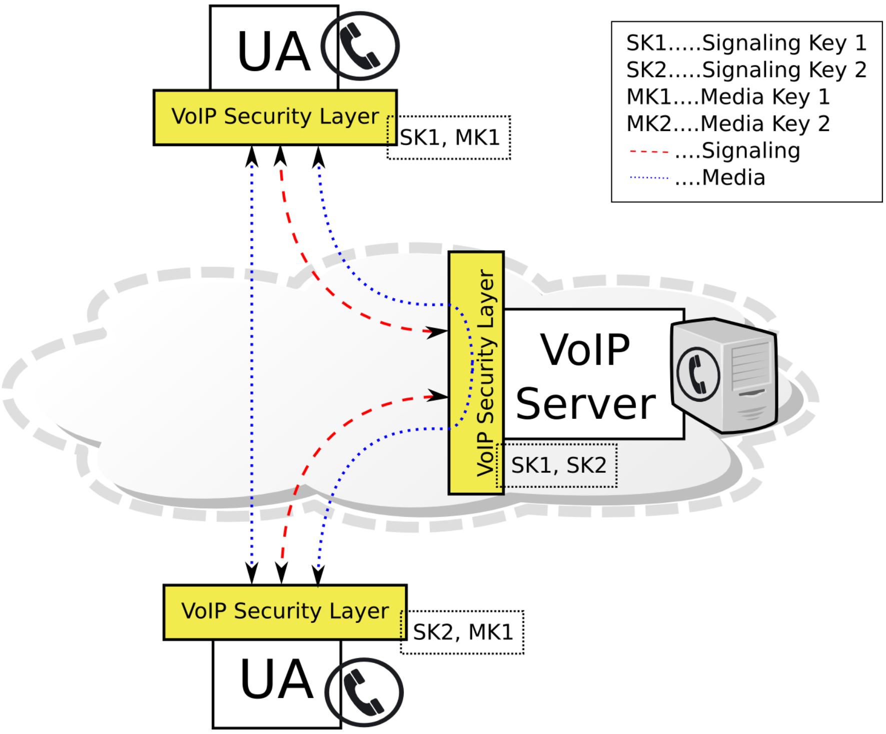


Figure 9.2: Communication paths of the participants via the VoIP security layer.

any communication that happened prior to the disclose of the long-term private keys cannot be decrypted or successfully crypto-analyzed by an untrusted third party.

Figure 9.2 illustrates the communication path of the data and the different encryption keys utilized, showing the necessity of all data being routed through the VSL before accessing the network. The VSL will forward data to the address requested by the VoIP protocols operating on the application layer, and therefore, is not required to distinguish between signaling and media data, although different encryption keys (*SK1, SK2, M1*) are used for both kinds of data. Due to the assumption that UAs only communicate with signaling or proxy servers and because media sessions can only be created with the support of signaling protocols, the signaling encryption key has to be known before a media session can be created.

When looking at the approach and the underlying VoIP system, the overall security gains can be described as follows:

- The authentication should not rely on potentially weak credentials, therefore we use cryptographic material for the authentication. It relies on the possession of the user's private key, and additionally requests known credentials.

- The system uses strong “transport” layer encryption, as we build on state-of-the-art algorithms and do not have to cover protocol legacies as in SSL/TLS implementations. The use of FHMQV [126], a high performance and efficient protocol, enables the establishment of secure communications including mechanisms to withstand active and passive attacks such as MitM and impersonation.

- The encryption of content data does not use deprecated or compromised algorithms and technologies, instead we use secure AES, which is considered safe by Schneier[1], Zimmermann[2] and Weiss[3]. But the encryption algorithm can be replaced, as soon as it is considered unsafe.

- Each VoIP packet is transferred encrypted, as our proposed solution does not contain unencrypted data at any time (except at the endpoints) after the key agreement phase to guarantee data integrity and confidentiality. Even by seizing a user’s private key, no information can be decrypted retrospectively, as PFS prevents compromising of the data’s confidentiality.

- For the implemented encryption scheme the data sovereignty remains with the communicating partners. Untrusted third-parties only have the possibility to capture the encrypted data, but no chance of decrypting the content.

VoIP Protocol Using the Security Layer

Figure 9.3 illustrates a possible communication flow in a closed VoIP system (where the VoIP Server, UA Alice and UA Bob are trusted). The red lines illustrate unencrypted messages and the black lines are encrypted messages. After the key-exchange phase only the data between the VoIP server to the VSL is unencrypted. Beginning with UA Alice that wants to contact the VoIP server, but does not have the signaling encryption key yet, the VSL implementation initiates the key exchange mechanism with the VoIP server’s security layer, resulting in both establishing a shared secret. Once the key is available, any delayed signaling data can be encrypted and transferred.

The information required to establish a shared secret between Alice and Bob is passed through the signaling protocol in order to have it transferred to the other UA. Through this exchange of

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

[2] https://www.philzimmermann.com

[3] http://cryptolabs.org/ruedi/

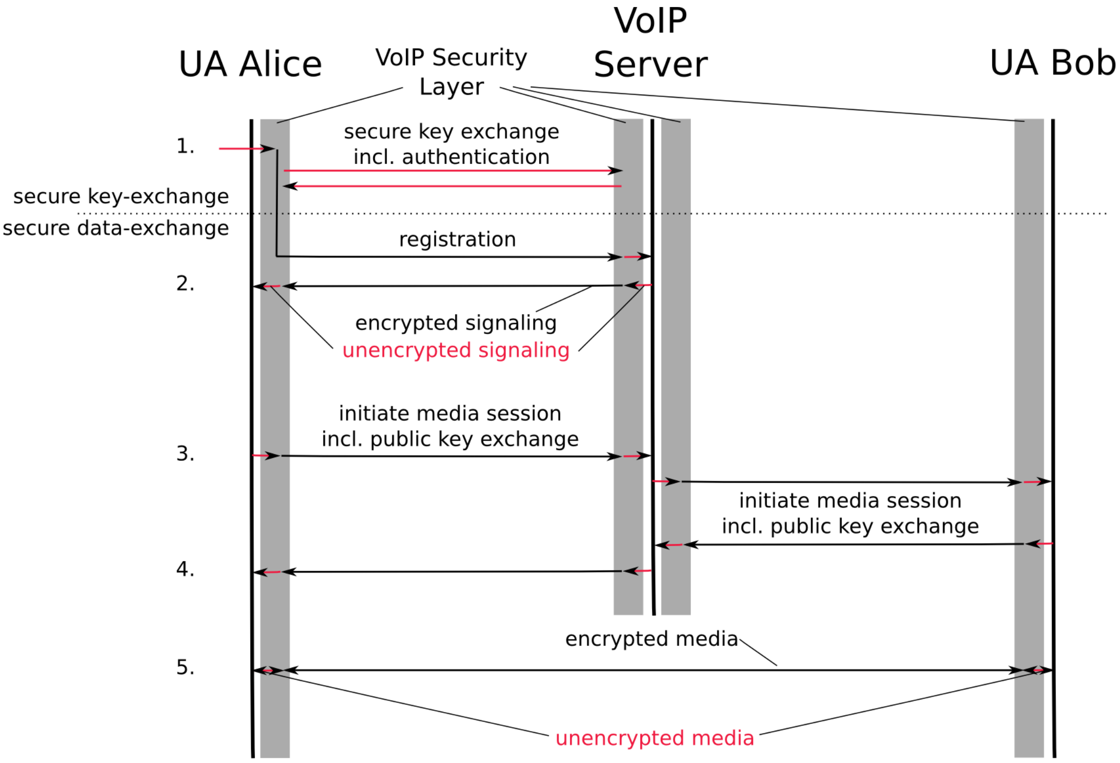


Figure 9.3: Communication flow of the proposed VoIP security layer approach.

data between the UAs, which is protected by signaling encryption, the media encryption key is derived. The key will remain valid until the session is finished.

A secure VoIP communication is established by following steps:

1. The signaling encryption key is created using a secure key exchange mechanism (we use FHMQV), initiated by the UA while enabling the UA to communicate with the signaling server until the signaling encryption key expires (e.g., after one hour), which requires the UA to trigger the key exchange mechanism, or the UA goes offline.

2. The conventional VoIP signaling is encrypted using the established cryptographic key (i.e., SIP messages).

3. The UA initiating a media session passes its ephemeral public key to the communicating partner, using the secured signaling connection. The static public key from the caller will be stored by the already trusted VoIP server in order to transfer only verified public keys. Both keys (ephemeral and static public) are required to generate a shared secret.

4. The call acknowledge message from the called UA carries the UAs ephemeral public key and the VoIP server adds the verified static public key, with which the shared secret, namely the media encryption key, can be generated. The information exchanged includes all required information in order to calculate a shared secret, which allows both communicating partners to implicitly authenticate each other. The media encryption key is discarded once the media session concerned has been finished.

5. Once the call starts the media stream is transmitted encrypted directly or via a RTP proxy between both UAs.

Figure 9.4 shows the optimized key distribution for the proposed key-agreement phase to derive a shared secret between the UA and the VoIP server.

The system allows the interconnection of more than one closed VoIP systems, by establishing a trusted relationship between the systems by exchanging their static public keys, as seen in Figure 9.5. The VoIP clients provide a transient trust relationship to each other without the need for additional verification using static public keys. The distribution and the management of the static public keys is the responsibility of the VoIP servers, and therefore can forgo an additional Certification Authority (CA). The approach provides a simple mechanism to extend the network to new members, e.g., external partners or project teams.

Cryptographic Protocol for VoIP Protection

To protect the communication between UAs and the VoIP server as well as between UAs from both passive and active attackers, we first establish a secure communication channel before we begin to exchange VoIP data (as seen in Figure 9.3). Our secure communication session consists of two phases:

- Shared key-agreement phase using asymmetric cryptography with the goal of easy key distribution and ensuring trust relationships for the communication.

- Secure data-exchange phase using symmetric cryptography with session keys with the goal of fast encryption with minimal overhead and protection of the data integrity.

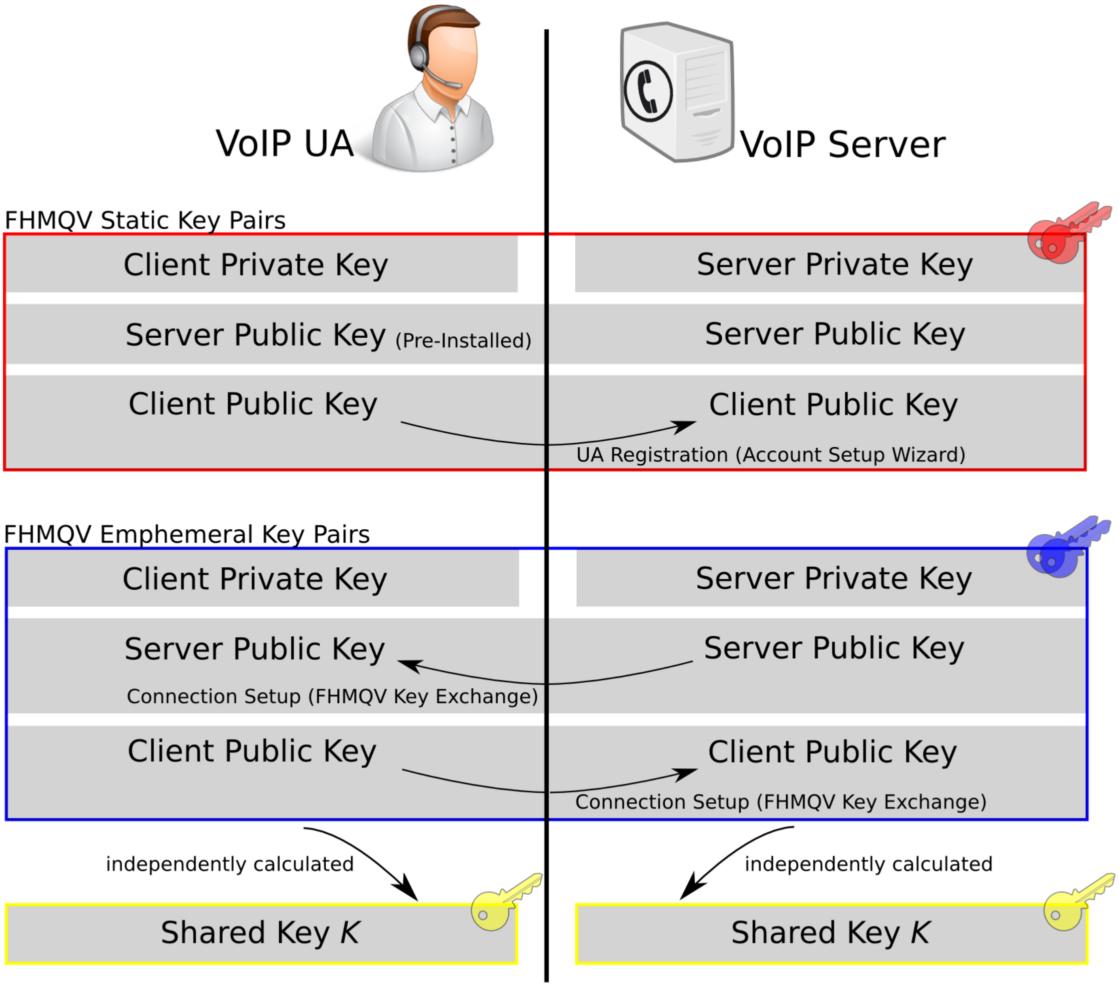


Figure 9.4: Key distribution of the secure VoIP communication approach.

Key-Agreement Phase

For the key-agreement phase, we use the elliptic curve FHMQV protocol [126]. The elliptic curve FHMQV key-agreement between two parties A and B over an insecure channel can be summarized as follows [126]:

- A has a static key-pair (S_a, s_a), where the public-key S_a is publicly known and sent to B.
- B has a static key-pair (S_b, s_b), where the public-key S_b is publicly known and sent to A.

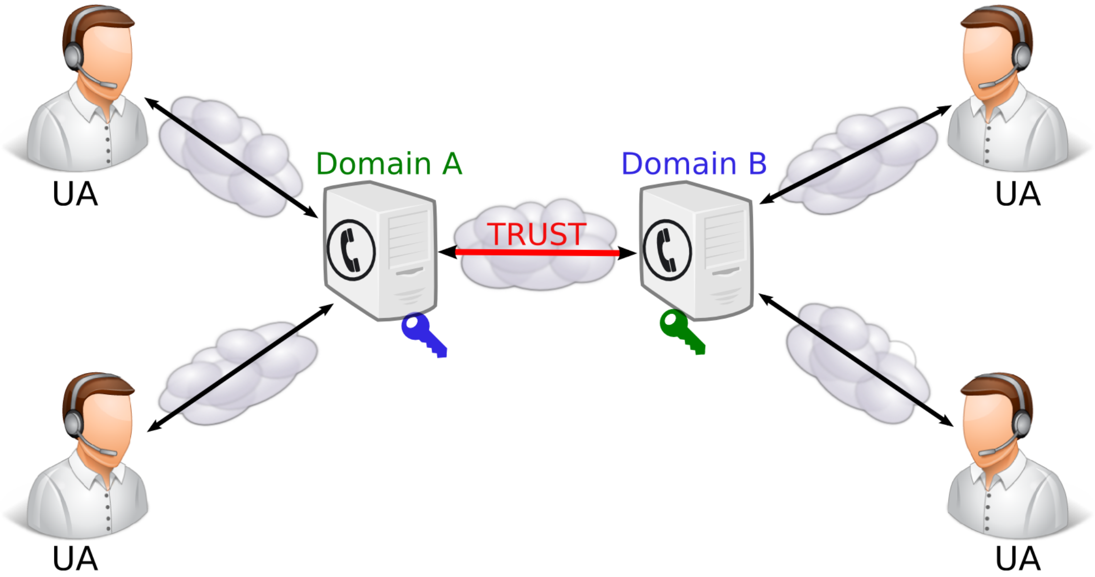


Figure 9.5: Interconnection of two secure VoIP domains.

1. A generates an ephemeral key-pair (E_a, e_a) and sends the ephemeral public-key E_a in a request to B to establish a secure channel using a shared key.
2. B also generates an ephemeral key-pair (E_b, e_b) and sends its ephemeral public-key E_b back to A in the reply.
3. Now, A calculates the shared key K using the parameters $(S_a, s_a, E_a, e_a, S_b, E_b)$.
4. Meanwhile, B has also calculated the shared key K using the parameters $(S_b, s_b, E_b, e_b, S_a, E_a)$.
5. To provide *forward secrecy*, both parties A and B delete their ephemeral key-pair (E_a, e_a) and (E_b, e_b), as soon as they are finished with calculation of the shared key K.

For our proof of concept implementation, we have chosen FHMQV [126] with a shared key K with a key-length of 256 bits and a separated key-id with a length of 32 bits. Each UA uses a different signaling key for communicating with the VoIP server. The key-id is transferred in an encrypted manner to the UA as last part of the key exchange mechanism, which is required later by the VoIP server to map incoming messages to different UAs. Consequently, the UAs do not have a shared secret which could be used to encrypt or decrypt other UAs signaling messages. Even if a third party eavesdrops on the whole communication, the shared key cannot be derived.

The right key can only be derived if the private key and the public key correspond to each other, so a MitM attack would fail.

The media stream will not be encrypted with the same shared key that is used for signaling messages, because the media content should not be readable by the VoIP server. Therefore, the caller and the callee derive their own shared key to encrypt the media content, exactly as it was done for the signaling messages with the VoIP server.

In order to prevent DoS attacks, the client has to calculate a client puzzle. The proof of work puzzle calculates SHA512 hashes from the first 16 bytes of the client's static public key, the first 16 byte of the client's ephemeral key and an 8 byte salt until a hash value is created, whose first 17 bits are zeros. The corresponding salt value is added to the initial key exchange request sent to the server. With the salt and both client keys, which are known by the server at this point, the proof of work can be verified. The additional computational overhead associated with the calculating this proof of work, makes the system unattractive for DoS attacks.

Secure Data-Exchange Phase

After the key-agreement phase, the two communicating parties are in possession of a shared key (for signaling and media transmission), so that we can now switch to a more efficient symmetric data encryption. An authenticated block cipher is used, namely the AES in GCM [94] with a key-length of 256 bit, which provides data integrity, authenticity/trustworthiness, confidentiality and privacy.

The GCM is a very efficient high-performance mode of operation for symmetric block ciphers and it also can be easily pipelined or parallelized to boost the performance. The input parameters for the AES-GCM are the plaintext T_p, the initialization vector IV and the additional authentication data AAD. The outputs are the encrypted ciphertext T_c and the authentication tag $ATAG$. The authentication tag $ATAG$ provides authentication of the transmitted messages.

The advantage of GCM compared to similar authenticated encryption methods is that it can be used without limitations (as it is patent free, contrary to OCB) and is rather simple (especially compared to CCM, which is known to be overly complex). AES-GCM will be used e.g., in upcoming versions of TLS too.

As Ferguson [45] described, it is important that the security of the AES-GCM depends on choosing unique initialization vectors IV when performing encryption with the same key. For our implementation, we have chosen a length of 128 bits for both the IV and the $ATAG$. The IVs are generated using a cryptographically secure pseudo-random number generator seeded by the shared secret. For each message (packet), a new IV is used. As AAD we use a key-id, which was also exchanged in the previous key-agreement phase, hence every key created uses its own distinct key-id. Consequently, every message sent within an active session will authenticate the key-id. If one cannot decrypt or authenticate the received encrypted message, the worst case is assumed: that a man in the middle is actively tampering with the messages and the session is aborted immediately.

Authentication of Server and Client

Our proposed solution relies on implicit authentication of the VoIP server as well as the UAs through the VSL. The VoIP server as well as the UA have to know the static public key of the communicating partner and their own private and public key pair (ephemeral and static). Only if the UA static public key is known by the VoIP server will the server communicate with the UA. If the UA static public key is not known to the server, the messages from the communicating partner will be rejected and not processed. Thus the system provides strong protection against application-level DoS attacks, therefore enhancing system availability. This procedure does come with some overhead, especially for transferring the UA public key to the VoIP server and leaves room for improvement, though we believe that the security benefits outweigh the drawbacks.

The UAs have to know the public key of the VoIP server to verify if the server is the trusted server and not a MitM. If the known public key and the delivered public key from the server are not the same, the UA terminates the communication to this server. The proposed approach helps to prevent MitM attacks and also simple VoIP attacks, e.g., brute-force or malformed VoIP message attacks, because the attacker is not able to communicate with the VoIP system without having the private key from a trusted user.

This approach works well in a companywide system where key distribution (i.e., static public key) can be based on other security mechanism already established in the company, e.g., protected email communication. After the initial distribution with our approach it is easy to extend the network by adding new users to the VoIP server. New VoIP servers can be added by trust relationships between servers. Additionally, users can always directly verify the fingerprint of the public key of the communicating parties, e.g., for increased privacy.

9.3 Design and Implementation of a VoIP Security Layer for SIP and RTP on Mobile Devices

To show the real-world applicability of the proposed solution, a reference system has been implemented using the in-house developed VSL combined with open-source software solutions, which were adapted to satisfy the system's requirements. The adjustments included simple application configurations and changes/additions to the application's source code. The VoIP system utilizes the SIP signaling protocol and the RTP media transport protocol.

Secure Solution Architecture and Components

The decision about which software to use was based on its feature set and adaptability to our requirements, and how additional features (that do not have an direct effect on the implementation) are supported.

Figure 9.6 shows the software components used in the reference system as well as their communication setup. The system deploys the following components:

- **VoIP server**

  The VoIP server Kamailio[4] was used, because it's highly flexible and fully supports the use of client side ICE and enables end-to-end (UA-to-UA) transfer of customized SDP parameters. Kamailio version 4.0.4 without source code modifications was operated. It runs only on network ports which are not reachable from an external network.

  The VoIP server Asterisk[5] was also tested. Though operating properly with our security layer implementation, it was excluded due to shortcomings in the ICE implementation for NAT traversal.

- **Secure Proxy**

  The server side implementation of VSL is independent of the VoIP server Kamailio. It is reachable from an external network and acts as transparent proxy, relaying messages to and from the only locally reachable VoIP server Kamailio. This module is in charge of encrypting every VoIP communication with the outside world, whereas the communication with Kamailio is unencrypted. Therefore, this component should be located in the same security zone as the VoIP server in the data center.

[4] `http://kamailio.org`

[5] `http://asterisk.org`

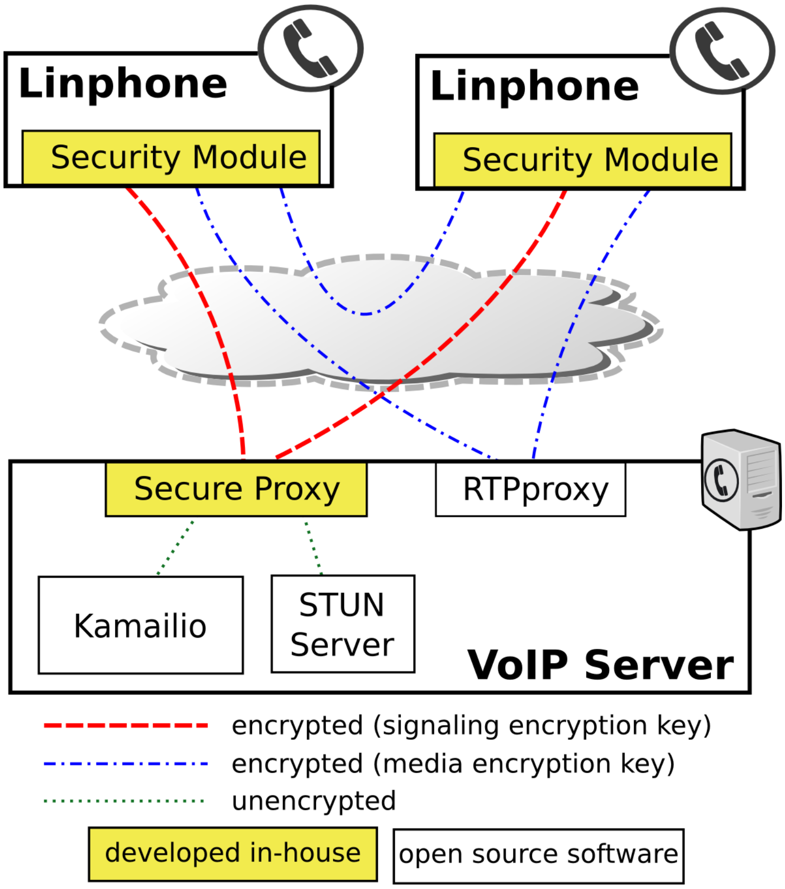


Figure 9.6: Secure VoIP solution components and traffic flow for the proof of concept.

- **RTP media relay server**

  To enable VoIP functionality when a NAT is involved, a media relay server becomes necessary. This requires basic media relay software which does not have to understand the data it is relaying. This allows the passage of encrypted data from UA to UA, without the need to implement the VSL for the media relay server. The media relay server is used in case ICE is not able to find a direct communication path between UAs. An unmodified RTPproxy[6] is used for this purpose.

- **STUN server**

  To allow the use of the ICE functionality, a dedicated STUN server is required. STUN server Stuntman[7] was used for this purpose. STUN messages carry and provide informa-

[6] http://rtpproxy.org

[7] http://stunprotocol.org

tion about a UA's availability on the network. Because the same communicating partners are involved, requests to this component are encrypted with the signaling encryption key. STUN requests exchanged between both UAs during ICE setup are encrypted with the according media encryption key, which was negotiated before the media transport.

- **User Agent**

  The open-source UA solution Linphone[8] was chosen due to its availability on multiple platforms. Its core is implemented in native C/C++ and provides hooks to attach a "tunnel" extension module. This module is proprietary software and was not available to us. Therefore, our proposed and implemented security layer was integrated with the Linphone as an additional software module. Besides a few minor changes, for example to allow interactions for security reasons with the Graphical User Interface (GUI), the application was not modified.

  The module was written in C++, whereas cryptographic functions were provided by Crypto++ library[9]. By using this approach it is possible to route every packet of network data that originates from or is addressed to the application through the newly created module. This way security layer functionality has been introduced without the need to have additional software installed on UA devices. Although Linphone supports multiple platforms, our implementation utilizes Linphone's mobile Android version for this proof of concept.

UA Setup Procedure

To be able to establish a shared secret between a UA and a VoIP server, additional information such as the other's static public key, which represents the public portion of the non-ephemeral certificate required by the FHMQV scheme, must be known. The VoIP server's static public key is shipped with the UA application. Due to the requirement that the public key of a UA needs to be known by the security layer implementation of the VoIP server before they can communicate with each other, a UA account setup procedure was introduced (see Figure 9.7).

1. When the UA application is first started or after the application's private data has been deleted, a static client key pair is generated and displayed in the "Account Setup Wizard" (see Figure 9.8).

[8] http://linphone.org

[9] http://cryptopp.com

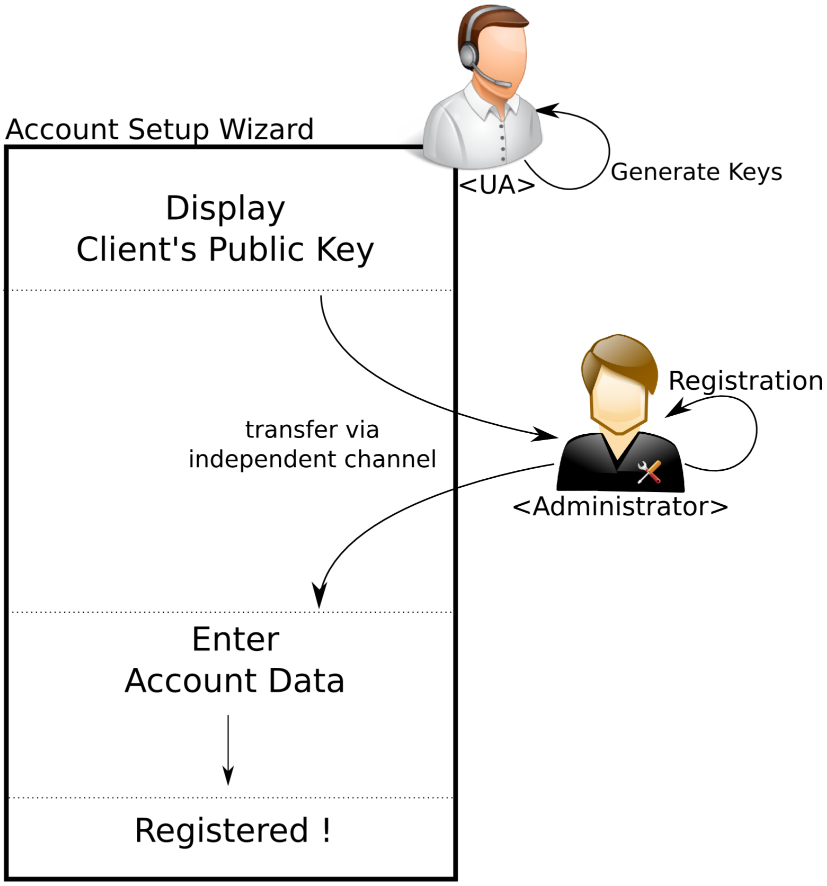
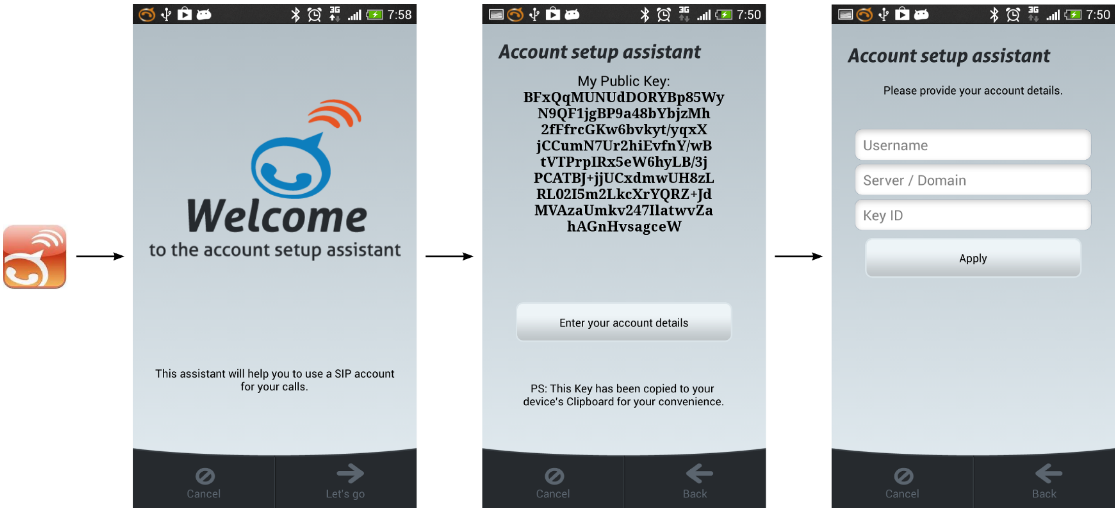


Figure 9.7: Registration setup of a UA using the VoIP security layer.

2. It is up to the user to transfer this key to the administrator of the VoIP system.
3. Consequently, the administrator replies with login data including username (which is only a pseudonym with no connection to the real name), server address and KID, which can be entered at any time.
4. On the following screen of the "Account Setup Wizard", the user enters the received details to finish the account setup.

With this approach, the UA's static public key have to be stored on the secure proxy server, and the UA must be informed about the KID to use before any direct communication between both partners takes place. To transfer this information, a separate communication channel (independent of the proposed VoIP solution) has to be used.



Welcome

to the account setup assistant

This assistant will help you to use a SIP account for your calls.

Account setup assistant

My Public Key:
BFxQqMUNUdDORYBp85Wy
N9QF1jgBP9a48bYbjzMh
2fFfrcGKw6bvkyt/yqxX
jCCumN7Ur2hiEvfnY/wB
tVTPrpIRx5eW6hyLB/3j
PCATBJ+jjUCxdmwUH8zL
RL02I5m2LkcXrYQRZ+Jd
MVAzaUmkv247IlatwvZa
hAGnHvsagceW

PS: This Key has been copied to your device's Clipboard for your convenience.

Account setup assistant

Please provide your account details.

Figure 9.8: First step of the account setup of the mobile App for using the VoIP security layer.

Security Layer Implementation

The VSL features ephemeral key exchange and encryption capabilities. It distinguishes between SIP and RTP messages by identifying the function invoking a security layer feature, which is necessary to be able to choose the correct encryption key (signaling or media encryption key).

Key Exchange

The proposed key exchange mechanism utilizes, contrary to all other communication of this system, TCP for data transmission. A separate port of the VoIP server is used, which implements HTTP headers. To have the payload structured properly while supporting simple assembly/disassembly, the payload is formatted with JavaScript Object Notation (JSON). TCP with its initial handshake verifies whether the VoIP server to be contacted is responding before any information used for the key exchange is transmitted.

The UA that initiates the key exchange phase generates an ephemeral FHMQV key pair before the signaling procedure of VoIP begins, if no key for this dedicated communicating partner is yet known. Along with the KID, the proof of work from the client puzzle, the public key of this newly created key pair is included in the key exchange request. The respective ephemeral public key of the communicating partner is received in the key exchange response. When both communicating partners know the other's ephemeral public key and have the static public key from the communicating partner (which have been exchanged already), the shared secret can be calculated.

Encryption/Decryption

Figure 9.9 shows the structure of an encrypted data packet, created by encryption function of our security layer implementation. This structure is tailored for usage with our security layer implementation, making it light-weight and reducing unnecessary overhead. Consequently, bandwidth and resource requirements can be kept low. It consists of:

- **KID** (Key Identifier) - 32 bit: Used for key identification on the VoIP server. The KID to be used is negotiated during the key exchange. Content of this field is used as *AAD* during AES encryption.
- **IV** (Initialization Vector) - 128 bit: Cryptographic information used to encrypt this packet's data.
- **Chk** (Check) - 128 bit: Information used to authenticate the contents of the encrypted packet (AES *ATAG* information).
- **DL** (Data Length) - 16 bit: Specifies the size of the payload (bytes).
- **Dta** (Data) - dynamic length: Payload provided by or addressed to the VoIP server or UA application.
- **Rnd** (Random Data) - dynamic length: Additional random data (e.g., padding) to prevent an attacker from knowing the length of the plaintext, making attacks based on packet size analysis more ineffective. This field size is dynamic and is between the AES block size and twice the AES block size, not exceeding the total allowed packet size.

As the fields "KID", "IV" and "Chk" provide data, which the communicating partner requires to be able to decrypt properly, those three fields are not part of the encrypted data. Both, SIP and RTP utilize UDP for network transmissions, the encryption/decryption functions are only implemented for use with UDP, although they do not support UDP fragmentation. Due to packet size restrictions of the transport network media (Maximum Transmission Unit (MTU) size), the allowed maximum packet size of 1400 bytes has been defined. To reduce the packet size, the payload data is optionally compressed using gzip [33] before it is encrypted, but only if compressed data is smaller than the original payload data. Especially for small data packets, the

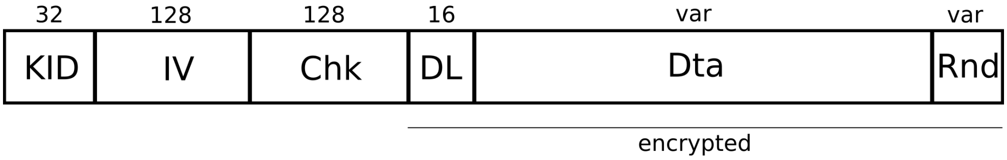


Figure 9.9: Structure of the encrypted data packet by the VoIP security layer.

gzip headers may cause the compressed packet to become bigger. Compression related security concerns can be discarded due to the subsequent encryption of the data, as similar approaches already discussed [92, 144].

9.4 Conclusion

We propose an approach for securing VoIP communications by installing an additional security layer to existing VoIP components. This generic concept can be applied to different VoIP solutions with different UAs (e.g., soft phones or mobile phones) and is independent of the respective VoIP implementation. The approach is based on trusted clients and VoIP Server and is therefore designed for closed systems like company networks.

The aim of the approach was to develop a system using existing cryptographic primitives and focus on making it highly secure, while maintaining an acceptable quality of service for phone calls by using high performance cryptography mechanisms. In the signaling case, the UA and the VoIP server know the key for encryption and decryption of the messages. In the case of communication data only the caller and the callee know the key, and not even the VoIP server is able to decrypt the communication packets. Thus, in the case of a passive eavesdropping VoIP server only the signaling data can be interpreted, but not the communication data.

The approach gives back the control of phone communication to its parties and therefore ensures the privacy of phone calls on the Internet. This is very important for critical communications, e.g., to protect company information against industrial espionage.

CHAPTER 10

Evaluation and Discussion of the Security and Voice Quality Aspects of the Proposed Solution

10.1 Introduction

For VoIP systems it is essential that the effect of increased security on the main use case of talking on the phone should be minimized. Enhanced cryptographic algorithms which are resource intensive are not suitable for VoIP systems, especially for mobile devices.

Delay is one of the most critical quality performance measurements and it was analyzed in order to identify the effect of the presented security layer on network operation and quality of service.

The analysis presented compares the VSL handshake with other secure transmission approaches (such as SSL). It also compares the reference system running with the security layer and without the security layer.

10.2 Evaluation of the Security Measures Covered by the VoIP Security Layer

The transparent security layer for VoIP systems enables secure communication and offers protection against the identified threats as described in Chapter 8 (see Table 8.1). Table 10.1 presents the residual risk level of the identified threats, after the effectiveness evaluation of the implemented requirements (as defined in Chapter 8).

	REQ-1	REQ-2	REQ-3	REQ-4	REQ-5	Residual risk level
Threat-1	-	X	-	-	-	Low
Threat-2	-	-	X	-	-	Low
Threat-3	X	-	-	-	-	Low
Threat-4	-	-	-	X	-	Low
Threat-5	-	-	-	-	X	Low
Threat-6	-	-	X	-	-	Low

Table 10.1: Correlation of the threat classes and the introduced requirement classes including the residual risk level.

- **Threat-1 (Service abuse – identity theft)**: Weak password authentication was replaced with strong key-based authentication, which is also feasible on mobile devices (as shown in Chapter 9).
- **Threat-2 (Malformed messages)**: The threat class for malformed VoIP messages was reduced, because the VoIP server receives data only after successful key-establishment. Although the key-agreement of the VSL can also receive malformed messages, the complexity of the key-exchange protocol is much lower than the SIP/RTP protocol and the sender also has to complete the cryptographic proof of work. This mechanism reduces the risk level to a lower level.
- **Threat-3 (Eavesdropping and interception)**: The risk of interception by a non-sophisticated intelligence service was reduced to a lower level through the use of a secure channel with end-to-end encryption, a key-based authentication mechanism for MitM protection and PFS support.
- **Threat-4 (Service abuse – fraudulent calls)**: After an identity theft attack, an attacker may make fraudulent calls. The protection from Threat-1 also reduces the risk level of this attack too.
- **Threat-5 (Modification)**: The authenticated secure channel reduces this risk level to a lower level, because MitM is not possible with our implementation.
- **Threat-6 (DoS/Service abuse – Malicious messages)**: The risk of valid but malicious VoIP messages from an untrusted sender was reduced, because the VoIP server receives data only after successful key-establishment.

The VSL provides confidentiality, integrity, authenticity/trustworthiness and privacy to all communication parties in a closed VoIP system. It does this through the use of a secure channel

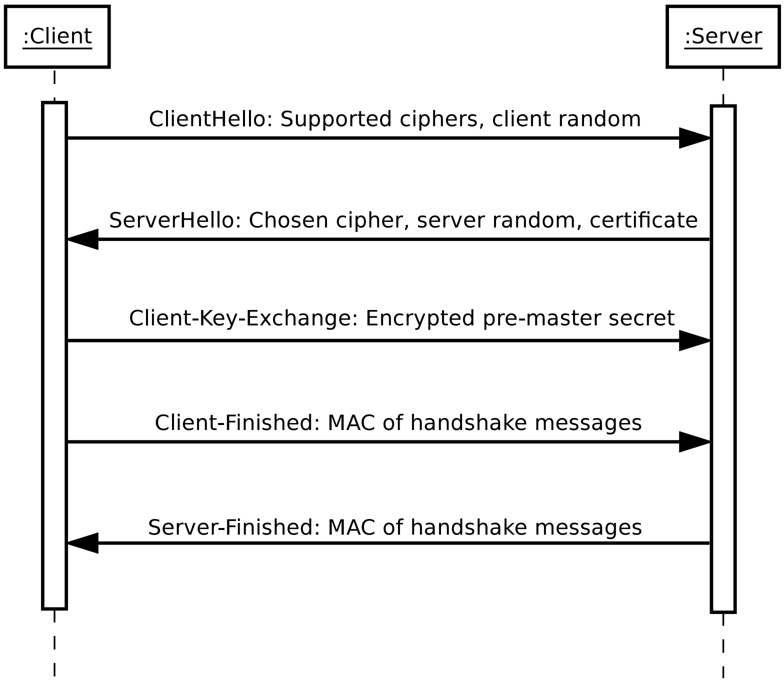

:Client
:Server
ClientHello: Supported ciphers, client random
ServerHello: Chosen cipher, server random, certificate
Client-Key-Exchange: Encrypted pre-master secret
Client-Finished: MAC of handshake messages
Server-Finished: MAC of handshake messages

Figure 10.1: Sequence diagram of a common TLS handshake based on RFC 5246 [34].

with end-to-end encryption and MitM protection using a key-based authentication mechanism. The proposed countermeasures from Chapter 9 reduce the risks of VoIP systems to level "Low". Therefore all risks are at an acceptable level and no additional steps are necessary.

10.3 Security and Quality of the VoIP Security Layer

To show the usability and the efficiency of the VSL, various evaluations were carried out based on the proof of concept implementation. SIP as well as RTP can use other protection mechanisms (e.g., TLS [34] as a transport protocol) to provide confidentiality, integrity, or authenticity. In addition to the weaknesses of TLS as described in Chapter 3, the use of TLS costs throughput and is not suitable for low-bandwidth VoIP communications (such as 3G networks). Shen *et al.* [137] showed that the baseline UDP performance is between three times (in the worst case) and 17 times (in the best case) faster than when using TLS.

A major advantage of our proposed security layer is the reduction in the number of requests in the handshake phase to save resources. Usually a TLS handshake needs five messages between the server and the client, as presented in Figure 10.1.

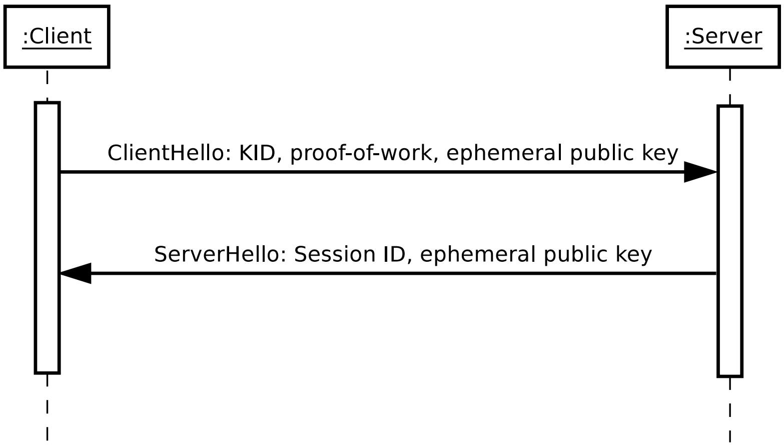

:Client
:Server
ClientHello: KID, proof-of-work, ephemeral public key
ServerHello: Session ID, ephemeral public key

Figure 10.2: Sequence diagram of the VoIP security layer handshake.

The following steps are involved in a TLS handshake as described in RFC 5246 [34]:

- The client and the server exchange “hello” messages to agree on algorithms, exchange random values, and check for session resumption.
- They exchange the necessary cryptographic parameters for the derivation of a *pre-master* secret.
- They exchange information (i.e., certificates and cryptographic information) for authentication.
- Both generate a master secret from the *pre-master* secret.
- Allow the client and server to verify that the handshake occurred without tampering (via a MAC) by an attacker.

As presented in Figure 10.2, the VSL approach needs only two messages between the server and the client and also does not need the TCP handshake (because we use UDP). For the use of the VoIP system in low-bandwidth networks, each extra request reduces the quality of the service. The VSL can use TCP or UDP as a transport layer depending on needs of the application. For the media transmission UDP is preferred for performance reasons.

The following steps are involved in the VSL handshake for building a secure channel:

- The client generates the ephemeral keys for the FHMQV key exchange.

- The client sends the key id, the proof of work, and its own ephemeral public key to the server.
- The server verifies the client, generates a unique session ID and its own ephemeral keys.
- The server sends the session ID and its own ephemeral public key to the client.
- Both derive the shared secret via the FHMQV protocol.

If the media stream is established directly between the UAs (without an RTP proxy), as is the case in conventional VoIP systems, it is easy for an eavesdropper to identify which parties (IP addresses) are communicating. A mandatory use of an RTP proxy may help to better protect anonymity, because all VoIP packets have to go through the RTP proxy, and no direct communications are visible. In case of many simultaneous VoIP communications it is more difficult to identify the communicating parties without comprehensive network analysis. Additionally, using inconsequential pseudonyms can be used to hide real identities by making it harder to map users to communications on the application layer.

The time required to transfer data packets from one UA to another UA is essential to the quality of the communication that the system is able to provide. Executing cryptographic functions always requires a certain amount of processing power, which is especially limited on mobile devices. It takes time until a data packet is ready to be sent out on the network or is ready to be processed by the application after being received from the network. This increases the total processing time and consequently, increases the total transfer delay between UAs.

Impact of the Packet Size

Data packet size is important because the amount of data transferred on the network can influence the quality of network operation. Additional information and headers introduced by security layer functions may increase the size of the data packets on the network. To validate the usability of the VSL, the packet size of the protocols (including different voice codecs) and the throughput of the VSL on a mobile device were measured.

Figure 10.3 shows the average size of the signaling packets with security layer functions enabled, with TLS (Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA and a self-signed certificate) as the transport protocol and finally without any protection mechanisms. It illustrates that the size of signaling packets is decreased by the use of gzip compression as part of the VSL encapsulation process. As well as the higher number of TLS requests, the packet size is also larger when

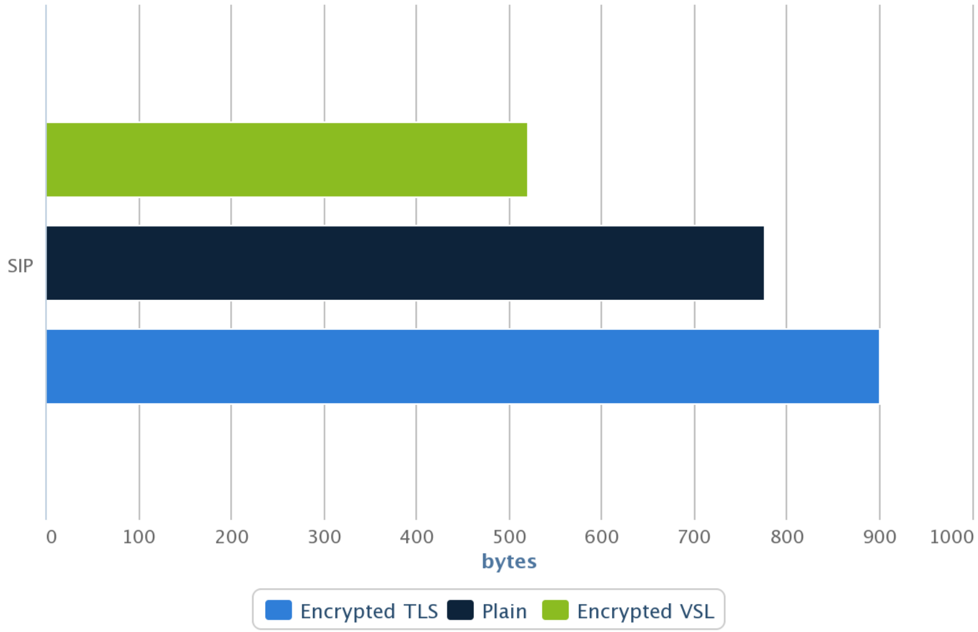


Figure 10.3: Average packet size of the encrypted and plain SIP messages.

using TLS compared to the VSL. Therefore, the VSL is better suited than TLS to low-bandwidth networks.

The VSL adds between 70 and 102 bytes (uncompressed) to each packet. This could be decreased by reducing the random number of padding bytes (between 32 and 64) for security reasons. Therefore, further work on optimizing the packet size would be useful.

The unencrypted and encrypted packet size of the media transmission including various voice codecs as measured by the mobile App are presented in Figure 10.4. Through the additional bytes for security purposes, it is essential to use voice codecs with a small overall packet size to get the same quality of service. ITU G.722 with a high MOS value needs more bytes than for example Global System for Mobile Communications (GSM) or Speex[1] 8 KHz. The traditional ITU G.711 codec uses no compression (different to the others) and therefore the additional security bytes are not noticeable, because the reduction of the bytes through the compression is predominant.

Figure 10.5 presents the throughput of the VSL of a common VoIP communication with a length of approximately 110 seconds at the UA (i.e., the mobile device). The effective bit rate is only

[1] http://www.speex.org/

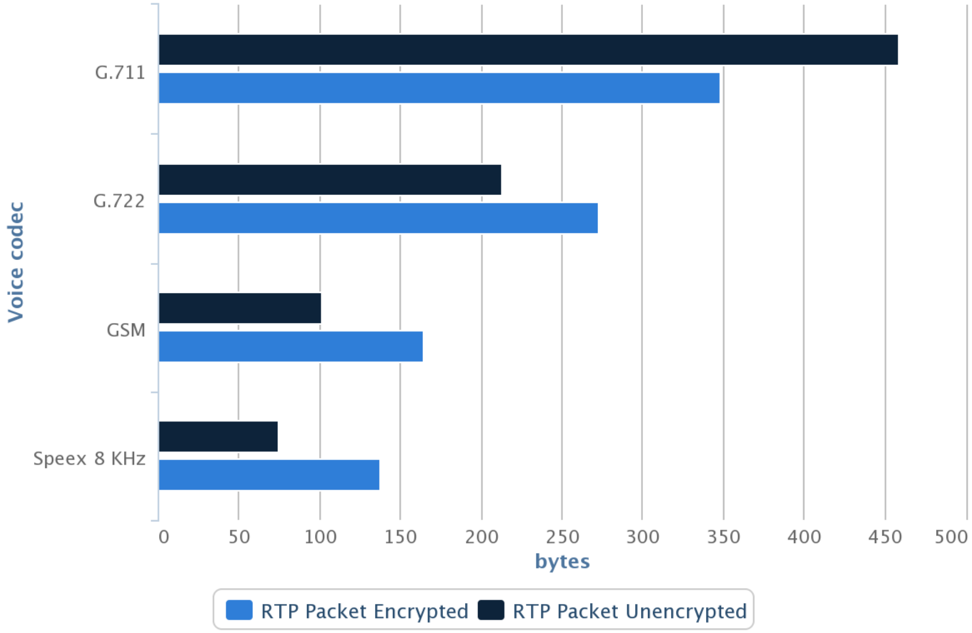


Figure 10.4: Average packet size of encrypted RTP packets by the use of different voice codecs.

better for ITU G.711, because of the reduction in bytes due to the compression. For all other codecs the security layer adds additional bytes for security. Especially in case of operating under bad network conditions with high network latency and packet loss, choosing a media codec with lower bandwidth requirements may partially make up for potential quality degradation.

Impact of the Additional Cryptographic Operations

The time required to perform the encryption and decryption of the communication streams has been determined by using the timing functionality within the security layer implementation of the UA application. Figure 10.6 shows the average time requirement for encryption as well as decryption operations on five Android based devices, namely a Samsung Galaxy R (GT-I9103), a Samsung Galaxy Tab 2 10.1 (GT-P5110), a HTC One X, a Samsung Galaxy S4 Mini (GT-I9195) and a Sony Experia Z. The average time for processing one packet of data recorded from multiple measures is shown, for RTP packets only.

The ITU Telecommunication Standardization Sector (ITU-T) describes in the specification G.114 [73] that the recommended one-way overall delay for voice should not be more than 150 ms. For a private network 200 ms is a reasonable goal and 250 ms must be the maximum. Otherwise

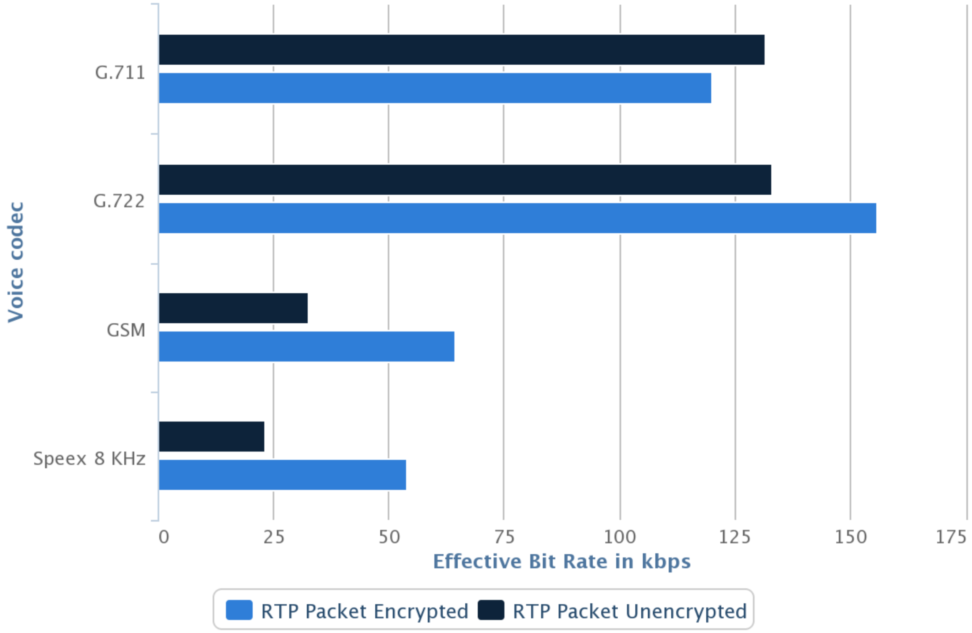


Figure 10.5: Effective bit rate of a typical VoIP communication through the VoIP security layer on a mobile device.

the voice quality is recognizable worse. This should be considered if additional security mechanisms are to become established. Figure 10.7 presents the impact of the maximum encryption and decryption delay (as seen in Figure 10.6) in comparison to ITU-T recommended one-way overall delay for voice, which should not be more than 150 ms. In terms of the allowed delay, the time required for encryption and decryption is negligible, because this is less than five percent of the overall delay. The benefits of VSL concerning confidentiality, integrity, availability, authenticity/trustworthiness and privacy outweigh the additional delay.

Although these figures may be influenced by various factors (and may be different on other devices) and considering that a typical network delay can be multiple hundreds of milliseconds, the impact of the introduced cryptographic functions on the total transfer time requirements and subsequently on communication quality is very low. Despite significantly higher bandwidth requirements for the transfer of RTP data, these figures indicate that the additional security measures introduced only have little effect on overall QoS of the underlying VoIP system.

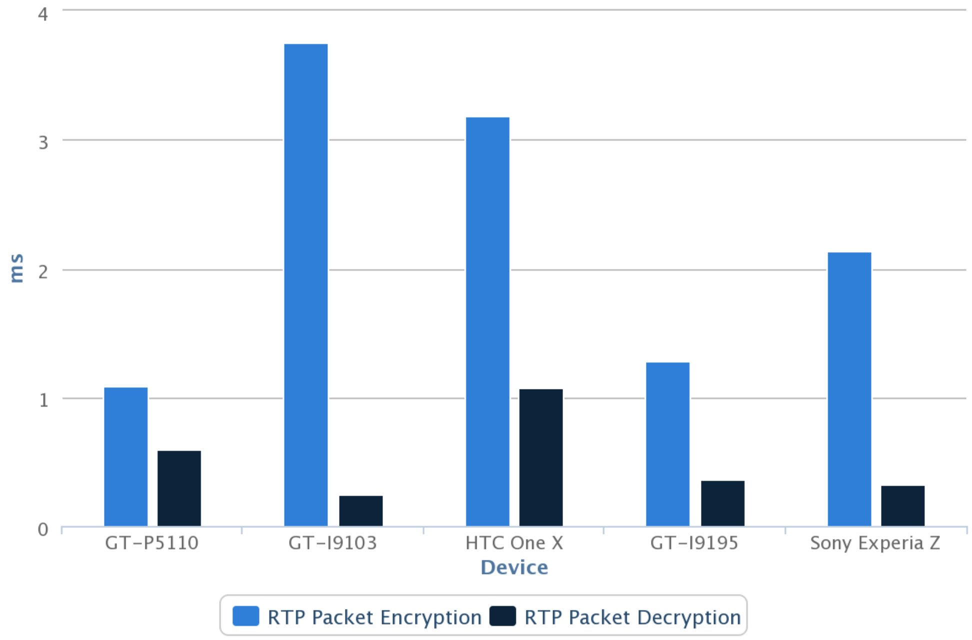


Figure 10.6: Average encryption and decryption delay of the VoIP security layer.

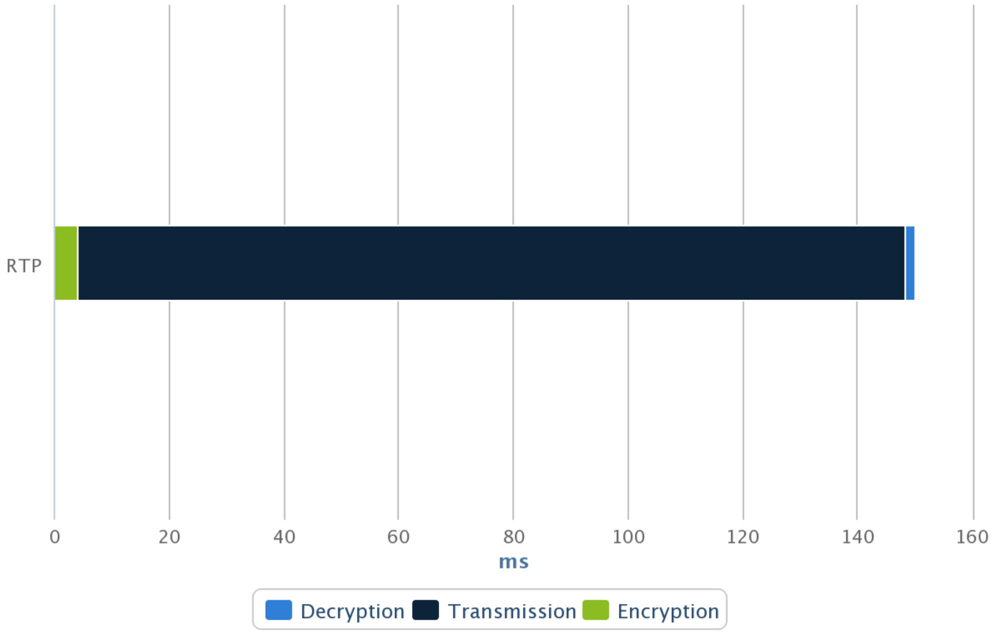


Figure 10.7: Impact of encryption and decryption delay in comparison to the ITU-T recommended one-way overall delay.

10.4 Video and Chat Support

The VSL operates between the application layer and the network layer of the IP stack, which means that there is no restriction on the SIP features supported. The signaling and media layers are encrypted irrespective of the type of messages or media that they carry. This means that in addition to securing voice calls, the VSL can also secure SIP-based videos calls and instant messaging. This additional benefit of the VSL makes it more attractive as a complete secure communications solution for organizations or companies.

10.5 Conclusion

The evaluation of the countermeasures showed that the introduction of the VSL lowered the risk levels for the VoIP systems to an acceptable level. The proof of concept implementation and the QoS analysis demonstrated that the presented approach works and can be used with acceptable voice quality. The approach can be used for closed user groups to ensure secure voice communication using mobile devices over weak network connections (due to the small packet sizes and the optimized protocol) and provides protection against the main identified VoIP attacks.

Moreover, the VSL seems to be an appropriate approach for establishing a secure channel for VoIP solutions also in low-bandwidth networks. This makes the VSL a viable option for mobile phones. The reduction of complexity (e.g., in comparison to the legacy TLS protocol) helps to increase the security of the proposed approach. Schneier [133] described this as: “Complexity is the worst enemy of security”.

CHAPTER 11

Conclusion and Future Research

11.1 Findings of the Thesis

This thesis used a common four step security approach to protect VoIP systems against current threats and vulnerabilities and to enable secure communication:

1. Three VoIP honeynet solutions were operated at different locations to capture the data of real-world attacks over a long evaluation period.
2. The captured data was analyzed to get detailed information about threats and vulnerabilities of current VoIP systems. A risk assessment was carried out based on the potential damage of the threats and the likelihood of their occurrence.
3. Based on the major risks, countermeasures were designed and implemented in a transparent VoIP security layer.
4. The effectiveness of the implemented countermeasures was examined using an implementation of the security layer for mobile devices.

Efficient attack collection

The first part of this thesis presented an extensible and flexible honeynet solution (including an analyzing engine) for capturing attacks, independent of the protocol used. This approach can successfully collect information about attacks without mixing productive and attack data. As part of this work the honeynets were configured to automatically capture known and unknown attacks against the VoIP honeypots. Two VoIP specific honeynet solutions and one PSTN specific VoIP honeynet solution were operated to capture attacks.

During the long period of operation, the honeynets accumulated a lot of data, which made automatic analysis of the captured data a prerequisite for investigating the attacks. A dedicated VoIP attack analyzing engine was designed and implemented to perform this analysis. To obtain details about the attacks, the engine used properties of the IP stack, the signaling information and the transmitted media from the captured data. Additionally various external sources (such as DNS or geolocation services) were used to improve the information content of the results and to provide more details about the attacks and the attackers' behavior.

The dynamic structure and flexibility of the VoIP analyzing engine allowed customizable analysis and reporting to be automated, e.g., monthly reports of the attacks performed or daily reports of the IP addresses used.

Analysis of VoIP attacks

The second part of this thesis described the analysis and evaluation of the captured real-world attacks provided by the VoIP attack analyzing engine. An attacker model for fraudulent calls was uncovered and the proposed business model behind these attacks was presented.

The main threats to VoIP systems were identified as:

- **Signaling threats**: Brute-force takeover of an existing account for identity theft, malformed messages with the intention of service abuse or denial of service, and finally interception/modification of the plain signaling messages (i.e., third-parties are able to identify who is talking to whom, etc.).
- **Media transmission threats**: Interception of the communication by compromised providers (eavesdropping) and fraudulent calls with the intention of avoiding costs for chargeable calls to PSTN systems.

A risk assessment was carried out based on an analysis of the potential damage posed by a threat and the likelihood of it occurring. It identified the following high level risks: *identity theft*, *eavesdropping and interception*, and *fraudulent calls*.

Improving the Security of VoIP systems

Using the major risks as a reference, countermeasures were designed and implemented to secure VoIP systems. These measures provided confidentiality, integrity, availability, authenticity/trustworthiness and privacy for VoIP communications in closed systems (such as companies).

A non-invasive security layer (called VSL) was designed and implemented for existing VoIP protocols (i.e., SIP and RTP). The VSL uses an authenticated secure channel for the transmission of signaling and media data. Through the use of a lightweight handshake (including MitM protection and PFS based on FHMQV), the VSL provides secure and non-traceable VoIP communications.

The security layer was especially designed to be suitable for mobile devices. The implementation demonstrated that the VSL is indeed usable in low-bandwidth networks with little degradation of voice quality. Although, the VSL needed some additional bytes for security reasons, the increased packet size may be compensated for by using low bandwidth voice codecs.

An evaluation of the implementation indicated that a balance had been struck between security and usability, which would be acceptable to the general public. Thus the transparent VoIP security layer gives control of the voice data back to the communicating parties and ensures the privacy of phone calls on the Internet.

11.2 Possible Future Research

As a part of the future work, the honeypots could be made more attractive in order to trap more sophisticated attackers. For example, the use of an additional gateway from the honeynet to another external system or the use of other VoIP servers (maybe with known vulnerabilities) would make it more attractive to attackers.

Ongoing research on the security status of VoIP systems is needed, because real-world attacks against VoIP systems are still evolving. An automated derivation or adaptation of the protection mechanisms would be an effective countermeasure against attackers.

For the VSL approach presented, optimization of packet sizes on the media channel is required to decrease network based delays. Additionally, a thorough analysis of possible side channel attacks against the approach is required, to ensure protection against additional privacy violations (e.g., comprehensive network analysis to identify communication partners).

With the right future work, the transparent security layer could be integrated into existing VoIP systems to provide a previously unavailable level of security for Internet telephony.

Bibliography

[1] G. Aghila and D. Chandirasekaran. An Analysis of VoIP Secure Key Exchange Protocols Against Man-In-The-Middle Attack. *International Journal of Computer Applications*, 33(7):46–52, 2011.

[2] H. Al-Allouni, A. E. Rohiem, M. Hashem, A. El-moghazy, and A. E.-A. Ahmed. VoIP Denial of Service Attacks Classification and Implementation. In *National Radio Science Conference (NRSC)*, pages 1–12, 2009.

[3] N. Al Fardan and K. Paterson. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *IEEE Symposium on Security and Privacy (SP)*, pages 526–540, 2013.

[4] F. Andreasen, M. Baugher, and E. Wing. RFC 4568 – Session Description Protocol (SDP) Security Descriptions for Media Streams, 2006.

[5] D. F. Aranha, P. S. L. M. Barreto, C. C. F. P. Geovandro, and J. E. Ricardini. A Note on High-Security General-Purpose Elliptic Curves. *IACR Cryptology ePrint Archive*, 2013.

[6] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and N. K. RFC 3830 – MIKEY: Multimedia Internet KEYing, 2004.

[7] A. Badach. *Voice over IP - die Technik: Grundlagen, Protokolle, Anwendungen, Migration, Sicherheit*. Carl Hanser Verlag, 2007.

[8] G. V. Bard. A Challenging But Feasible Blockwise-Adaptive Chosen-Plaintext Attack on SSL. In *Proceedings of the international conference on security and cryptography (SECRYPT)*, pages 7–10, 2006.

[9] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for Key Management - Part 1: General (Revision 3). In *NIST Special Publication 800-57*, 2012.

[10] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. RFC 3711 – The Secure Real-time Transport Protocol (SRTP), 2004.

[11] M. Bedner and T. Ackermann. Schutzziele der IT-Sicherheit. *Datenschutz und Datensicherheit (DuD)*, 34:323–328, 2010.

[12] M. Bellare and C. Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer Berlin Heidelberg, 2000.

[13] D. J. Bernstein. Curve25519: New Diffie-Hellman Speed Records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer Berlin Heidelberg, 2006.

[14] D. J. Bernstein and T. Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. `http://bench.cr.yp.to/`. (Last accessed: February 13, 2014).

[15] D. J. Bernstein and T. Lange. SafeCurves: Choosing Safe Curves for Elliptic-Curve Cryptography. `http://safecurves.cr.yp.to/`. (Last accessed: February 12, 2014).

[16] M. Bishop. *Introduction to Computer Security*. Pearson Education, Inc, 2003.

[17] S. Blake-Wilson and A. Menezes. Authenticated Diffie-Hellman Key Agreement Protocols. In S. E. Tavares and H. Meijer, editors, *Selected Areas in Cryptography*, volume 1556 of *Lecture Notes in Computer Science*, pages 339–361. Springer Berlin Heidelberg, 1998.

[18] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow. Elliptic Curve Cryptography in Practice. *IACR Cryptology ePrint Archive, Report 2013/734*, 2013.

[19] D. Brumley and D. Boneh. Remote Timing Attacks Are Practical. In *Proceedings of the 12th Conference on USENIX Security Symposium*, 2003.

[20] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz-Standards. `https://www.bsi.bund.de/EN/Publications/BSIStandards/standards.html`. (Last accessed: February 12, 2014).

[21] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. RFC 3428 – Session Initiation Protocol (SIP) Extension for Instant Messaging, 2002.

[22] R. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In B. Pfitzmann, editor, *Advances in Cryptology - EURO-CRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer Berlin Heidelberg, 2001.

[23] Y.-L. Chen, W.-C. Ku, Y.-C. Yeh, and D.-M. Liao. A Simple Text-Based Shoulder Surfing Resistant Graphical Password Scheme. In *IEEE International Symposium on Next-Generation Electronics (ISNE)*, pages 161–164, 2013.

[24] Y. Cherdantseva and J. Hilton. A Reference Model of Information Assurance & Security. In *Eighth International Conference on Availability, Reliability and Security (ARES)*, pages 546–555, 2013.

[25] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-Side Defense Against Web-Based Identity Theft. In *Network and Distributed System Security Symposium*, 2004.

[26] Cisco. Voice Over IP - Per Call Bandwidth Consumption. `http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml`. (Last accessed: February 12, 2014).

[27] A. Dainotti, A. King, k. Claffy, F. Papale, and A. Pescapè. Analysis of a "/0" Stealth Scan From a Botnet. In *Proceedings of the ACM conference on Internet measurement conference*, pages 1–14, 2012.

[28] J. P. Degabriele and K. G. Paterson. On the (in)Security of IPsec in MAC-then-encrypt Configurations. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 493–504, 2010.

[29] G. Delac, M. Silic, and J. Krolo. Emerging Security Threats for Mobile Platforms. In *Proceedings of the 34th International Convention (MIPRO)*, pages 1468–1473, 2011.

[30] H. Delfs and H. Knebl. *Introduction to Cryptography - Principles and Applications*. Information Security and Cryptography. Springer, 2007.

[31] D. E. Denning. An Intrusion-Detection Model. *IEEE Trans. Softw. Eng.*, 13(2):222–232, 1987.

[32] Der Spiegel Online. NSA-Programm "Quantumtheory": Wie der US-Geheimdienst weltweit Rechner knackt. `http://www.spiegel.de/netzwelt/netzpolitik/`

`quantumtheory-wie-die-nsa-weltweit-rechner-hackt-a-941149.html`. (Last accessed: February 12, 2014).

[33] P. Deutsch. RFC 1952 – GZIP File Format Specification Version 4.3, 1996.

[34] T. Dierks and E. Rescorla. RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2, 2008.

[35] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[36] W. Diffie, P. C. van Oorschot, and M. Wiener. Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

[37] DIN Norm DIN VDE 31000 Teil 2: Allgemeine Leitsätze Für Das Sicherheitsgerechte Gestalten Technischer Erzeugnisse, 1987.

[38] R. do Carmo, M. Nassar, and O. Festor. Artemisa: an Open-Source Honeypot Back-End to Support Security in VoIP Domains. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 361–368, 2011.

[39] S. Dritsas, I. Mallios, M. Theoharidou, G. F. Marias, and D. Gritzalis. Threat Analysis of the Session Initiation Protocol Regarding Spam. In *International Performance Computing and Communications Conference (IPCCC)*, pages 426–433, 2007.

[40] T. Duong and J. Rizzo. Here Come The XOR Ninjas. 2011.

[41] C. Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. Oldenbourg, 2007.

[42] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[43] G. Epiphaniou, C. Maple, P. Sant, and G. Safdar. Effects of Iterative Block Ciphers on Quality of Experience for Internet Protocol Security Enabled Voice over IP Calls. In *Information Security, IET (Volume: 6, Issue:3)*, pages 141–148. IEEE, 2012.

[44] E. Eren and K.-O. Detken. *VoIP Security*. Hanser, 2007.

[45] N. Ferguson. Authentication Weaknesses in GCM. Technical report, 2005.

[46] N. Ferguson, B. Schneier, and T. Kohno. *Cryptography Engineering - Design Principles and Practical Applications*. Wiley, 2010.

[47] G. Flaig, M. Hoffmann, and S. Langauf. *Internet-Telefonie. VoIP mit Asterisk und SER*. Open Source Press, 2005.

[48] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication, 1999.

[49] M. Garuba, J. Li, and Z. Yi. Security in the New Era of Telecommunication: Threats, Risks and Controls of VoIP. *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, pages 587–591, 2008.

[50] S. Gastellier-Prevost, G. G. Granadillo, and M. Laurent. A Dual Approach to Detect Pharming Attacks at the Client-Side. In *New Technologies, Mobility and Security*, pages 1–5, 2011.

[51] S. Gauci. Distributed SIP Scanning During Halloween Weekend. `http://blog.sipvicious.org/2010/11/distributed-sip-scanning-during.html`. (Last accessed: February 12, 2014).

[52] MaxMind - GeoIP | IP Address Location Technology. `http://www.maxmind.com/app/ip-location/`. (Last accessed: February 12, 2014).

[53] O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2000.

[54] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

[55] O. Goldreich. On Post-Modern Cryptography. Cryptology ePrint Archive, Report 2006/461, 2006.

[56] M. Gruber, F. Fankhauser, S. Taber, C. Schanes, and T. Grechenig. Security Status of VoIP Based on the Observation of Real-World Attacks on a Honeynet. In *The Third IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*, pages 1041–1047, 2011.

[57] M. Gruber, F. Fankhauser, S. Taber, C. Schanes, and T. Grechenig. Trapping and Analyzing Malicious VoIP Traffic Using a Honeynet Approach. In *The 6th International Conference on Internet Technology and Secured Transactions (ICITST)*, pages 442–447, 2011.

[58] M. Gruber, M. Maier, M. Schafferer, C. Schanes, and T. Grechenig. Concept and Design of a Transparent Security Layer to Enable Anonymous VoIP Calls. In *Proceedings of the International Conference on Advanced Networking, Distributed Systems and Applications (INDS)*, 2014.

[59] M. Gruber, C. Schanes, F. Fankhauser, and T. Grechenig. Voice Calls for Free: How the Black Market Establishes Free Phone Calls – Trapped and Uncovered by a VoIP Honeynet. In *Eleventh Annual International on Privacy, Security and Trust (PST)*, pages 205–212, 2013.

[60] M. Gruber, C. Schanes, F. Fankhauser, M. Moutran, and T. Grechenig. Architecture for Trapping Toll Fraud Attacks Using a VoIP Honeynet Approach. In J. Lopez, X. Huang, and R. Sandhu, editors, *Network and System Security*, volume 7873 of *Lecture Notes in Computer Science*, pages 628–634. Springer Berlin Heidelberg, 2013.

[61] M. Gruber, P. Wieser, S. Nachtnebel, C. Schanes, and T. Grechenig. Extraction of ABNF Rules from RFCs to Enable Automated Test Data Generation. In L. Janczewski, H. Wolfe, and S. Shenoi, editors, *Security and Privacy Protection in Information Processing Systems*, volume 405 of *IFIP Advances in Information and Communication Technology*, pages 111–124. Springer Berlin Heidelberg, 2013.

[62] V. K. Gurbani and V. Kolesnikov. Work in Progress: A Secure and Lightweight Scheme for Media Keying in the Session Initiation Protocol (SIP). In *Principles, Systems and Applications of IP Telecommunications (IPTComm)*, pages 32–41, 2010.

[63] V. K. Gurbani and V. Kolesnikov. A Survey and Analysis of Media Keying Techniques in the Session Initiation Protocol (SIP). In *Communications Surveys and Tutorials*, pages 183–198, 2011.

[64] M. Handley, V. Jacobson, and C. Perkins. RFC 4566 – Session Description Protocol (SDP), 2006.

[65] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. RFC 2543 – SIP: Session Initiation Protocol, 1999.

[66] D. Hoffstadt, A. Marold, and E. Rathgeb. Analysis of SIP-Based Threats Using a VoIP Honeynet System. In *11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 541–548, 2012.

[67] C. Holmberg, E. Burger, and H. Kaplan. RFC 6086 – Session Initiation Protocol (SIP) INFO Method and Package Framework, 2011.

[68] Honeynet Project. Know Your Enemy: Defining Virtual Honeynets. `http://old.honeynet.org/papers/virtual/`. (Last accessed: March 25, 2014).

[69] Honeynet Project. Know Your Enemy: Honeynets. `http://old.honeynet.org/papers/honeynet/`. (Last accessed: March 25, 2014).

[70] Honeynet Project. The Honeywall Project. `https://projects.honeynet.org/honeywall/`. (Last accessed: February 13, 2014).

[71] Infonetics Research (Diane Myers – Principal Analyst of VoIP, UC and IMS). VoIP and UC Services and Subscribers Report. `http://www.infonetics.com/pr/2013/1h13-VoIP-UC-Services-Market-Highlights.asp`. (Last accessed: February 12, 2014).

[72] B. Isemann, M. Gruber, M. Grünberger, C. Schanes, and T. Grechenig. Chaotic Adhoc Data Network – A Bike Based System for City Networks. In *The 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE 2014)*, 2014.

[73] ITU-T. G.114 One-Way Transmission Times. In *Series G: Transmission Systems and Media, Digital Systems and Networks*, 2003.

[74] R. Jin and B. Wang. Malware Detection for Mobile Devices Using Software-Defined Networking. In *Second GENI Research and Educational Experiment Workshop (GREE)*, pages 81–88, 2013.

[75] A. Johnston, J. Yoakum, and K. Singh. Taking on WebRTC in an Enterprise. *Communications Magazine, IEEE*, 51(4):48–54, 2013.

[76] A. Joux. Authentication Failures in NIST Version of GCM. Technical report, 2006.

[77] A. Juels and J. G. Brainard. Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In *NDSS*. The Internet Society, 1999.

[78] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/CRC Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[79] J. Kelsey. Compression and Information Leakage of Plaintext. In J. Daemen and V. Rijmen, editors, *Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 263–276. Springer Berlin Heidelberg, 2002.

[80] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX, 1883.

[81] A. Keromytis. A Comprehensive Survey of Voice over IP Security Research. *Communications Surveys Tutorials, IEEE*, 14(2):514–537, 2012.

[82] H. Kersten, J. Reuter, and K.-W. Schröder. *IT-Sicherheitsmanagement nach ISO 27001 und Grundschutz. Der Weg zur Zertifitierung.* Friedr. Vieweg & Sohn Verlag, 2008.

[83] M. Koetter. dionaea – Catches Bugs. `http://dionaea.carnivore.it/`. (Last accessed: February 12, 2014).

[84] H. Krawczyk. The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). In J. Kilian, editor, *Advances in Cryptology - CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer Berlin Heidelberg, 2001.

[85] H. Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. In V. Shoup, editor, *Advances in Cryptology - CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566. Springer Berlin Heidelberg, 2005.

[86] H. Krawczyk, M. Bellare, and R. Canetti. RFC 2104 – HMAC: Keyed-Hashing for Message Authentication, 1997.

[87] T. Krovetz and P. Rogaway. The Software Performance of Authenticated-Encryption Modes. In A. Joux, editor, *Fast Software Encryption*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer Berlin Heidelberg, 2011.

[88] M. Kulin, T. Kazaz, and S. Mrdovicg. SIP Server Security With TLS: Relative Performance Evaluation. In *IX International Symposium on Telecommunications (BIHTEL)*, pages 1–6, 2012.

[89] B. Laurie and R. Clayton. "Proof-of-Work" Proves Not to Work. 2004.

[90] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An Efficient Protocol for Authenticated Key Agreement. *Des. Codes Cryptography*, 28(2):119–134, 2003.

[91] A. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 446–465. Springer Berlin Heidelberg, 2000.

[92] C. Lv and Q. Zhao. Integration of Data Compression and Cryptography: Another Way to Increase the Information Security. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, pages 543–547, 2007.

[93] D. McGrew and E. Rescorla. RFC 5764 – Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP), 2010.

[94] D. A. McGrew and J. Viega. The Galois/Counter Mode of Operation (GCM). *NIST Modes Operation Symmetric Key Block Ciphers*, 2005.

[95] A. Menezes. Another Look at HMQV. Cryptology ePrint Archive, Report 2005/205, 2005.

[96] A. Menezes, M. Qu, and S. Vanstone. Some new key Agreement Protocols Providing Mutual Implicit Authentication. In *Selected Areas in Cryptography*, 1995.

[97] A. Menezes and B. Ustaoglu. On the Importance of Public-key Validation in the MQV and HMQV Key Agreement Protocols. In *Proceedings of the 7th International Conference on Cryptology*, pages 133–147, 2006.

[98] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 5 edition, 2001.

[99] R. C. Merkle. Secure Communications over Insecure Channels. *Commun. ACM*, 21(4):294–299, 1978.

[100] F. Miller. *Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams*. C.M. Cornwell, 1882.

[101] V. Miller. Use of Elliptic Curves in Cryptography. In H. Williams, editor, *Advances in Cryptology – CRYPTO '85 Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer Berlin Heidelberg, 1986.

[102] W. Nagao, Y. Manabe, and T. Okamoto. A Universally Composable Secure Channel Based on the KEM-DEM Framework. In J. Kilian, editor, *Theory of Cryptography*, volume 3378 of *Lecture Notes in Computer Science*, pages 426–444. Springer Berlin Heidelberg, 2005.

[103] M. Nassar, R. State, and O. Festor. VoIP Honeypot Architecture. In *10th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 109–118, 2007.

[104] R. C. Newman. Cybercrime, Identity Theft, and Fraud: Practicing Safe Internet – Network Security Threats and Vulnerabilities. In *Proceedings of the 3rd annual conference on Information security curriculum development (InfoSecCD)*, pages 68–78, 2006.

[105] A. Niemi. RFC 3903 – Session Initiation Protocol (SIP) Extension for Event State Publication, 2004.

[106] F. Palmieri and U. Fiore. Providing True End-to-End Security in Converged Voice over IP Infrastructures. In *Computers and Security (Volume: 28, Issue: 6)*, pages 433–449, 2009.

[107] K. G. Paterson and N. J. AlFardan. Plaintext-Recovery Attacks Against Datagram TLS. In *Network and Distributed System Security Symposium (NDSS 2012)*, 2012.

[108] D. Perez-Botero and Y. Donoso. VoIP Eavesdropping: A Comprehensive Evaluation of Cryptographic Countermeasures. In *Second International Conference on Networking and Distributed Computing (ICNDC)*, pages 192–196, 2011.

[109] J. Postel. RFC 768 – User Datagram Protocol (UDP), 1980.

[110] J. Postel. RFC 793 – Transmission Control Protocol (TCP), 1981.

[111] N. Provos. Developments of the Honeyd Virtual Honeypot. `http://www.honeyd.org/`. (Last accessed: February 13, 2014).

[112] N. Provos and T. Holz. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional, 2007.

[113] M. Raepple. *Sicherheitskonzepte für das Internet*. dpunkt.verlag GmbH, 2001.

[114] T. Raffetseder, C. Kruegel, and E. Kirda. Detecting System Emulators. In J. Garay, A. Lenstra, M. Mambo, and R. Peralta, editors, *Information Security*, volume 4779 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2007.

[115] J. M. Reason and D. G. Messerschmitt. The Impact of Confidentiality on Quality of Service in Heterogeneous Voice over IP Networks. In *Management of Multimedia on the Internet*, volume 2216, pages 175–192. Springer Berlin Heidelberg, 2001.

[116] E. Rescorla and N. Modadugu. RFC 4347 – Datagram Transport Layer Security, 2006.

[117] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[118] A. Roach. RFC 6665 – SIP-Specific Event Notification, 2012.

[119] P. Rogaway. Evaluation of Some Block Cipher Modes of Operation. Technical report, Cryptography Research and Evaluation Committee (CRYPTREC), 2011.

[120] P. Rogaway, M. Bellare, and J. Black. OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.

[121] C. Rohlf and Y. Ivnitskiy. The Security Challenges of Client-Side Just-in-Time Engines. *IEEE Security & Privacy*, 10:84–86, 2012.

[122] J. Rosenberg. RFC 3311 – The Session Initiation Protocol (SIP) UPDATE Method, 2002.

[123] J. Rosenberg and H. Schulzrinne. RFC 3262 – Reliability of Provisional Responses in the Session Initiation Protocol (SIP), 2002.

[124] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261 – SIP: Session Initiation Protocol, 2002.

[125] SANS Internet Storm Center. Port Details | SANS Internet Storm Center. `http://isc.sans.edu/port.html?port=5060`. (Last accessed: February 12, 2014).

[126] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard. A Secure and Efficient Authenticated Diffie-Hellman Protocol. In *Proceedings of the 6th European Conference on Public Key Infrastructures, Services and Applications (EuroPKI)*, pages 83–98, 2010.

[127] K. A. Scarfone and P. M. Mell. SP 800-94. Guide to Intrusion Detection and Prevention Systems (IDPS). Technical report, 2007.

[128] M. Schafferer, M. Gruber, C. Schanes, and T. Grechenig. Data Retention Services with Soft Privacy Impacts: Concept and Implementation. In *Proceedings of the International Conference on Software Engineering and Service Science (ICSESS)*, 2014.

[129] G. H. Schildt, D. Kahn, C. Krügel, and C. Moerz. *Einführung in die Technische Informatik*. Springer-Verlag, 2005.

[130] K. Schmidt. *Der IT Security Manager*. Carl Hanser Verlag, 2006.

[131] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., 1996.

[132] B. Schneier. *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*. Springer-Verlag New York, Inc., 2003.

[133] B. Schneier. *Secrets & Lies: Digital Security in a Networked World*. Wiley Publishing, Inc., 2004.

[134] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889 – RTP: A Transport Protocol for Real-Time Applications, 1996.

[135] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 3550 – RTP: A Transport Protocol for Real-Time Applications, 2003.

[136] G. Seroussi. Elliptic Curve Cryptography. In *Information Theory and Networking Workshop*, 1999.

[137] C. Shen, E. Nahum, H. Schulzrinne, and C. P. Wright. The Impact of TLS on SIP Server Performance: Measurement and Modeling. *IEEE/ACM Trans. Netw.*, 20(4):1217–1230, 2012.

[138] E. Skoudis and T. Liston. *Counter Hack Reloaded. A Step-by-Step Guide to Computer Attacks and Effective Defenses*. Pearson Education, Inc., 2006.

[139] R. Sparks. RFC 3515 – The Session Initiation Protocol (SIP) Refer Method, 2003.

[140] R. Sparks and T. Zourzouvillys. RFC 6026 – Correct Transaction Handling for 2xx Responses to Session Initiation Protocol (SIP) INVITE Requests, 2010.

[141] L. Spitzner. Honeypots: Definition and Value of Honeypots. `http://www.tracking-hackers.com/papers/honeypots.html`. (Last accessed: February 13, 2014).

[142] L. Spitzner. The Honeynet Project: Trapping the Hackers. *Security & Privacy Magazine, IEEE*, 1(2):15–23, 2003.

[143] R. Stewart. RFC 4960 – Stream Control Transmission Protocol, 2007.

[144] L. Tawalbeh, M. Mowa, and W. Aljoby. Use of Elliptic Curve Cryptography for Multimedia Encryption. In *IET Information Security*, 2013.

[145] The Guardian. Edward Snowden: NSA whistleblower answers reader questions. `http://www.theguardian.com/world/2013/jun/17/edward-snowden-nsa-files-whistleblower`. (Last accessed: February 12, 2014).

[146] The Guardian. NSA Files: Decoded – What the revelations mean for you. `http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded`. (Last accessed: February 12, 2014).

[147] The Guardian. NSA surveillance: A guide to staying secure. http://www.theguardian.com/world/2013/sep/05/nsa-how-to-remain-secure-surveillance. (Last accessed: February 12, 2014).

[148] P. Thermos and A. Takanen. *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*. Addison-Wesley Professional, 2007.

[149] A. Tsolkas and F. Wimmer. *Wirtschaftsspionage und Intelligence Gathering: Neue Trends der wirtschaftlichen Vorteilsbeschaffung*. Vieweg+Teubner Verlag, 2012.

[150] C. Valli. An Analysis of Malfeasant Activity Directed at a VoIP Honeypot. In *8th Australian Digital Forensics Conference*, 2010.

[151] C. Valli and M. Al-Lawati. Developing Robust VoIP Router Honeypots Using Device Fingerprints. In *1st International Cyber Resilience Conference*, 2010.

[152] VoIP Honey – a Comprehensive VoIP Honeypot. http://voiphoney.sourceforge.net/. (Last accessed: February 13, 2014).

[153] VoIP Security Alliance. VOIPSA, VoIP Security and Privacy Threat Taxonomy. http://www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf. (Last accessed: January 04, 2014).

[154] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. Technical report, 1996.

[155] F. Weingarten, M. Schloesser, and J. Gilger. Honeymap. http://map.honeycloud.net/. (Last accessed: January 25, 2014).

[156] W. Werapun, A. A. El Kalam, B. Paillassa, and J. Fasson. Solution Analysis for SIP Security Threats. In *International Conference on Multimedia Computing and Systems (ICMCS)*, pages 174–180, 2009.

[157] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose. Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on Fon-iks. In *IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2011.

[158] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot Me if You Can: Uncovering Spoken Phrases in Encrypted VoIP Conversations. In *IEEE Symposium on Security and Privacy (SP)*, pages 35–49, 2008.

[159] A. C.-C. Yao and Y. Zhao. OAKE: A New Family of Implicitly Authenticated Diffie-hellman Protocols. In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security (CCS)*, pages 1113–1128, 2013.

[160] P. Zimmermann and A. Johnston. RFC 6189 – ZRTP: Media Path Key Agreement for Unicast Secure RTP, 2011.

[161] T. Zourzouvillys and E. Rescorla. An Introduction to Standards-Based VoIP: SIP, RTP, and Friends. *Internet Computing, IEEE*, 14(2):69–73, 2010.