FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Secure Operation of Open Source Private Branch Exchange (PBX) Servers to Provide reliable VoIP Services

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Martin Karl Maier
Matrikelnummer 0827842

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig
Mitwirkung: Florian Fankhauser

Wien, 19. August 2012

(Unterschrift Verfasser/In)

(Unterschrift Betreuung)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Secure Operation of Open Source Private Branch Exchange (PBX) Servers to Provide reliable VoIP Services

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Business Informatics

by

Martin Karl Maier
Registration Number 0827842

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Thomas Grechenig
Assistance: Florian Fankhauser

Vienna, 19. August 2012

(Signature of Author)

(Signature of Advisor)

Secure Operation of Open Source Private Branch Exchange (PBX) Servers to Provide reliable VoIP Services

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Business Informatics

by

Martin Karl Maier
Registration Number 0827842

elaborate at the
Institut of Computer Aided Automation
Reseach Group Industrial Software
to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Thomas Grechenig
Assistance: Florian Fankhauser

Vienna, 19. August 2012

Statement by Author

Martin Karl Maier
Passail 96, 8162 Passail

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.

I hereby declare that I am the sole author of this thesis, that I have completely indicated all sources and help used, and that all parts of this work - including tables, maps and figures - if taken from other works or from the internet, whether copied literally or by sense, have been labelled including a citation of the source.

______________________________ ______________________________

(Place, Date) (Signature of Advisor)

Acknowledgements

Ich möchte mich hiermit bei allen bedanken, die mich während meiner Uni-Zeit unterstützt und auch gestützt haben.

Allen voran meinen Eltern, welche mich stets gefördert haben und ohne deren Hilfe ich sicher nicht da wäre wo ich jetzt bin. Weiters danke ich meinem Arbeitgeber, der Firma cargo-partner GmbH in Fischamend, welche mir und meinen universitären Aktivitäten immer sehr viel Verständnis und zeitliche Flexibilität entgegengebracht hatte.

Und Vera, danke fürs sein wie du bist und fürs viele Verständnis auch wenn ich mal nicht so gut aufgelegt bin. I love U!

Abstract

Since the invention of the telephone about 150 years ago, communication systems have been subject to continuous improvement and form an important factor in modern life. Current development favors Voice over IP based systems, which utilize regular Internet infrastructure and their advantages. Also the ongoing evolvement of software development enables open source products to mature and to form a viable alternative to other software design approaches.

The designers of open source private branch exchange (PBX) Asterisk pursue utilizing open source software for the development of a Voice over IP systems. As it combines advantages of both, open source software and Voice over IP systems, a promising future of Asterisk is guaranteed. Although great benefit can be achieved, the openness of and integration into regular Internet infrastructure evokes doubts about the security of the system.

This thesis presents an approach to point out potential security threats of an open source PBX system, to analyize their system vulnerabilities and to estimate the risk of such vulnerabilities getting exposed. For an Asterisk based reference system, mitigation strategies for identified risk are presented.

Keywords

Asterisk, Voice over IP security, security evaluation, risk mitigation.

Kurzfassung

Seit der Erfindung des Telefons vor etwa 150 Jahren haben sich Kommunikationssysteme ständig weiterentwickelt und stellen heute einen wichtigen Teil der Kommunikationskultur dar. Die aktuelle Entwicklung bewegt sich hin zu Voice over IP (VoIP) Systemen, die auf der Infrastruktur des Internet aufsetzen. Auf der anderen Seite trägt die laufende Evolution von Softwareentwicklungsstandards dazu bei, Open Source Software Produkte laufend in ihrer Qualität zu verbessern und somit auch eine praktikable Alternative zu kommerziellen Softwareentwicklungsansätzen zu bieten.

Trotz des großen Nutzens, welcher durch den Einsatz des Open Source Konzept in Verbindung mit VoIP Systemen erzielt werden kann, entstehen durch die daraus resultierende Offenheit und Integration in die Internet Infrastruktur neue Sicherheitsprobleme.

Die weitverbreitete Open Source Telefonanlagensoftware Asterisk wird im Rahmen der vorliegenden Arbeit herangezogen, um examplarisch bestehende Sicherheitsprobleme bei VoIP-Lösungen aufzuzeigen. Dabei wird eine Methode vorgestellt, deren Anwendung zur zuverlässigen Identifikation von Sicherheitsbedrohungen in Open Source Telefonanlagen führt.

Anhand einer Beispielinstallation wird ein Maßnahmenkatalog präsentiert, der zur Minimierung der dargestellten Risken führt. Der Maßnahmenkatalog beschreibt technische sowie organisatorische Maßnahmen, welche einen sicheren Betrieb einer Open Source Telefonanlagensoftware ermöglichen.

Schlüsselwörter

Asterisk, Sicherheit von Voice over IP Systemen, Sicherheitsanalyse, Risikominderung

Contents

1 Introduction

Telephones (or Phones) enable voice communication of people separated by even large geographical distance and are one of the most heavily used tools in our modern world. Nowadays phones are considered to be indispensable to businesses, individuals and other organizations nowadays. With the Internet becoming widely available for users around the world, the emergence of new technologies enabling voice calls developed at increasing speed. The so-called Voice over Internet Protocol (VoIP) approach allows to use Internet infrastructure for handling media sessions without the requirement of having dedicated phone lines.

In the older days, Private Branch Exchange (PBX) systems were responsible for intra-office communication which links the whole office environment to typically one telephony provider uplink. With the rise of VoIP, the role of classic PBX systems changed to a sort of multimedia exchange system with PBX vendors adopting to these changes. At the same time, open source solutions started to develop.

Such an open source VoIP PBX has a lot of benefits compared to stand-alone, old fashioned phone exchanges. It utilizes standard in-house Internet Protocol (IP) infrastructure, does not require additional cabling, can be installed on standard server equipment or does not require specific hardware, to name only a few.

Open source software is primarily licensed under open-source GNU General Public License Version 2 (GPLv2) License [43], which states that a vendor always has to provide the source code and is allowed to demand money for its products. In the majority of cases, open source projects under GPLv2 license are either for free or are available at quite reasonable prices. As VoIP systems only call for limited hardware aquisition costs and licensing costs can be considered as very low, they are able to achieve a drastic decreases in the overall Total Cost of Ownership (TCO). Hence, this solution is considerably attractive for commercial usage.

Throughout this thesis, the focus lies on Asterisk open source PBX [36] which will be used as reference PBX system. Nevertheless, the findings can be applied to other open source PBX systems as well.

Compared to older PBX Systems, using their own hardware infrastructure, security of an VoIP based PBX system needs to be seen more wholistic because according to Porter [86] *converged VoIP and data networks inherit all the security weaknesses of the IP protocol.* When it comes to Asterisk, the easy adaptability to existing IP networks brings a lot of benefits, but implies disadvantages too. By sharing the physical media with other applications such as e-mail or web sites, the transfered traffic can be monitored, captured, analyzed or even manipulated more easily. The requirement to have a secure and reliable network environment is more important than ever as due to the flexibility of IP systems multiple systems could be harmed simultaneously. Having a secure network environment is therefore a necessity and precondition to secure Asterisk operation.

By the use of a reference system this thesis will analyze the expected behavior of an Asterisk installation. It tries to help pointing out potential security threats and analyzes vulnerabilities of the whole system. After estimating the risks of vulnerabilities getting exploited, mitigation strategies will be presented to make the operation of Asterisk more secure. The proposed mitigation strategies will not focus on Asterisk only, but will direct the readers attention towards the whole system environment.

Chapters 2 and 3 of the thesis introduce theoretical concepts required to conduct a security analysis on a Linux based Asterisk installation, which is described in Chapter 4 in detail. This is followed by the a security analysis identifying possible threats and vulnerabilities (Chapter 5).

Through the implementation of proposed mitigation strategies presented in Chapter 6, advantages of Asterisk can be fully utilized leading to a sustainable PBX solution which could propose a viable alternative for businesses today.

2 Technical Fundamentals

This chapter describes the technical fundamentals required for the consecutive security analysis. The concepts and strategies described will help to understand the considerations of the security analysis conducted in Chapter 5.

2.1 Voice Over IP

VoIP describes the ability to transfer voice communication via IP [87] networks. Specific applications and protocols enabling VoIP can be classified as layer 4 (application layer) protocols on Transmission Control Protocol / Internet Protocol (TCP/IP) protocol stack [20] or as layer 7 protocol on International Organization for Standardization (ISO) Open Systems Interconnection (OSI) protocol stack [121], although these voice protocols are usually only used in combination with TCP/IP stack. Figure 2.1 compares both protocol stacks.

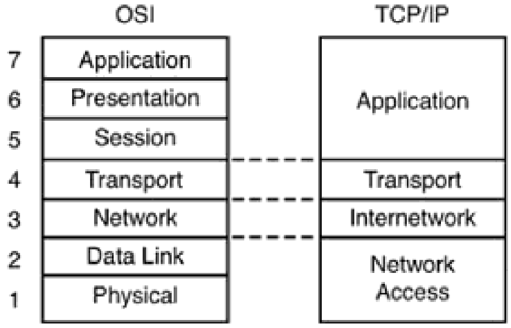

Figure 2.1: TCP/IP and OSI Model comparison ([78])

VoIP protocols share the underlying layers with other applications like Simple Mail Transfer Protocol (SMTP) [61] for sending e-mails or Hypertext Transfer Protocol (HTTP) [40] for browsing web pages. Some of these underlying layers can be exchanged easily without the loss of functionality. For example, there is no difference in functionality of VoIP protocols whether a wireless or a wired connection is used on layer 1 (TCP/IP stack layer 1 [20]), although it makes a big difference in terms of security as the attack pattern will be different.

VoIP protocols share also TCP/IP stack layer 2 protocol IP with other applications. Today, IP is the main protocol used in the Internet and is responsible for addressing and determining routes to other hosts on the network. In case access to the public Internet is not needed and there is a different addressing/routing protocol in place, replacement of IP on this layer is possible.

On the transport layer (TCP/IP stack layer 3) VoIP protocols can make use of Transmission Control Protocol (TCP)[88] or User Datagram Protocol (UDP)[89].

VoIP protocols fulfill two main purposes whereas each protocol can be assigned to one of those purposes. The first group of protocols is responsible for signaling and session setup, including all session management functions like call forwarding or the setup of video calls, whereas the second group is responsible for transferring payload data like encoded video or audio streams.

Nowadays, widely used signaling protocols are the Internet Engineering Task Force (IETF)-defined Session Initiation Protocol (SIP) [95] and International Telecommunication Union - Telecommunication (ITU-T) recommendation H.323 [119] as well as vendor proprietary protocols like Cisco's Skinny Call Control Protocol (SCCP). As an example, text-based SIP, is using elements of HTTP or SMTP protocols and is capable of doing session management, covering session setup, call transfer up to the tear down of sessions. Multiple SIP Transactions between two user agents can be part of a single SIP dialog. Nevertheless, SIP is not able to handle the setup of the media streams and is utilizing Session Description Protocol (SDP) [49] for this purpose. SDP describes the capabilities of the media-endpoint and is incorporated into SIP messages. These protocols are typically text-based and do not make use of any security features like encryption or authentication by default and hence are very fragile in terms of security. As a consequence, it is possible to use Transport Layer Security (TLS) Protocol [35] to encapsulate SIP messages allowing encrypted session management.

A protocol capable of transferring media streams is the Real-time Transport Protocol (RTP) [101]. This protocol has the sole purpose of transferring payload data and is not capable of doing management tasks. Therefore, RTP utilizes Real-time Transport Control Protocol (RTCP) [101] to set up media connections or to provide information about the media stream. Media protocols as RTCP or RTP utilize UDP[89] on TCP/IP stack layer 3 as it favors timeliness over reliability in comparison to TCP. As most media transfer protocols, also RTP and RTCP are unencrypted and therefore not suited to be used in an non-private network environment. The secure counterparts of RTP and RTCP are Secure Real-time Transport Protocol (SRTP) [15] and Secure Real-time Transport Control Protocol (SRTCP) [15] enabling encryption and message authentication.

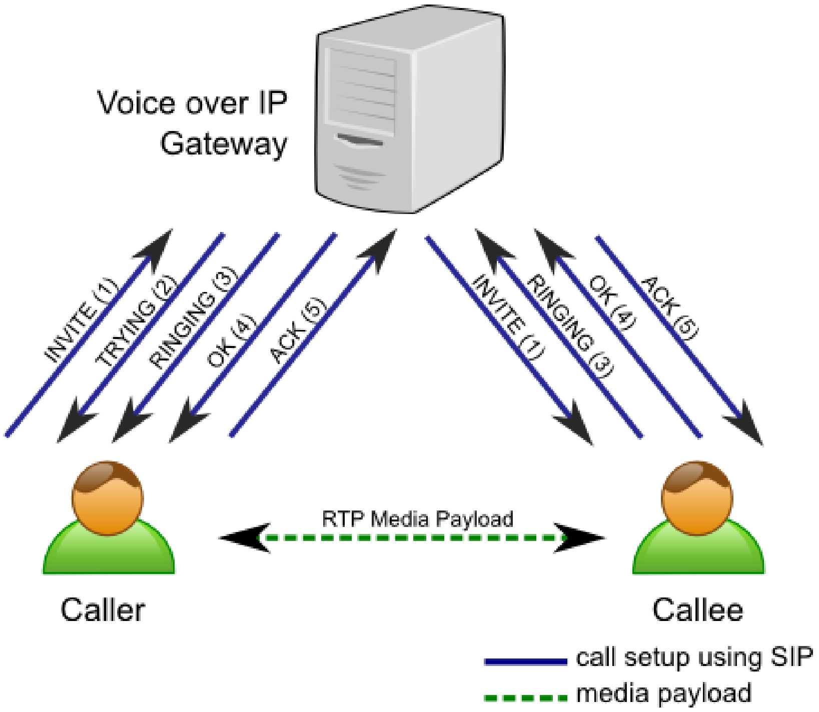

Figure 2.2: SIP Session Setup and Media Stream - Direct Media Stream (see [106])

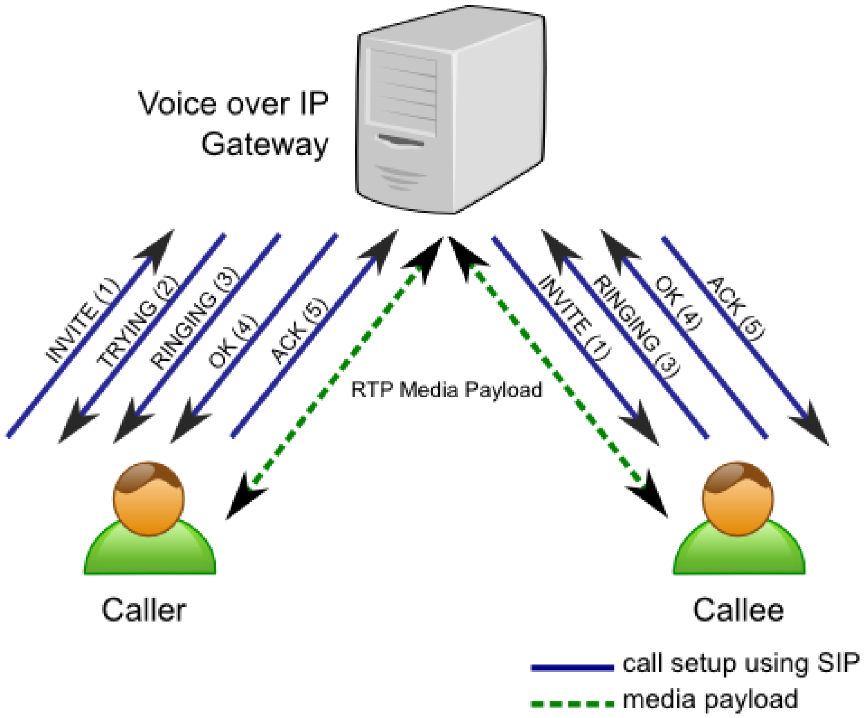

Figure 2.3: SIP Session Setup and Media Stream - Media Termination Point (see [106])

Figure 2.2 and Figure 2.3 show the possible flow of messages using SIP for a media session setup. They show that both, termination of media streams at both client devices as well as routing via a media termination point is possible.

The signaling process is initiated by the caller sending an *INVITE* message to the the VoIP gateway. This message is adjusted and relayed to the callee while the caller is informed about the ongoing attempt to establish a connection using a *TRYING* message. Once the callee has received and processed the session setup request, the incoming connection request is alerted to the end user and a *RINGING* message is used to inform about this calling state. Once the user accepts the connection request (answers the call), an *OK* message is used to notify the VoIP gateway. The gateway forwards this message to the caller who is answering with an *ACK* message to acknowledge the session creation. Whether a direct media connection (Figure 2.2) or media termination points (Figure 2.3) are used depends on the VoIP gateway's configuration and is indicated by the content of the messages exchanged during session setup.

Scenario one (Figure 2.2) illustrates the following: Once the session has been initialized by the VoIP gateway, media will be transfered directly between both client devices without the VoIP gateway being involved anymore up to the point when the session has to be teared down. Scenario two (Figure 2.3) utilizes a intermediary media gateway (or media proxy), having the media stream transfered from the first user to the gateway and from the gateway to the second user.

The second scenario allows more accurate session information which can, for example, be used for billing purposes. On the downside though, it causes higher network bandwidth consumption. The way the VoIP system is used heavily depends on the network environment and the system requirements.

VoIP systems allow to connect to other VoIP systems as well as to connect to foreign systems like the Public Switched Telephone Network (PSTN).

Connecting VoIP devices to PSTN providers may possibly require the installation of dedicated hardware responsible for de- and encoding of local VoIP traffic according to the PSTN network requirements. Those may vary according to the connection type and provider used as well as the physical location (country). The hardware module automatically becomes a media termination point. Another possibility to connect Asterisk to PSTN providers is using SIP Trunks. In this case, the telephony company will supply a SIP Trunk termination point in form of an IP address or Domain Name System (DNS) host name to which Asterisk is able to connect via the network infrastructure. Therefore, no additional hardware is required to be installed on the Asterisk server.

2.2 PBX Asterisk

Asterisk is an IP based modular telephony engine which allows to build Voice over IP systems tailored to particular business cases and requirements. In its core it is an engine for Asterisk's proprietary Scripting Language dialplan [102], allowing a very flexible management of calls and building the logic of any Asterisk system [66]. In addition to dialplan scripting functions and applications, Asterisk is extended by various modules shown in Figure 2.4. The behavior of Asterisk is controlled by text based configuration files.

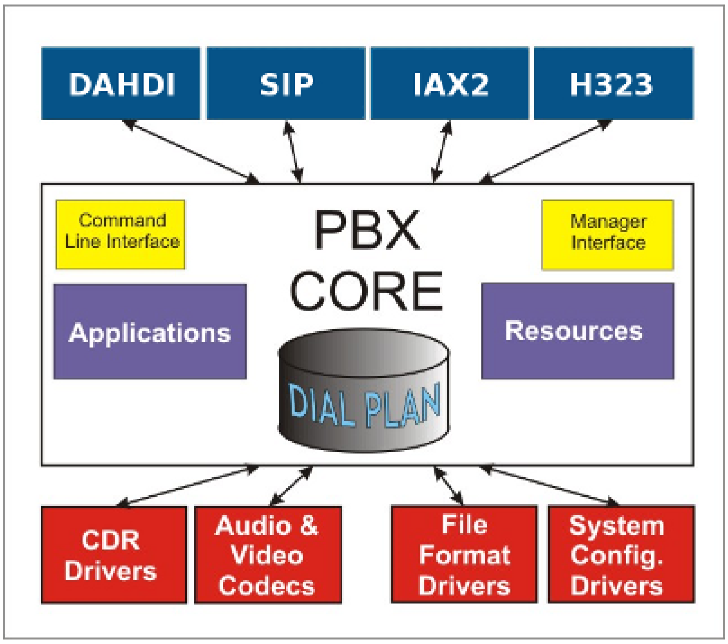

Figure 2.4: Asterisk Architecture (see [66])

Asterisk is made up of different types of modules:

- **Channel Drivers** are in charge of communicating with VoIP devices outside of Asterisk
  - *Digium Asterisk Hardware Driver Interface (DAHDI)* for communication with Peripheral Component Interconnect (PCI) hardware interface cards to connect with PSTN using T1/E1, Integrated Services Digital Network (ISDN) or analog lines.
  - *SIP*, which implements VoIP protocol SIP [95] inside Asterisk, for interactions with locally connected client devices like IP Phones or Video Conferencing Stations as well as connecting to uplink trunks of SIP telephony providers.
  - *Inter-Asterisk Exchange (IAX2)* [107], an UDP based protocol using the same port for signaling and transferring media streams, mainly for interconnecting multiple Asterisk systems together, although it also supports connecting to client devices.
  - *H.323 [119], SCCP [38], Media Gateway Control Protocol (MGCP) [42], Extensible Messaging and Presence Protocol (XMPP) [96]* to allow interoperability with other third party PBX systems implementing other, partly vendor proprietary, VoIP protocols.

  These modules translate particular signaling or protocols to enable Asterisk's core to process them.

- **Dialplan Applications and Functions** help to implement the whole process of call handling and routing. Applications implement features (answer calls, play sound prompts, hang up) and can be invoked by Dialplan scripts. Functions on the other hand may be used to modify various call settings during an active media session.
- **Resource modules** provide external resources which can be utilized by Asterisk. Such resources may include music to be played while a call is on hold or to play recorded voice which is used to lead through a voice mailbox menu.
- **Management Modules**. Asterisk provides two different management modules which make it possible to monitor Asterisk or to execute scripts. These are:
  - The *Command Line Interface (CLI)*, allowing to interact with Asterisk by issuing text based commands.
  - The *Asterisk Manager Interface (AMI)*, allowing to interact with Asterisk utilizing a TCP/IP socket. AMI enables Asterisk to respond to commands or send automatic notifications when triggered by certain events. Graphical User Interfaces (GUI) with the goal of making management and administration of Asterisk easier, may use AMI for communicating with Asterisk.
- **Other Drivers** help performing tasks requiring the usage of resources outside of the Asterisk PBX core.
  - **Call Detail Record (CDR)** and **Call Event Log (CEL)** drivers are in charge of writing call logs to files or databases. These logs can later be used for cost allocations, invoicing or for debugging reasons.
  - **CODECs** are responsible for en- or decoding of media streams. Asterisk is able to en- or decode a variety of different voice and video codecs. The decision about the codes used can be dictated by external connecting devices, by the network bandwidth requirements and by the hardware resources available. A codec enabling a high degree of compression will consume less bandwidth but requires more Central Processing Unit (CPU) power on the server.

    These drivers can also be used to enable various methods of bridging call media between participants. In case session members or media termination points can not agree on a common media codec, such a bridging module is required to convert the session media stream to a different codec in real-time throughout the media session.
  - **File Format Drivers** are used to convert media streams to and from formats which can be stored on disk drives. For example, the usage of a voice mail system requires to have this module implemented.
  - **Configuration Drivers** allow to retrieve Asterisk configuration sections, namely dialplan scripts, from other sources than regular plain text files. Hence, it is possible to manage big Asterisk installations more efficiently by putting configuration sections into external databases.

Asterisk is released as open source and is written in C, it is supposed to run on any Linux based machine which has at least kernel version 2.6 installed. It is licensed under the GPLv2 [43] and various editions can be downloaded for free. Currently, there are multiple versions available, which differentiate themselves through the target audience, the support and the extent of pre-configured features:

- **Asterisk Engine**. The mere Asterisk software without any pre-configurations

- **Basic Asterisk PBX: AsteriskNOW**. Designed for small business providing basic pre-configured VoIP features.
- **Commercial PBX / UC: Switchvox**. Implements powerful Unified Communication features and is designed for small and medium sized businesses. This edition is not free of charge, but is already pre-configured and does not require high efforts in customizing and the initial setup. Although some license fees are applied, the lower system setup efforts may, in the end, balance or even lower the total costs.

2.3 Secure Operation of Software

The current tendency towards information systems distributed throughout the entire Internet which are becoming increasingly bigger has led to the emergence of many new security threats [80]. Vulnerabilities in software can jeopardize intellectual property, consumer trust and business operations. Moreover, a broad spectrum from process control systems to commercial application products, depends on secure and reliable software [124].

For these reasons, the integrity of software has to be secured, especially while it is being executed. Software security requires security matters to be considered right from the beginning of the Systems Development Life Cycle (SDLC) as it can not be patched in later or once the software was finalized [69].

However, a software development process focusing on security matters, does not automatically ensure software boing operated in a secure manner. Every software is executed within a specific environment and can be configured through external parameters. To guarantee a secure operation of software, security considerations concerning the environment as well as external configuration parameters are a necessity.

The Asterisk SDLC helps to meet software security criteria during the design and implementation phases of Asterisk. As this thesis focuses primarily on configuration and environmental issues of a secure Asterisk software operation, the source code of Asterisk is not evaluated [1].

2.3.1 IT Security

Nowadays, Information Technology (IT) systems store and process sensitive data including personal health records, financial details or details about recent vacations, to name only a few. In case these sensitive data ends up within wrong hands, people could be harmed. Hence, it is a requirement to have bespoken data as well as systems managing it secured.

Threats to IT systems are very diverse in terms of their objective and how they proceed to achieve their goals. Examples of such objectives range from simple programmers curiosity to completely shutting down services and systems. The list of threats includes Trojan horses, network based Denial of Service (DoS) attacks or simple frustration of former employees. IT security therefore seeks to minimize a systems vulnerability to such threats by various means.

2.3.1.1 CIA Triad

According to ISC2's Certified Information Systems Security Professional (CISSP) [50] certification, security can be achieved in case an IT system follows the main principals of the CIA triad:

- **Confidentiality** describes the concept of limiting access to information to a trusted group of persons and prevents access or disclosure of data to unauthorized systems. Data has to be secured during all stages of handling information, including the acquisition of data, the processing and the storage of information.

[1] Vulnerability reports show that security issues are often caused by errors in software. For example and according to Allen et al. [2], *CERT has observed through an analysis of thousands of vulnerability reports that most vulnerabilities stem from a relatively small and recurring number of common programming errors.* Because of this, vulnerabilities and risks caused by errors in software can not be omitted and are included in this thesis's security evaluation anyway.

The day-to-day business case of making a payment over the Internet using a credit card illustrates the importance of Confidentiality, as sharing private information like credit card details could violate the privacy of individuals or even create severe damage in case the data becomes disclosed. To maintain privacy of people of whom the system stores information, the IT system has to ensure information being kept secure and accessible only by authorized persons during and after the transaction.

Techniques like cryptography or Access Control List (ACL) allow to maintain confidentiality effectively. For ACL's to function correctly the concepts of Authentication and Authorization must be understood:

  - *Authentication*, the concept of identifying an individual or system - to identify WHO is accessing.
  - *Authorization*, the concept of enabling access for a previously identified entity (using Authentication) - to identify WHAT can be accessed by WHOM.

- **Integrity** requires an IT system to detect any modification of source information. During Business transactions carried out over the Internet both parties have to agree on a common understanding of the business deal. In case the modification of some related information (like the amount being charged to the credit card) remains undetected, a common understanding does not exist anymore and the trustworthiness of the system is ruined.

- **Availability** specifies that information must be accessible and of integrity in case the particular information is needed. Therefore designers of IT systems have to take precautions for all kinds of outage scenarios, from hardware fault to a hacker attack.

  Availability can be measured by the ratio of time to which a system is operable and is in a committable state within a given interval, which is usually the time a system should fulfill these requirements [55].

2.3.1.2 Security Policies

To help an IT system to fulfill the CIA principles the introduction of a security program is recommended. It may contain a set of

- **Information Security Policies**: The Policy that governs the organization's approach to information security management [79].

- **Procedures**: A document containing steps specifing how to achieve an activity [79].

- **Standards**: A mandatory requirement. Examples include ISO/IEC 20000 (an international standard), an internal security standard for Unix configuration, or a government standard for how financial records should be maintained. The term standard is also used to refer to a code of practice or specification published by a standards organization such as ISO or BSI [79].

- **Guidelines**: A document describing best practice which recommends what should be done. Compliance to a guideline is usually not enforced [79].

- **Baselines**: A benchmark used as a reference point. For example:

  - An ITSM baseline can be used as a starting point to measure the effect of a service improvement plan.
  - A performance baseline can be used to measure changes in performance over the lifetime of an IT service.

  - A configuration management baseline can be used to enable the IT infrastructure to be restored to a known configuration if a change or release fails [79].

- A **Mandatory Security Awareness Training for Staff**: Security awareness is the knowledge and attitude of an organization's member regarding the protection of physical and, especially, information assets of that organization [11].

- An **Incident Response Plan**: Incident response is an organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way to limit damage and to reduce recovery time and costs. An incident response plan includes a policy defining, in specific terms, what constitutes an incident and providing a step-by-step process that should be followed when in case of an incident [32].

- A **Compliance Program**: Ensuring that a standard or set of guidelines is followed, or that proper, consistent accounting or other practices are being employed [79].

Tactical tools like standards, guidelines and procedures are used to achieve and support the compulsory directives of a security policy. The security policy is also considered to define strategic security goals of an IT system and may consist of regulatory, advisory and informative sections. By the use of certain documents like procedures and standards, a security policy will control items of security relevance including

- physical and environmental access control (controlling access to physical resources like servers or routers)
- organizational security (for example: control of who is allowed to grant access to resources)
- backup and recovery procedures
- emergency response procedures

Having a restrictive security policy implemented usually decreases a systems flexibility and increases a systems complexity. By providing only low flexibility, a system might not be that competitive and efficient in business compared to other solutions. High complexity on the other hand denotes that maintenance and administration efforts, including trainings for administration staff, will be higher, thus resulting in higher costs. Therefore it is a necessity to define goals and the target group as one of the first steps while creating a security program. Depending on the defined goals, the correct trade-off between security, complexity and flexibility can be found as described in detail for wireless network solutions by Herrera-Joancomarti et al. [52].

Once all security policies have been defined, it is required to insure their effectiveness which can be achieved by recurring reviews and audits or through conducting overall system security tests [28]. These activities should in equal parts being carried out by the institutions IT staff, the institutions non-IT staff and external parties. For security tests the use of software like vulnerability scanner Nessus is encouraged. Amongst other features, Nessus is able to perform configuration auditing and vulnerability analysis of foreign systems and, amongst others, allows to identify available but not yet applied software patches [113].

2.3.1.3 Definition of Terms

In IT security distinctive terms are used to describe concepts or behaviors. Following term definition is based on the Information Technology Infrastructure Library (ITIL) Glossary [79]:

- **Threat**: A threat is everything that might exploit a vulnerability. Any potential cause of an incident can be considered a threat. For example, a fire is a threat that could exploit the vulnerability of flammable floor coverings. This term is commonly used in information security management and IT service continuity management, but also applies to other areas such as problem and availability management.
- **Vulnerability**: A weakness that could be exploited by a threat – for example, an open firewall port, a password that is never changed, or a flammable carpet. Missing control is also considered to be a vulnerability.
- **Risk**: A possible event that could cause harm or loss, or affect the ability to achieve objectives. A risk is measured by the probability of a threat, the vulnerability of the asset to that threat and the impact it would have when occurring. Risk can also be defined as uncertainty of outcome and can be used in the context of measuring the probability of positive as well as negative outcomes.
- **Asset**: Any resource or capability. Assets include anything that could contribute to the delivery of a service. Assets can be attributable to one of the following types: management, organization, process, knowledge, people, information, applications, infrastructure, and financial capital.

2.3.2 Linux System Security

Linux is an Operating System (OS) using the paradigm of free and open source software development and distribution. The Linux kernel, its main component, was first released by Linus Torvalds in 1991 [114] and is licensed under the terms of the GPLv2 [43].

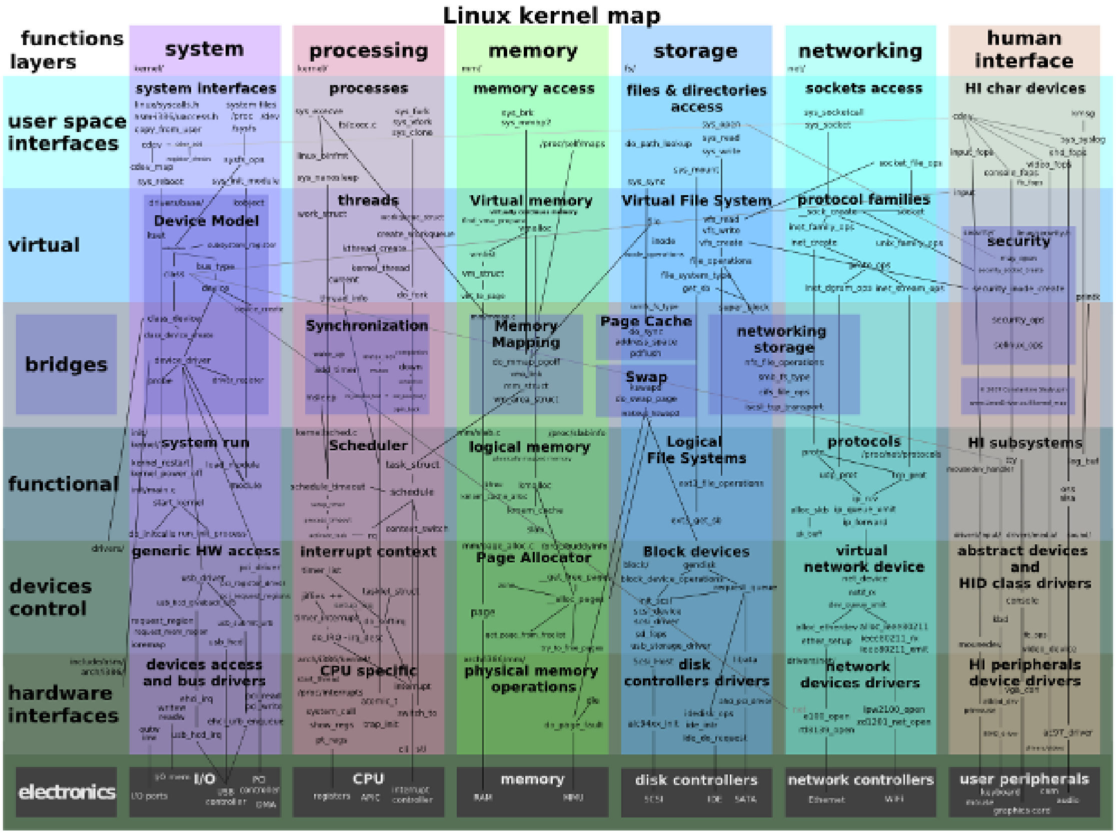

Figure 2.5: Linux Kernel Map [67]

The Linux kernel is responsible for, but not restricted to, managing hardware resources or providing an interface to software. A detailed list of the Linux kernel functions and duties can be found in Figure 2.5, showing that it is in charge of system management functions, processing, memory, storage management, networking and of interfaces to humans (users). These functions are implemented using multiple layers, from the user-near layer *user space interfaces* which provides interfaces for client applications and users, via *virtual*, *functional* and *device control* layers, to the hardware-near layer *hardware interfaces* which offers functionality to manage physical hardware resources. Figure 2.5 aims at illustrating the high complexity of the Linux kernel. Hence, not every relation is explained in detail as these relations are of no direct importance to this thesis.

In terms of security, the Linux kernel implements hierarchical protection domains which is enforced in hardware through the x86 CPU architecture. Protection domains, as described by Schoeder et al. [99], are often illustrated using rings as shown in figure 2.6. Code executed in context of the inner circle (ring 0) does have most permissions and full access to all system and hardware resources. In a modern OS like Linux, only kernel code is executed in context of ring 0. Applications are usually executed in context of ring 3, which grants less permissions in terms of memory access or access to CPU resources.

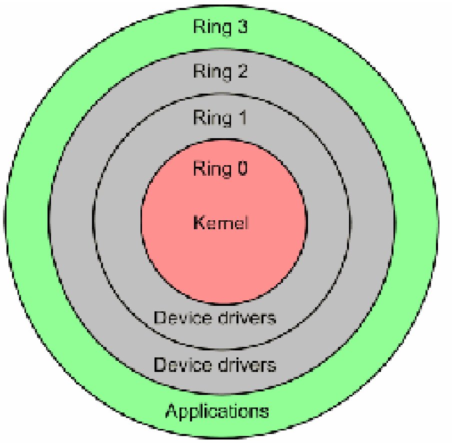

Figure 2.6: x86 CPU rings (see [9])

The Linux kernel also allows to load and unload kernel modules on demand during operation. Using this method device drivers can be loaded into the kernel having them elevated to be executed in context of a lower circle (ring 0-2).

2.3.2.1 Linux Security Concepts

To achieve comprehensive security on a Linux OS requires to understand its incorporated security mechanisms and permission control. The OS offers multiple mechanisms which are briefly described in the following:

- **User / User-group Concept**. Linux was designed as a multiuser system. Hence, it is possible for multiple users to work on one single machine in different roles [115]. These roles are

  - a *normal* user
  - an *administrative* user - on Linux usually called `root`.

  Every User has a unique Identifier (ID), an assigned home directory and is member of at least one user group. Every application executed on a Linux system will run in the context of a user, which determines the application permissions on the host system.

  Linux systems use file `/etc/passwd` [83] to store and manage their known user accounts. Each line of this file represents one user account and defines common parameters of this account, including the users *primary group* or the *home directory path*. It also defines if an account can be used for interactive logon. If so, a command line interpreter program (shell), starting at logon time, is defined.

  In older Linux versions, the actual logon password was stored in this file too. Modern Linux systems make use of cryptographic hash algorithms and store their passwords in separate files which can only be accessed through the system and its root user.

  User administration for multiple Linux-based systems within one network environment may also be accomplished using directory services [53] like the Lightweight Directory Access Protocol (LDAP) or the Network Information Service (NIS), thus allowing to administer user accounts more efficiently.

- The **File Permission Concept** helps to control access to files and folders. Each object on a file system, including files, folders or symbolic links will have

  - Permissions
  - an owner user
  - and an owner group

  assigned. The permissions are illustrated by a character string (for example `-rwxr-----`) and can be split up into permission groups. The first character does not define any permission as it defines the type of the object stored on the file system (a regular file is indicated by `-`, a folder by `d` and a symbolic link by `l`).

  The remaining nine characters of the permission string are split up into three groups of three characters. Each of these groups indicates permissions in the same manner, but are applied to different sets of users. These groups are

  1. *User*, applying permissions to the owner user
  2. *Group*, applying permissions to every user of the owners group
  3. *Others*, applying permissions to all other valid users

  Each three-character subset of the permission string indicates which permissions a group of users will have.

  1. Character: `r,` grants permissions to read a file system object.
  2. Character: `w,` grants permissions to write on a file system object.
  3. Character: `x,` grants permissions to execute executables or scripts stored in the file system.

  In case a − is written instead of the above listed characters, the permission is not granted. This demonstrates that the Permission string `-rwxr-----` grants full permission for the owner user and read permissions for all user members of the owner group. It also points out that this permission is applied to a regular file [14]. For modifying permissions, owner user or owner group, the linux commands *chmod* and *chown* can be used.

  For situations where the traditional file permission concept is not sufficient, the Linux kernel allows the use of ACLs, for which file and folder permissions can be more detailed. For example, an ACL allows to grant permission to append a file but not to delete it.

- **Remote Access**. Linux-based systems make use of text terminals for communication with its users. Shell programs like `bash` or `sh` allow an user to enter commands and to receive feedback from the system instantly. Text terminals on Linux-based systems allow to redirect its in and output to different destinations, including the systems console or remote applications utilizing network protocol TELNET [70] or Secure Shell (SSH) [127]. SSH encrypts all payload data transfered between the Linux-based host and the client application, which TELNET does not.

  In addition the text-based terminals, Linux-based systems allow to run graphical enhanced applications on remote clients using the X window system [73].

- **Startup Process and Linux Run levels**. After hardware has been initialized (Basic Input/Output System (BIOS)) and a boot loader like GNU [43] Grand Unified Bootloader (GRUB) has been started, the Linux system passes through multiple stages:

  - The **boot loader** is stored in the Master Boot Record (MBR) of a physical storage and is responsible to starts the OS kernel.
  - The **kernel** initiates all system hardware drivers and mounts the root file system `/`.

  - **Init** is the first application started by the kernel and responsible to bring the system into the desired run level.
  - The **Run level** describes the mode of operation and determines which applications and services are supposed to run. Valid run levels are:

Run level	Description
0 - Halt	Shuts down the system
1 - Single-User	Single-User operation for administrative tasks
2 - Multi-User	Multi-User operation without network services
3 - Multi-User with Networking	Multi-User operation
5 - Graphical enhanced	Multi-User operation with network services and display manager (X Windows System)
6 - Reboot	Restarts the system

- **Security-Enhanced Linux (SELinux)** is a Linux feature enabling the use of extended access control security policies implemented in form of a set of Linux kernel modifications and some user-space tools [13]. It is a project of the United States National Security Agency (NSA) and the SELinux community and enforces access control policies that confine applications and services to a minimum amount of privileges required to do their job. SELinux can operate in three modes, which can be configured using its main configuration file:
  - **enforcing** will fully enforce SELinux policies
  - **permissive** prints warnings but does not enforce the policies
  - **disabled** will fully disable SELinux
- **Logging Mechanisms** and **Monitoring**. Linux enables the usage of logging mechanisms for reasons of detecting hardware problems, user problems or security breaches to name only a few. Software packets `syslog` or its superior version `rsyslog` implement these logging functions and collect messages from applications as well as from the Linux kernel. Per default the Linux system stores the log messages into files located in the file system folder `/var/log`, file `/var/log/messages` for example.

  `syslog` software also offers the possibility of sending the log messages of a system to a central network logging server.

  Similar to other network devices, Linux offers the possibility to retrieve information about a system's vital state using Simple Network Management Protocol (SNMP), which could be used by centralized network monitoring solutions.

2.3.3 Network Security

Network technologies are the foundation of telecommunication services and enable remote access to applications as well as content management and distribution. Within this context, recommendations X.700 [120] and X.800 [122] of the ITU-T provide a general framework to ensure the security of telecommunication services [1].

2.3.3.1 Security Objectives

Security exposures can affect user data, applications and the network infrastructure itself. ITU-T recommendations X.509 [118] and X.800 [122] identify several types of information threats (attack patterns) that can be classified as follows [1]:

1. **Passive Attacks**
   - Interception of the identity of one or more of the participants by a third party with a mischievous intent, violates the entities confidentiality.
   - Data interception through clandestine monitoring of the exchanges during a communication by an outsider or an unauthorized user, violates its integrity and confidentiality.
2. **Active Attacks**
   - Replay of a previous message, in its entirety or in part, after its recording (violating the security principal integrity).
   - Defective or criminal manipulation of the content of an exchange by substitution, insertion, deletion or reorganization of user's data exchanged in a communication by a non-authorized third party (violating integrity and availability).
   - Users repudiation or denial of their participation in part or in the entirety of a communication exchange (violating integrity and availability).
   - Misrouting of messages from one user to another (violating confidentiality).
   - Analysis of the traffic and examination of the parameters related to a communication amongst users (e.g. absence or presence, frequency, direction - violating confidentiality).
   - Masquerade, whereby one entity pretends to be another entity (violating integrity and confidentiality of the other entity).
   - Denial of service and the impossibility of accessing the resources usually available to authorized users following the prevention, interruption of a communication or the delay imposed on time-critical operations (violating availability).

Based on the proceeding threats, the objections of security measures are [1]:

- Prevent an outsider other than the participants from reading or manipulating the contents or the sequences of the exchanged messages without being detected. In particular, this third party must not be allowed to play back old messages, replace blocks of information, or insert messages from multiple distinct exchanges without detection.
- Impede the falsification of payment instructions or the generation of spurious messages by users with dubious intentions. For example, dishonest merchants or processing centers must not be capable of re-utilizing information about their clients bank accounts to

generate fraudulent orders. They should not be able to initiate the processing of payment instructions without expediting the corresponding purchases. At the same time, the merchants will be protected from excessive revocation of payments or malicious denials of orders.

- Satisfy the legal requirements for valid contracts to allow conflict resolution, particularly in the area of consumer protection and privacy protection.
- Assure access to the service according to contractual terms.
- Give the same level of service to all customers, irrespective of their location and the variations in climate, temperature, humidity, erosion, etc.

2.3.3.2 Security Implementation - Infrastructure Considerations

In order to achieve the security objectives, it is necessary to have a secure network environment. A brief overview of security techniques to create a secure network environment follows:

- **Firewalls** are essentially filters which prohibit forbidden information from passing through while allowing approved information [126]. Firewalls can be hard- or software-based depending on the protection goal and the infrastructure.

  Within a core network environment they are most effective when installed as network entrance points by defining both, incoming and outgoing ACL's for every network segment attached. In this context, network entrance points are defined as the only gateway to access a certain network segment in which firewalls also take over the functionality of network routers. Firewalls do require a high degree of maintenance, as the ACL's may have to be altered for every service installed, but are an effective way to stop unauthorized access before even entering a network.

- **Network Address Translation (NAT), Port Address Translation (PAT)** allows to make certain services accessible from outside networks without granting direct access to the service's residual internal network using virtual addresses and ports on the outside network behind which the real service is hidden. The network device configured for NAT or PAT (mostly routers or firewalls) picks up the request from the outside virtual address and port and forwards it seamlessly to the inside network.

  This concept can not only protect a single application or service, it may also be used to hide whole network segments and all clients attached to such a segment. This results in disallowing the direct access to an internal client by request generated from outside networks.

- **Demilitarized Zone (DMZ)**, borrowed from military terminology, is a physical or logical subnetwork which is located between different network areas, usually these are the public Internet on the one side and the internal private network on the other side. This allows to create a layered defense against malicious network attacks [12].

  As shown in Figure 2.7, services available to the public (e.g. web based services) should be hosted within this intermediary and isolated DMZ network. In case a service becomes exposed, the attacker can only access resources within the DMZ network but is still prohibited to access internal resources.

- **Proxy Servers** allow to inspect application layer data transfered between networks. A HTTP proxy server can either inspect only the Uniform Resource Locator (URL) requested in HTTP headers or inspect the whole data content transfered. With the use of

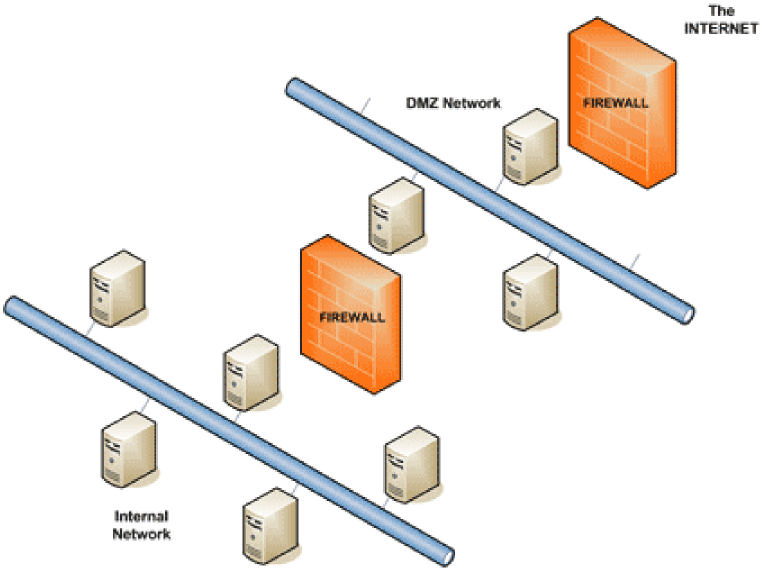

Figure 2.7: DMZ scheme (see [18])

Cisco-developed Web Cache Communication Protocol (WCCP) [103], Proxy servers can be implemented in a transparent manner.

Proxy servers can be considered as OSI layer 7 (application layer) firewall for multiple protocols. They are introduced to control an application's access to the Internet and may utilize security implementations like virus scanning capabilities.

- **Quality of Service (QoS)** enables a network router or any network device in charge of IP packet forwarding to prioritize based on multiple indicators. An indicator could be the source or destination TCP port of a data packet. Also the Differentiated Services (DS) field [77] in the IP header can request a network routing device to adjust a packet handling priority.
- **Network Access Control (NAC)** mechanisms control if client devices are allowed to connect to a private network. The decision whether a client device is authorized or not can be based on the clients Media Access Control (MAC) Address or by implementing IEEE standard 802.1x [105].

  NAC mechanisms can be used in wired and wireless network environments and allow client device authentication and automated Virtual Local Area Network (VLAN) assignment.
- **Encryption** - Today most computer network environments use shared media to transfer data from one point to another. Shared media implies that others are able to see, capture or manipulate data. Therefore, multiple methods to encrypt information transfered on data networks have been invented and are endorsed to be used for confidential data at least.

Once considering/designing the overall security of a network infrastructure, a balanced set of security measures is required. They need to be adjusted to one another allowing to create contemporary security measures which have to be defined in detail in governing documents (using security procedures for example (see Section 2.3.1)).

2.3.3.3 Security Implementation – Monitoring

In big network environments unexpected changes and troubles are often discovered very late and network administrators often tend to loose track of what is going on within their networks. Therefore, monitoring systems are a necessity for secured network operation.

These systems help to identify possible problems early on or even before they become a real problem with impact on users or other systems. They also allow to troubleshoot existing problems more efficiently by providing information about the vital state of the network and systems attached.

Many network monitoring solutions are based on SNMP [26], which provides basic functionality to collect vital data about remote network devices. Monitoring solutions heavily differ in the way they store and present the collected data. Some solutions focus on providing graphed historical data (utilizing a Round-Robin Database (RRD)) whereas others focus on real time visualization and alerting.

2.4 VoIP Security

This section illustrates security concerns regarding VoIP systems by highlighting weak spots and identifying attack patterns. Founded on general IT security concerns (previous Sections 2.3.1 and 2.3.3), this chapter gives an overview of attack patterns on VoIP systems.

2.4.1 Attack Patterns on IT Infrastructure to Harm VoIP Systems

This section explains attack patterns with the intention of harming a VoIP system, while making use of IT infrastructure vulnerabilities to achieve their goals.

2.4.1.1 Man-in-the-Middle (MitM) Attacks on VoIP Systems

From a logical point of view, MitM attacks are conducted by malicious devices placing themself between two communicating devices in a way, that the whole data stream between these two non-malicous devices can be intercepted or even modified [63].

Alltough specific MitM attacks can also harm other systems than VoIP systems as well, every MitM attack does have a goal to achieve which could be to take illegal advantage of a VoIP system.

Network based MitM attacks use vulnerabilities of network infrastructure devices and protocols to redirect data packets. One protocol heavily targeted in the context of a Local Area Network (LAN) is the Address Resolution Protocol (ARP). It is stateless and serves the purpose of linking hardware MAC addresses to IP addresses. Bhirud et al. [17] identified attack methods based on ARP like

- *ARP Spoofing*, which is the process of creating and injecting fake ARP requests and ARP reply messages into the network.
- *ARP Cache Poisoning*, which aims at modifying the ARP cache of a host or network device, which is updated once an ARP reply message is received. In case such a reply message has been modified or spoofed, the host will store wrong information in its ARP cache and the host may contact a malicious device instead of the correct one.

One can also distinguish in regard to the context the MitM attack is conducted [63], which could be either in context of a LAN or in context of the public Internet. Attacks can be conducted in both contexts, but differ in the approach taken. Due to the used routing mechanisms data packets from a single source sent to the same destination may take different routes through the Internet. In case of using public Internet, a MitM attack can be mounted on the manipulation of routing protocols, whereas this method is noneffictive in the context of a LAN which makes use of static routes only.

In case a VoIP system is exposed to a successful MitM attack, all actions initiated by victims of that attack can be subject to eavesdropping, manipulation or denial.

2.4.1.2 Denial of Service (DoS) Attacks on VoIP systems

The goal of a DoS attack is to restrict the operation of the attacked service or device using brute force. This can be achieved through generating simple capacity overload, network congestions or by an aligned modification of network payload data [63]. Howard [54] defines DoS as ..., *if* *a service is supposed to be available and it is not, then service has been denied.*, not specifying the cause of the denial.

A capacity overload can be created by an attacker by sending excessive amounts of requests to the target within a short period of time. As the targeted server system is requested to handle each of these requests while consuming system ressources, the target may be caused to run out of ressources hence dening further requests. Attacks of such kind are mostly implemented in some kind of automated manner and may not be restricted to only one device creating malicious messages (Distributed Denial of Service (DDoS) attack).

A overload of local system ressources does not necessarily have to be the only impact on the targeted server in case of a DoS attack but may also have the effect of jamming the network uplink.

Fuchs et al. [44] illustrated the possible impact of DoS attacks on VoIP based emergency phone services.

2.4.1.3 Eavesdropping

Unfortunately, there are also a variety of ways that an attacker can compormise the privacy of VoIP conversations.... With the appropriate access to the right point in your network, an attacker can perform a variety of attacks beyond simply listening to your conversations [37]. Once an attacker has gained access to the network by compromising a network switch or a phone for example, eavesdropping allows to capture and record data transfered within the network. The collected data can be used for analysis and allows an attacker, according to Endler et al. [37], to gain the following information:

- *Trivial File Transfer Protocol (TFTP) Configuration Files*: As many IP Phones rely on TFTP servers to retrieve their configuration, this configuration, which may store passwords, is exposed to the attacker.
- *Numbers*: An attacker may collect numbers from all incoming and outgoing calls. This information can be used to mount more complex attacks like signaling manipulation or Spam over Internet Telephony (SPIT)
- *Call pattern*: Instead of just collecting numbers, call pattern tracking enables the attacker to know who is talking to whom. With this data, corporate espionage becomes possible.
- *Conversation Analysis*: Capturing of in-stream information like the actual voice stream or Dual-tone multi-frequency signaling (DTMF) tones allows an attacker to get in possession of possibly confidential information.

2.4.1.4 Server manipulations

Manipulations on the server system may open potential points which an attacker could use to mount an attack. Before this kind of manipulation can take place, the attacker has to have shell access to the server system. Once this is achieved, following actions may be taken:

- *Server process manipulation*: Asterisk software utilizes processes and threads to complete its work load. These processes are executed on the OS and can be influenced from the OS level. By the forceful removal of such a thread, unexpected system behavior is created.
- *Change of executeables*: Executeables and libraries stored on the server systems Hard Disc Drive (HDD) implement Asterisk's functionality. Modification of these files may limit or change Asterisk's behavior.

- *Configuration changes*: Changes of Asterisk's configurations on VoIP allows an attacker to change Asterisk's behavior and allows the attacker to open up touch points to mount other attacks. For example, a new subscriber number for a malicious user can be created.
- *Codec manipulation*: An attacker may be able to change the encoding methods of Asterisk with the effect of media streams becoming unusable.
- *Disk overflow*: In case an attacker fills the local HDDs with data, regardless of the type, Asterisk server may not be able to store any more information causing DoS.

2.4.1.5 Media Stream Manipulations

In VoIP based systems, as introduced in Section 2.1, RTP protocol is used to carry media streams. Due to the use of stateless UDP, the recipient has no possibility to check whether all data packets sent by the originator, have been received. RTP packets might be lost en route, might arrive at the receiving endpoint duplicated or out of sequence as they transit the network. The receiving end point incorporates an audio buffer, which rearranges the packets to create the best playback as possible. By spoofing the RTP header data, the receiving end point can be tricked into rejecting messages from the non-malicious end point in favor of RTP packets of malicious end points [37].

2.4.2 Attack Patterns on VoIP systems

The design of VoIP protocols like SIP also enables attackers to find appropriate points of contact to mount an attack on the system. This section will give a brief overview of attack patterns which mount their attacks on the designed behavior of SIP.

2.4.2.1 Denial of Service (DoS) Attacks based on VoIP capabilities

Additionally to the effects of DoS attacks described in the previous section (Section 2.4.1.2), DoS combined with MitM attacks may also target SIP protocol [30] instead of the network infrastructure.

Key to such an attack is the introduction of malicious messages which cause either the subscriber or server to tear-down the SIP session while simultaniously misleading the other SIP session participant into believing that the SIP session is alive. This participant hence is about to wait until a timeout is reached and the SIP session is considered dead.

A well-formed *CANCEL* message introduced into the SIP session setup process (between step 3 and 4 of Figure 2.3, for example) will prohibit the session from being established successfully. On the other hand, introducing message *BYE* can cause an already existing SIP session to perish unexpectedly.

In addition to the already presented types, Chen [29] identified following possible DoS based attack patterns.

- *Legitimate Message Flooding*: An attacker attempts to consume all available system or network ressources for legitimate applications to become unuseable. Ressources in question are, for example, the network bandwidth or the maximum number of concurrent TCP/UDP connections [37]. Today, flooding attacks have become common practice on the Internet as more and more DoS tools are available. As a reference, packetstormsecurity.org [81] lists more than 250 DoS tools today.

- *Destributed Reflection DoS*: A Distributed Reflection Denial of Service (DRDoS) attack uses legitimate hosts called *reflectors* to send a large number of packets to a victim by the use of IP spoofing [117]. The attacker sends packets with a spoofed IP address, using the IP address of the victim as source to the reflectors, which will reply to the victim. This can cause a lot of simultanious requests and in turn causes DoS at the victim.

- The introduction of *Malformed Messages*: Malformed messages can be described as messages holding information which identifies the message as a SIP message, but does have bogus information stored in its header fields. For example, a modified *To* field disallows Asterisk to identify and route the session setup request accordingly.

  Such message modifications may also be directed towards Asterisk's SIP parser. The parser has to cope with manipulations like a header field containing unexpected special characters or an unexpected long length of the header field data.

2.4.2.2 Session Hijacking

By introducing well-formed *301 - Moved Permanently* or *302 - Moved Temporarily* messages, an attacker may be able to redirect incoming calls for subscribers to another malicious device. With these messages, the server is informed that the subscriber is now reachable under another network address without the real subscriber noticing [63].

2.4.2.3 Identity Theft

The Theft of Service pattern provides an opportunity for hackers to gain access to the VoIP network by imitating subscribers and/or seizing control of terminal devices and performing free calls [82]. This attack could be accomplished using several techniques, starting from simple placement of calls using an unattended device, to the theft of the identity of a legitimate user. In case the attacker uses the identity of the owner without consent, the charges for the calls will still be assigned to the owner [82]. This behavior is known as toll fraud, which is described by Nassar et al. [75] as *an attack scenario* in which *the malicious user aims to bypass billing*.

2.4.2.4 SPIT

Spam over Internet Telephony (SPIT) is similar to email SPAM that affects VoIP. *SPIT, in this context, refers to bulk, automatically generated, unsolicited calls* [37]. Telemarketing is certainly annoying as well but not considered as SPIT. SPIT, unlike many others, does not rely on faulty configurations, software bugs or other issues which could harm the system's integrity.

3 Business Informatics Fundamentals

This chapter establishes the fundamental understanding of business affairs concerned with this Thesis. First, basic considerations for introducing new innovative IT solution and the scope of application of VoIP solutions are presented. Afterwards, the process of security evaluation used within this thesis is described in detail.

3.1 Use of IT in Businesses: Motivation

In today's business environment, the main goal of profit oriented Businesses and Organizations is to earn a lot of money by trying to minimize expenses while maximizing their income. Any new IT solution will require spendings, which conflicts with the minimizing expenses approach.

Not only the setup, but also the evaluation process and the operation of IT solutions require the input of money: A Gartner Perspective on IT spending shows that the *worldwide end-user spending on IT* in 2010 was approximately 3,30 trillion USD with an increase of 3,3 percent year on year [46].

But IT does not only create costs, IT also helps to boost an organization's productivity by, for example, automating work flows which enables employees to handle a bigger work load. Deliberated investments on innovative IT solutions can create Return on Investment (ROI) already after a short period of time. Investments into new IT solutions often entail that changes to the current business processes are made in order to being able to merge the new solution into the current environment and hence, utilize its full potential of the new solution. Mathur [68] proposes a way to adopt the Business Processes to such changes using a *Business Process Transformation Grid*.

There are a few global and strategic decisions an organization should consider and take before any investment decision can be made:

- **How to serve business needs at best**

  Before any IT solution can help to create income, the organization has to know what the exact expectation of the IT systems are and which functionality these systems are expected to deliver. A corporate IT strategy should answer these questions. The business requirements have to be defined at first and an appropriate IT strategy has to be built around them to support the business in the best way possible. Within companies, IT can be seen as internal service provider supporting the companies core business.

  Apart from *what* is expected from IT systems, the IT strategy should also answer *how* its realization is going to be organized. This includes decisions on whether a centralized or de-centralized setup is implemented or how the local IT staff is organized, for example.

- **Degree of Outsourcing**

  The operation and maintenance of a local data center infrastructure can be quite expensive for small organizations. The alternative to this is the outsourcing of hardware infrastructure or IT services to an IT data center or cloud service providers. As *Outsourcing is* *proven as a good idea for many firms that need to reduce operating costs and improve information technology operations.... Many companies will be hurt by hastily following the herd down the outsourcing path without thoroughly evaluating the benefits and risk associated with it* [97], the decision on this question must be a sophisticated one.

A high degree of outsourcing also results in a high dependence on other companies (e.g. outsourcing providers) which may generate risks to the company. The only binding object with the provider might be a legal contract but no influence on the provider's strategic decisions or on relations to other contractors can be taken. Outsourcing helps to reduce local hardware and maintenance cost and therefore requires less IT staff on location. On the contrary, having things done by an outsourcing partner and having less local IT staff will inevitably result in the decline of intra-company know-how about the IT systems in operation. Beulen et al. [16] identified knowledge transfer and management as an important factor for a successful outsourcing operation.

The decision on the degree of outsourcing will also help to decide by whom and how the local IT environment shall be supported and maintained and will determine how the local IT organization has to be organized.

- **Product decisions**

  Before an IT solution product can be acquired, the organization has to define how such decisions should be made, what the motives are, how a vendor screening can be conducted and which characteristics an IT solution should have to be able to make an educated decision.

  Making individual decisions is often a fuzzy process which lacks a clear work flow structure. To improve this, Petrusel et al. [85] proposes a Data Decision Model which helps to overcome certain drawbacks of individual decision finding processes.

3.2 VoIP Solutions: Benefits and Scope of Operation

As mentioned in Section 3.1, running a successful business requires to keep control of spendings and minimize costs while maximizing the company's productivity at the same time. VoIP systems do not only allow to minimize operation costs, but they can also help to increase the company's productivity.

VoIP solutions emerged as fixed part in today's IT infrastructures and start to evolve in mobile networks as well. A Gartner survey [47] shows that *more than 50 Percent of mobile voice traffic will be carried using end-to-end VoIP by 2019.* This highlights the future potential of such a solution.

Amongst other things, VoIP solutions utilize the following features and functions which help organizations to optimize:

- **No Dedicated Infrastructure required.** As VoIP solutions can use regular IP networks, no setup and maintenance of a dedicated voice network is required. Infrastructure setup is easy and creates only low costs due to the variety of available products. This not only applies to VoIP equipment like Phones, but also to the network infrastructure devices like switches or routers. The use of IP network technology also enables a flexible network layout and hence, offers more flexibility to end users.
- As VoIP solutions heavily utilize **global standards** defined by the IETF and ITU-T, license fees for the use of this technology are only seldomly applied. Depending on the products selected, the costs for operating such a solution can be kept very low.
- VoIP solutions will increase the user's **acceptance**: Due to the use of standard IP network infrastructure and the variety of available end-user devices and applications, it is possible to benefit from this technology virtually everywhere and at any time.
- Easier **work environment integration** is only one of the features VoIP systems can offer. A subscriber may only have one phone number, but is reachable independent of his location. This could be at the office, at home (tele-working) or somewhere else (mobile phone).
- VoIP systems offer **more than voice**: As such systems utilize a media stream for transferring media content to other subscribers, not only voice data may be transfered on VoIP systems. This media stream can transfer many types of encoded media and enables to conduct video calls, make conference calls or make use of instant messaging functionality which can help to create a new experience to users.
- Use of **bandwidth is more efficient**: As VoIP system media streams can transfer different payload data encoded by different codecs, the configuration of the codec can be adopted to network quality pre-conditions. Therefore it is possible to utilize the available network connection in the most efficient way which can save costs on Internet network connections.

3.3 Software Development Life Cycle (SDLC)

The Systems Development Life Cycle (SDLC) describes the life cycle of software applications and supports software development projects during any phase.

According to Lewis [64] the SDLC *involves various phases which generally include planning, definition, requirements, design, building, implementation and maintenance*. During each of these phases, a project manager or IT professional is required to come up with deliverables, which depend on the requirements of the project. At the planning phase for example, the deliverables may include the project charter and SDLC templates [64].

To serve software development the best possible way, each organization has to adapt the SDLC according to their needs. Therefore, also Asterisk development is supported by an aligned SDLC process.

3.4 Security Evaluation Approach

In order to make improvements on a systems security it is required to analyze the system's behavior including all running services and programs as well as every external communication. The possible impact on the system in certain situations like while receiving a malformed message is needed to be evaluated.

Organizations often tend to focus on threats rather than on protecting their assets, which often leaves the entire system vulnerable [109]. Stango, Prasad and Kyriazanos [108] proposed an eight step model for threat analysis focusing on asset protection which is adopted below to suit to the business case discussed in this thesis. This model aims at evaluating possible security threats and helps to construct a mitigation plan using risk management.

It consists of the following 8 steps:

Step I: **Description of the system: Use Case (UC)s**

To describe a system it is required to understand every component and every interaction.

Unified Modeling Language (UML) UCs [6] have been chosen to create a holistic view of the Linux system implementing Asterisk, covering all concerned software services running. The UCs show what the system must be able to do by illustrating the interactions between UCs and actors involved. A descriptive table of each UC contains all information related to the system and the UC. It has to include the items and has to answer the following questions:

- **Goal** - What is the desired achievement?
- **Precondition** - What is required to be able to execute the UC?
- **Success condition** - Under which circumstances will the UC be executed successfully?
- **Fail condition** - Under which circumstances will the UC fail?
- **Actors** - Who is concerned?
- **Triggers** - Which event causes the UC to be executed?
- **External Resources** - Which external resources like configuration or log files are necessary and how are they accessed?
- **Service** - Which service or program is responsible for the execution?
- (optional) **Remark / Extensions**

Step II: **Analyze the technical background of the UCs**

To be able to create a comprehensive system description, timing and sequence informations have to be added to each UC. Sequences are illustrated by Sequence diagrams or textual descriptions showing object interactions according to their timing sequence.

Also, messages exchanged between objects including the means (network protocols, device drivers) and options (use of encryption, use of authentication, session persistence mechanisms) of transportation are described.

Step III: **Identify Assets**

This step is about determining everything within the system which could be damaged or violated.

Services are executable programs which are responsible for the provision of functionality described within UCs during step I and II. While every UC will be handled by one or more services, a single service might host the functionality of multiple UCs. In case a service is not able to further serve the functionality of one of its UCs, the service has to be assumed to be failed.

Defining each service as asset is proposed. Services create business value by providing functionality described by the UCs. Once a service fails, it has no business value anymore, showing that services are the actual asset of any system.

In addition to the system services, collected and stored data can be classified as assets of a system as well. Data assets include configuration files, accounting information, log files or customer data like addresses or billing instructions.

Step IV: **Determine Threats**

Threats can be identified by going through each previously defined UCs and creating threat hypotheses that violate confidentiality, integrity and availability [74], resulting in a tabulated threat profile. A threat source can be manifested as any circumstance or event with the potential to cause harm to a system.

The threat profile shows details for every threat including

- an ID
- a name
- a classification
- a description

In general, threats can be classified into six classes based on their effect [111] :

- **Spoofing** - Using someone else's credentials to gain access to otherwise inaccessible assets.
- **Tampering** - Changing data to mount an attack
- **Repudiation** - Occurs when a user denies having performed an action but the target of the action has no way to prove otherwise
- **Information disclosure** - Disclosure of information to a user who does not have permission to see it
- **Denial of Service** - Reducing the ability of valid users to access resources
- **Elevation of Privilege** - Occurs when an unprivileged user gains privileged status

The threat profile for a system describes all potential attacks, each of which needs to be mitigated or accepted.

According to Myagmar et al. [74] access points are what the attacker is going to use to gain access to assets. Examples of access points are open network sockets, configuration files or hardware ports.

Step V: **Determine Vulnerabilities**

This step aims at developing a list of system vulnerabilities that could be exploited. This can be achieved by analyzing the threats from the threat profile described in detail before.

Step VI: **Asset Mapping**

This step determines the importance of assets identified before and the risks that the owner of the asset is willing to accept [91], thus allowing to prioritize them by assigning values.

While assigning values to assets, personal priorities of people can be different. The values chosen should also represent an asset's importance based on a systems requirements and anti-requirements. Therefore, three different values have been proposed by Stango et al. [108]:

- **High** - Assets with this value have to be protected with a high level of security. They are directly linked to the control of the system or have a big financial value.
- **Medium** - Assets providing non-critical, but still important, common services with an intermediate financial value.
- **Low** - Value of assets of minor importance

Step VII: **Risk Management**

Combining and analyzing both, the importance of identified assets and the vulnerabilities of system components, results in a risk estimation (as defined in Section 2.3.1.3) for every identified vulnerability. This allows to state at which degree a system can get harmed after and during a possible exploit of a threat.

As the potential financial loss is directly attached to the degree of security violation caused by an exploited threat, it is required to minimize security vulnerabilities and maximize a system's overall security.

This step suggests to put a virtual *price tag* in terms of risk levels on each identified vulnerability. Four risk levels are proposed:

- **No Risk** - No possible harm to the system can be created
- **Low Risk** - Risk is present although unlikely to become exploited; financial loss is limited
- **Medium Risk** - Risk is present; medium financial loss is likely
- **High Risk** - Risk is present and easy to become exploited; financial loss can be devastating

Step VIII: **Mitigation Plan**

The Mitigation Plan defines which actions to be taken to mitigate a certain risk and selects the appropriate countermeasures.

According to Myagmar et al. [74], a risk can either be accepted, transferred or mitigated. In case a risk has been accepted by the responsible entity or has been transferred, it must not be included in the mitigation plan. The business owner is able to decide that a risk at stake does not have to be mitigated if either the mitigation is to expensive or the possible impact is too low. In this case the risk is accepted. In case the identified risk can be shifted to another system like a firewall, the risk is considered transfered.

The classification of threats listed in step IV can be used when deciding on a mitigation mechanism of a specific threat [111].

Once a mitigation strategy for a risk has been chosen, the required budget, time and resources for its implementation must be evaluated. As the available money in organizations is often subject to budgetary or other constraints, it is not always possible to mitigate all risks. Hence, in real life a cost-effective balance between risk mitigation costs and risks to be taken has to be found.

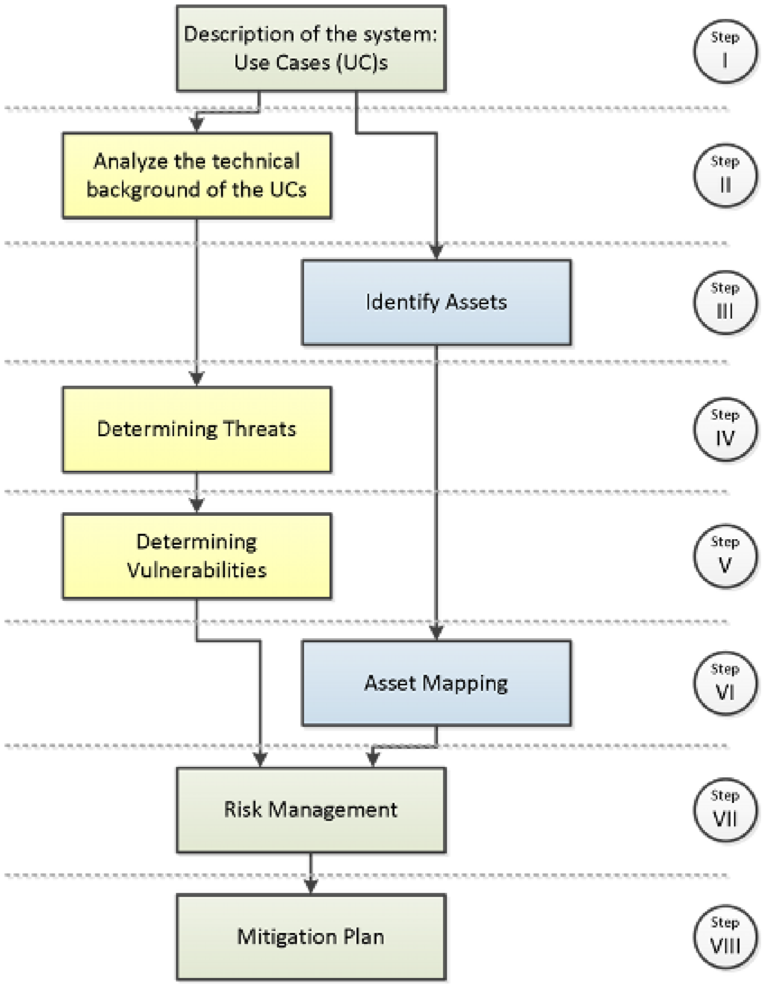

Figure 3.1: Security Evaluation Approach

Figure 3.1 shows a schematic flow diagram for the proposed security evaluation approach. The line connections illustrate dependencies between steps. For example, the results of step I are a precondition to steps II and III for being able to carry out the next step of the analysis.

This approach is used to evaluate a system while operating in a designed behavior. Although the approach would be capable of evaluating non-designed behavior, this thesis focuses on the described and expected behavior. Non-designed behavior can have several causes like software bugs or physical damage. Such influences can be addressed by regulations within IT security policies or by the implementation of quality insurance of software projects and are not covered in detail by this thesis.

4 Reference System Description

This thesis will apply the security evaluation approach described in 3.4 to a reference system installation.

4.1 Hardware and Software Configuration

This section describes the hard- and software configuration of the reference system in use which is hosted within an virtual environment. The system has a dedicated public IP address assigned and is connected to the Internet.

System Environment

Item	*Description*
OS	CentOS release 6.0 (final) [27]
Linux Kernel	version 2.6.18-194.el5
Software Management	RPM Package Manager (RPM) [59] based YUM
Hardware	VMware[123] based 1 virtual Central Processing Unit (vCPU) 1024M Random Access Memory (RAM) 10G HDD
HDD Partitions	/dev/sda1 (100M): Mount point /boot /dev/sda2 (9.7G): Mount point / no swap partition
Database	PostGre SQL 9.1.2 [90]

Table 4.1: Reference System: Environment

Asterisk

Item	*Description*
Version	1.8.6.0
Configurations	out-of-the-box using: plain text configuration files (stored in /etc/asterisk) databases (PostGre) for CDRs and Extended Logging
Protocols Supported	SIP
Subscribers Protocols	SIP
PSTN Uplink	SIP trunk
Management	CLI Access from localhost only Web Access (asterisk-gui package)

Table 4.2: Reference System: Asterisk

Communication Settings

Item	*Description*
Signaling	Use of unsecured signaling protocol (SIP)
Media	Use of unsecured media transport protocol (RTCP, RTP)
Database	Access to database restricted to localhost Asterisk – database communication unencrypted
Web Access	No access
Console Access	Use of encrypted communication (SSH)

Table 4.3: Reference System: Communication Settings

4.2 Description of the System: UCs

This section describes the behavior of the reference system as described in step I (section 3.4) and shows the system's functional behavior and its core functionalities. Each core functionality is explained by an UC showing how the system behaves under normal operation and normal circumstances.

This is the basis to the following security analysis as it is important to know what the system is doing and how the systems achieves its operation goals. This description also shows the system's components involved during operation. The knowledge about the used system components is essential as they need to be addressed during the security analysis as well.

ID and **name**	**Description**
1 - Client registration	Client registers on the system
2 - Call internal via SIP trunk	Establishes a media connection between two local system subscribers
3 - Call out-bound via SIP trunk	Local subscriber establishes a media connection with a foreign system subscriber, connecting via SIP trunk.
4 - Call in-bound	Foreign system subscriber establishes a media connection with a local system subscriber, connecting via SIP trunk.
5 - Call mailbox	Access of a local subscribers to their voice mail box
6 - Log access	Access of administrators and authorized users to Linux and Asterisk logging facilities
7 - Asterisk configuration change	Change of Linux and Asterisk configuration parameters by administrators or other authorized users

Table 4.4: List of UCs

Table 4.4 shows a list of identified UCs with a brief description followed by Tables 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 describing details of each UC.

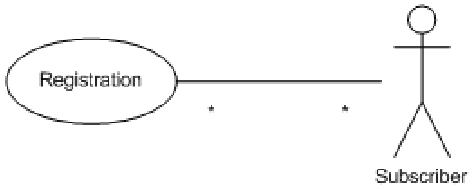

Figure 4.1: UC 1 - Client registration

UC 1 - Client registration (Figure 4.1)

Goal	Client is known by Asterisk server, holding following information • subscriber's name • subscriber's extension number • contact information (IP Address) • VoIP capabilities
Precondition	• Asterisk service is running properly • Client device is connected to network • Client device is capable to use SIP
Success condition	Subscriber information is stored in Asterisk local subscription storage
Fail condition	Subscriber information can not be stored in Asterisk local subscription storage
Actors	Subscriber
Triggers	Initiated by subscriber
External Resources	File system
Service	Asterisk system service
Remark	

Table 4.5: UC 1 - Client registration details

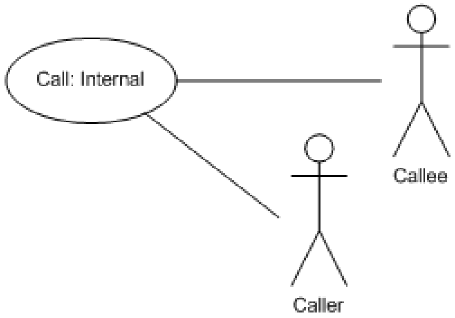

Figure 4.2: UC 2 - Call internal

UC 2 - Call internal (Figure 4.2)	
Goal	Establish a media session between to two local subscribers; controlled session tear-down
Precondition	• Asterisk service is running properly • Client devices are registered
Success condition	• Media session between two local subscribers established • Media bridge established • Controlled media session tear down
Fail condition	Media session could not be established
Actors	• Caller • Callee
Triggers	Initiated by caller
External Resources	File system
Service	Asterisk system service
Remark	

Table 4.6: UC 2 - Call internal details

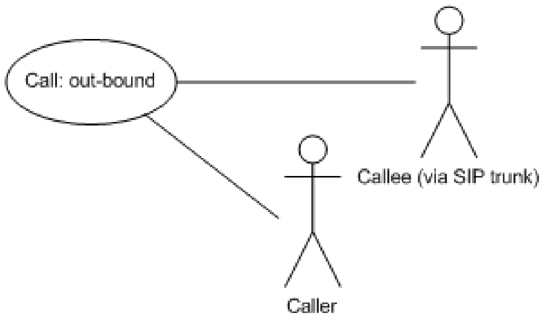

Figure 4.3: UC 3 - Call out-bound via SIP trunk

UC 3 - Call out-bound via SIP trunk (Figure 4.3)

Goal	Establish a media session between a local subscriber and a foreign system subscriber, initiated by the local subscriber; controlled session teardown
Precondition	• Asterisk service is running properly • Client device is registered • SIP trunk is available
Success condition	• Media session between the local subscriber and the foreign system subscriber is established • Media bridge established • Controlled media session tear down
Fail condition	Media session could not be established
Actors	• Caller • Callee (via SIP trunk)
Triggers	initiated by Caller
External Resources	• file system • SIP trunk
Service	• Asterisk system service • SIP trunk service
Remark	

Table 4.7: UC 3 - Call out-bound via SIP trunk details

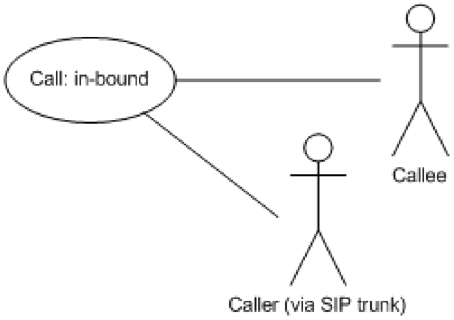

Figure 4.4: UC 4 - Call in-bound via SIP trunk

UC 4 - Call in-bound via SIP trunk (Figure 4.4)

Goal	Establish a media session between a local subscriber and a foreign system subscriber, initiated by the foreign subscriber; controlled session tear-down
Precondition	• Asterisk service is running properly • Client device is registered • SIP trunk is available
Success condition	• Media session between the local subscriber and the foreign system subscriber is established • Media bridge established • Controlled media session tear down
Fail condition	Media session could not be established
Actors	• Caller (via SIP trunk) • Callee
Triggers	Initiated by caller (via SIP trunk)
External Resources	• File system • SIP trunk
Service	• Asterisk system service • SIP trunk service
Remark	

Table 4.8: UC 4 - Call in-bound via SIP trunk details

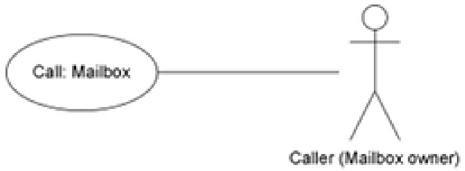

Figure 4.5: UC 5 - Call mailbox

UC 5 - Call mailbox (Figure 4.5)

Goal	Establish a media session between a local subscriber and Asterisk's voice mail module.
Precondition	• Asterisk service is running properly • Client device is registered
Success condition	• Media Session between the local subscriber and Asterisk voice mail system is established • Local subscriber is able to retrieve and administer the voice mail messages • Controlled media session tear down
Fail condition	• Media session could not be established • Local subscriber is not able to retrieve the voice mail messages
Actors	Caller (mailbox owner)
Triggers	Initiated by caller
External Resources	File system
Service	Asterisk system service
Remark	

Table 4.9: UC 5 - Call mailbox details

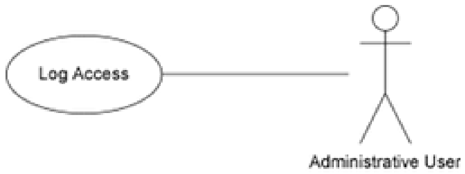

Figure 4.6: UC 6 - Log access

UC 6 - Log access (Figure 4.6)

Goal	Give an administrative user the possibility to access and search Asterisk and Linux system log files
Precondition	Linux OS has been started properly
Success condition	The administrative user is able to browse and search within log files
Fail condition	The log files can not be accessed or searched by the administrative user
Actors	Administrative user
Triggers	Initiated by the administrative user
External Resources	File system
Service	Linux OS
Remark	

Table 4.10: UC 6 - Log access details

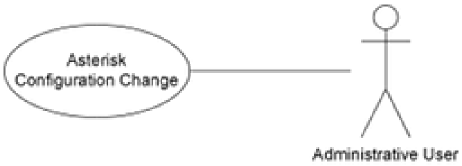

Figure 4.7: UC 7 - Asterisk configuration change

UC 7 - Asterisk configuration change (Figure 4.7)

Goal	Give an administrative user the possibility to access and modify the Asterisk configuration files
Precondition	Linux OS has been started properly
Success condition	The administrative user is able to do and apply the required configuration change
Fail condition	The configuration change can not be done or applied
Actors	Administrative user
Triggers	Initiated by the administrative user
External Resources	File system
Service	Linux OS
Remark	

Table 4.11: UC 7 - Asterisk configuration change details

5 Risk and Vulnerability Evaluation of Asterisk

This chapter evaluates and identifies security threats of the reference system described in Chapter 4 according to the model described in Section 3.4, steps II to VII.

5.1 Analyze the Technical Background of the UCs

This step of the security analysis evaluates the main possible points of contact which could be exploited to mount an attack and cause a security violation. This will be achieved through illustrating the exact flow of activities conducted by the system for each UC described in Section 4.2 where every activity is a potential point of contact to be utilized for a security attack.

For this evaluation, the according activities have been simulated on the reference system (as described in chapter 4). While running the simulation, the output files of Asterisk's debugging and Linux's logging mechanisms, both configured in verbose mode, have been recorded. The captured log files have been used to establish the following analysis.

- **Activity 1 - Authentication**

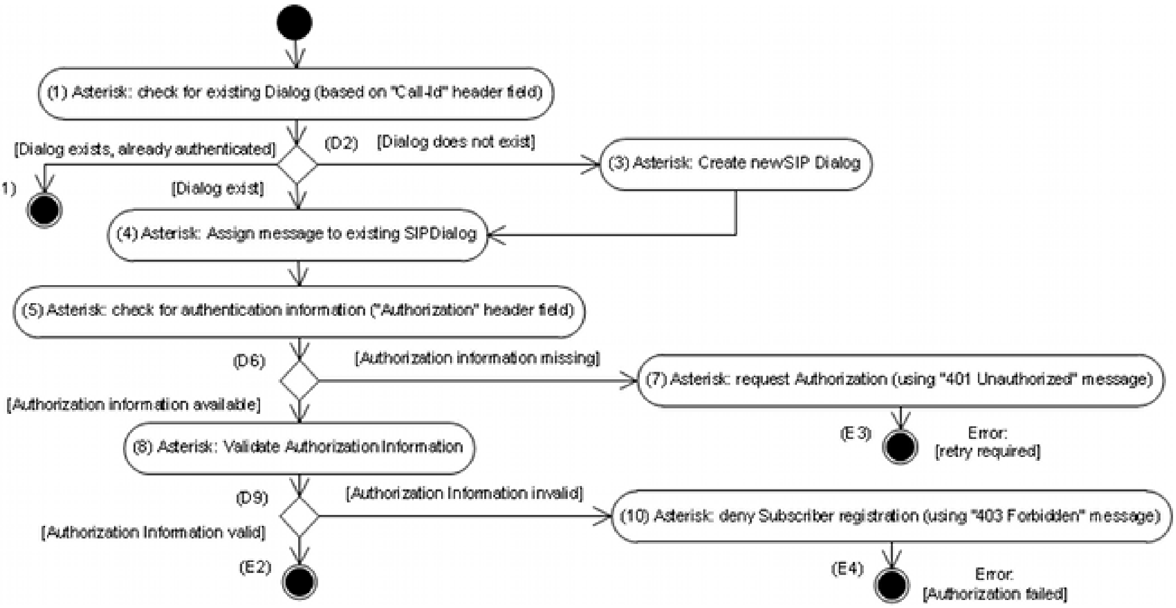
(1) Asterisk: check for existing Dialog (based on "Call-Id" header field)
[Dialog exists, already authenticated]
(D2)
[Dialog does not exist]
(3) Asterisk: Create newSIP Dialog
(E1)
[Dialog exist]
(4) Asterisk: Assign message to existing SIPDialog
(5) Asterisk: check for authentication information ("Authorization" header field)
(D6)
[Authorization information missing]
(7) Asterisk: request Authorization (using "401 Unauthorized" message)
[Authorization information available]
(8) Asterisk: Validate Authorization Information
(E3)
Error:
[retry required]
(D9)
[Authorization Information invalid]
(10) Asterisk: deny Subscriber registration (using "403 Forbidden" message)
[Authorization Information valid]
(E2)
(E4)
Error:
[Authorization failed]

Figure 5.1: Activity 1 - Authentication flow

Some UCs share the common activity **authentication** which is illustrated in Figure 5.1. Authentication is responsible for making sure a client is identified and Asterisk is able to decide whether access can be granted or not (authorization). Asterisk's default settings require authentication to happen at the beginning of every client device initiated SIP dialog using the Hash-based Message Authentication Code (HMAC) mechanism, which uses cryptographic hash value functions in combination with a shared secret key [33].

The activity **authentication** will be invoked by Asterisk every time a message containing SIP requests like *REGISTER* or *INVITE* is received. *Step 1* verifies whether the received message belongs to an existing SIP dialog which is identified using the *Call-ID* header field of the received SIP message. The following *Decision D2* determines the result of *step 1* and proceeds accordingly in three possible ways:

1. The message will be assigned to the current SIP dialog and *activity authentication* will be ended without error (*End E1*) if an appropriate SIP dialog already exists and authentication has succeeded earlier or Asterisk decides that no authentication is required for this SIP dialog. No authentication is required for SIP dialogs not initiated by a client, but initiated by Asterisk itself.
2. The message will be assigned to the current SIP dialog and *activity authentication* will continue (*step 4*) in case an appropriate SIP dialog exists but authentication is required and has not succeeded so far.
3. A new SIP dialog will be created, the message will be assigned to this new SIP dialog and *activity authentication* will continue (*step 3* followed by *step 4*) if no appropriate SIP dialog could be found.

In case the activity will be continued, *step 5* will verify if authorization details transported by header field *authorization* are included in the SIP request from the sender. At a sender's initial request, and after a new SIP dialog has been created using *step 3*, the client (sender) will not include any authorization information as the client does not have enough information to provide and the client does not know whether this service requires authentication. *Decision D6* evaluates the result of *step 5* and determines how to proceed:

1. The *activity authentication* is continued by proceeding to *step 8* in case authorization details have been provided.
2. A *401 UNAUTHORIZED* status reply message addressed to the creator of the request will be sent. This message contains a *WWW-Authenticate* header field providing information *digest algorithm* and *nonce* required by the client to compute the authorization information (*step 7*). Afterwards, the *activity authentication* will be ended using error *[retry required]* (*End E3*) which also requires Asterisk to stops all further processing of this request. Usually the client device renews its original request by adding the header field *authorization* causing Asterisk to re-start message handling and to re-start *activity authentication* as well. This will happen in case *step 5* has determined that authorization information are included within the request.

The client calculates the authorization information hash values using the HMAC algorithm. For this calculation the provided information in header field *WWW-Authenticate* and the logon details (user name and password) configured at the client device are used. The HMAC algorithm may utilize different cryptographic hash functions like Message-Digest Algorithm 5 (MD5) or Secure Hash Algorithm 1 (SHA-1). Header field *WWW-Authenticate* indicates which algorithm is going to be used.

During *step 8* Asterisk server carries out the same calculation and *Decision D9* determines if both hash values are identical:

1. The *activity authentication* will be ended without error (*End E2*), if both determined hash values are identical. This means that the client device and Asterisk used the same, and therefore valid, logon credentials.

2. The *activity authentication* will be ended with error *[Authorization failed]* (*End E4*) in case both calculated hash values differ. Before the activity is ended, a *403 FORBIDDEN* status reply message will be sent to the client, informing that authorization did not succeed and therefore, Asterisk will not continue handling the request (*step 10*).

This activity can be ended with multiple results. If it is ended without error (*Decision D2* option 1 and *Decision D9* option 1), the activity which invoked *activity authentication* will assume that the operation has been finished successfully and will continue normal operation. In case the *activity authentication* ended with an error, the invoking activity will stop immediately.

- **Activity 2 - Client registration**

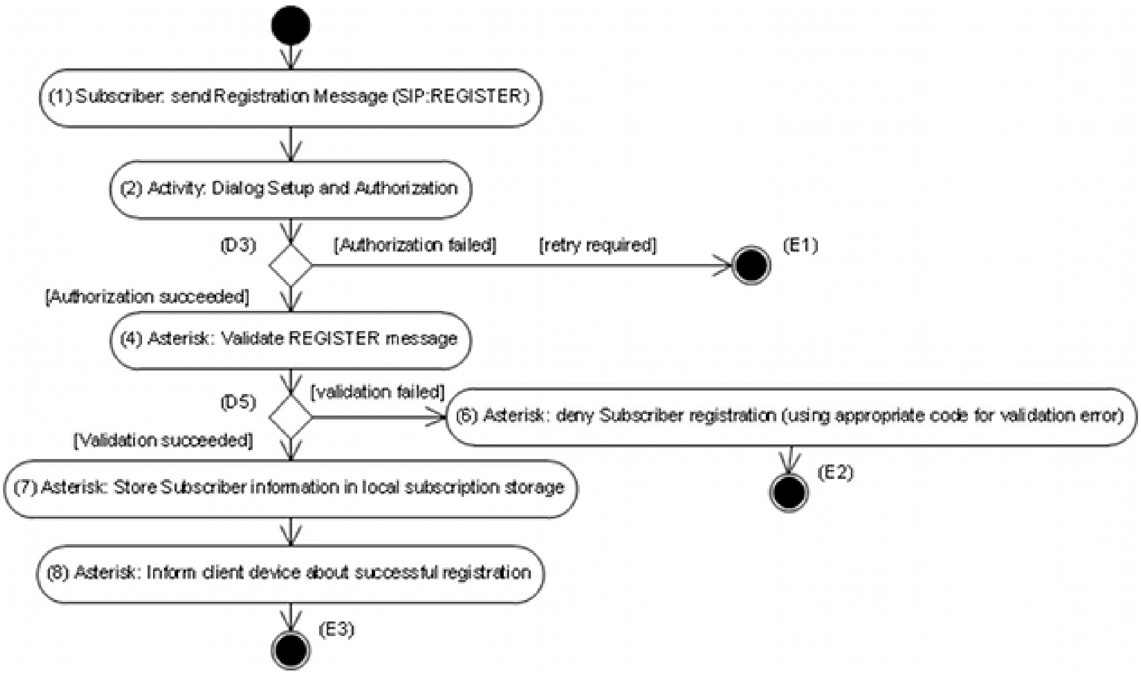
(1) Subscriber: send Registration Message (SIP:REGISTER)
(2) Activity: Dialog Setup and Authorization
(D3)
[Authorization failed]
[retry required]
(E1)
[Authorization succeeded]
(4) Asterisk: Validate REGISTER message
(D5)
[validation failed]
(6) Asterisk: deny Subscriber registration (using appropriate code for validation error)
(E2)
[Validation succeeded]
(7) Asterisk: Store Subscriber information in local subscription storage
(8) Asterisk: Inform client device about successful registration
(E3)

Figure 5.2: Activity 2 - Client registration flow

Figure 5.2 describes the flow of actions carried out by the Asterisk system while a client registers (UC 1). This activity is initiated by a client by sending a *SIP REGISTER* message to the Asterisk server (*step 1*). Once received, *step 2* invokes the *activity authentication* (Figure 5.1), with the result being evaluated by *Decision D3*:

1. The activity will be continued in case *activity authentication* ended without error.
2. The activity will be ended with error if proceeding *activity authentication* ended with an error, the corresponding SIP dialog will be scheduled for destruction (*End E1*).

Once the client is authorized, the validity of the received *SIP REGISTER* message will be verified by checking if all required header fields are available and if they hold valid information (*step 4*). The following (*Decision D5*) determines the validation's outcome:

1. Continue with *step 7* in case the validation succeeds.
2. The activity will end with error *[validation failed]* (*End E2*) in case the validation does not succeed. Further, the sender will be notified about the failure using a *400 BAD REQUEST* reply message (*step 6*).

During *step 7*, Asterisk saves the collected client information to its local subscription storage, elevating the client to the state of a **subscriber**. As next and last step, the subscriber will be notified about the succeeded registration by a *200 OK* reply message (*step 8* followed by *End E3*)

- **Activity 3 - Call internal**

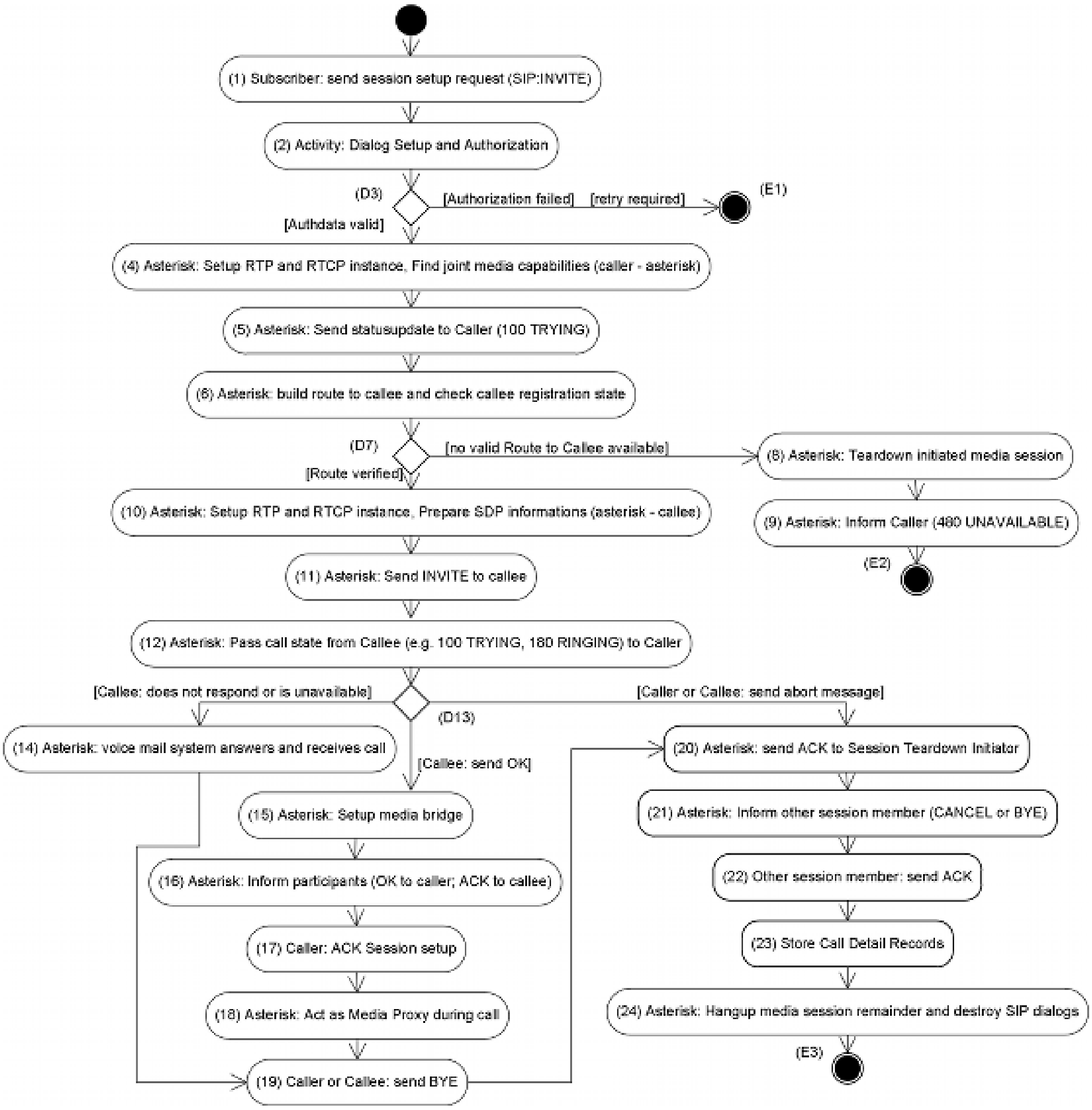
(1) Subscriber: send session setup request (SIP:INVITE)
(2) Activity: Dialog Setup and Authorization
(D3)
[Authorization failed] [retry required]
(E1)
[Authdata valid]
(4) Asterisk: Setup RTP and RTCP instance, Find joint media capabilities (caller - asterisk)
(5) Asterisk: Send statusupdate to Caller (100 TRYING)
(6) Asterisk: build route to callee and check callee registration state
(D7)
[no valid Route to Callee available]
(8) Asterisk: Teardown initiated media session
[Route verified]
(10) Asterisk: Setup RTP and RTCP instance, Prepare SDP informations (asterisk - callee)
(9) Asterisk: Inform Caller (480 UNAVAILABLE)
(E2)
(11) Asterisk: Send INVITE to callee
(12) Asterisk: Pass call state from Callee (e.g. 100 TRYING, 180 RINGING) to Caller
[Callee: does not respond or is unavailable]
[Caller or Callee: send abort message]
(D13)
(14) Asterisk: voice mail system answers and receives call
[Callee: send OK]
(20) Asterisk: send ACK to Session Teardown Initiator
(15) Asterisk: Setup media bridge
(21) Asterisk: Inform other session member (CANCEL or BYE)
(16) Asterisk: Inform participants (OK to caller; ACK to callee)
(22) Other session member: send ACK
(17) Caller: ACK Session setup
(23) Store Call Detail Records
(18) Asterisk: Act as Media Proxy during call
(24) Asterisk: Hangup media session remainder and destroy SIP dialogs
(E3)
(19) Caller or Callee: send BYE

Figure 5.3: Activity 3 - Call internal flow

Figure 5.3 describes the system's flow of actions while a call from one registered client to another is established and teared-down (UC 2) and further describes a scenario, where Asterisk acts as media gateway and implements media termination points (as described in section 2.3). Although Asterisk allows the usage of session setup without acting as media gateway, this configuration has been chosen because Asterisk has to perform additional tasks with more potential contact points compared to a scenario without Asterisk acting as media gateway.

This activity is invoked by a subscriber sending a *SIP INVITE* message to the Asterisk server (*step 1*). Once received, *step 2* starts the *activity authentication* (Figure 5.1), with the result being evaluated by *Decision D3*:

1. The activity will continue with *step 4* in case *activity authentication* ended without error.
2. The activity will be ended if *activity authentication* ended with error and the according SIP dialog will be scheduled for destruction (*End E1*).

Asterisk determines the joint media codec compatibilities between the caller and the Asterisk media gateway component, which is a codec both participants are capable of handling and in turn is going to be used for media communication between the caller and the server. Further, Asterisk creates a new internal instance able to handle the media stream (*step 4*). Once created, Asterisk informs the caller about the current call status using a *100 TRYING* status reply message (*step 5*). As next step (*step 6*), Asterisk tries to determine the path to the callee by performing an internal lookup. Locally registered subscribers will be found in Asterisk's local subscription storage. *Decision D7* is used to determine the following procedure depending on the lookup result:

1. The activity continues with *step 10* if Asterisk was able to find the callee in its local subscription storage.
2. Asterisk will destroy the instance to handle the media stream created during *step 4* and will notify the caller by sending a *480 UNAVAILABLE* status reply message (*step 8* followed by *step 9*) in case the callee lookup was not successful. Message handling will be finished and this activity is terminated (*End E2*).

Once callee's the registration has been found, Asterisk will setup another instance for media handling between the callee and the server and will prepare an appropriate SDP description to be included in the following *SIP INVITE* message sent to the callee. The media codec to be used is depending on the Asterisk configuration (*step 10* followed by *step 11*). Next, Asterisk will pass all call status messages like *100 TRYING* or *180 RINGING* received from the callee to the caller, having all participants (callee, server, caller) informed about the current call setup status (*step 12*). At the following *Decision D13*, Asterisk decides how to proceed based on the behavior and the messages received from the callee:

1. Asterisk proceeds with *step 15* if it receives a *SIP ACK* status message from the callee, indicating that the call has been accepted. At *step 15*, Asterisk is responsible for creating a media bridge between both formerly created media handling instances (during *step 4* and *step 10*). In case the used codecs differ, Asterisk will recode the media stream.

   This will be followed by an update of the call status (*step 16*) when Asterisk sends a *200 OK* status reply message to the callee and a *SIP ACK* message to the caller. Once the caller has acknowledged this *SIP ACK* message using a *200 OK* status reply message, the media session is established and is ready to exchange media data (*step 17* followed by *step 18*).

   The session will remain in this state until either the caller or the callee are requesting a termination of the media session using a *SIP BYE* message (*step 19*).
2. If the callee does not respond to the call request within a certain period of time, the voice mail system is invoked (*step 14*). It answers the call and allows the caller to leave a message. Asterisk records this message and stores it in the appropriate voice mailbox folder on its HDD. After leaving a message, the caller hangs up and session tear-down is initiated using a *BYE* message (*step 19*).
3. Asterisk continues with *step 20* in case the callee is not willing or not able to accept the call request by sending a message like *SIP CANCEL*.

Step 21 acknowledges the session termination using a *200 OK* status reply message. This message is either sent to the sender of the *SIP BYE* message or, in case no media session was established, to the caller after Asterisk received a cancellation message from the callee. This is followed by Asterisk sending an appropriate update message (either *SIP BYE* or *SIP CANCEL*) to the other participant of the media session and by receiving a *200 OK* status reply message from this participant (*step 21* followed by *step 22*). In case a call was established before, passing through *step 15* until *step 19*, the details including duration, caller and callee for the concerning media session will be store in the CDR database. Before ending the activity (*End E3*), all remaining Asterisk system resources and the appropriate SIP dialog is destroyed (*step 24*).

During this activity, messages may be lost or a subscriber may become unresponsive due to underlying network environment issues or various other reasons. In this case, the creation of a media session is not possible.

- **Activity 4 - Call out-bound via SIP trunk**

  Out-bound calls are calls from local registered subscribers with the intention to create a media session with subscribers on a foreign system as described in UC 4.

  The process of session setup is the same process utilized in **Activity 3 - Call internal** (Figure 5.3) with on distinctive difference that the callee is not a local registered subscriber to be found in the local subscription storage. As a foreign system subscriber is not directly registered on this Asterisk instance, and this instance does not have any information about the subscriber's state, the lookup of a path to the callee (*step 6*) must be extended. Therefore, not only the local subscription table is searched, also a SIP trunk configuration can indicate groups of subscribers which can be reached via SIP trunk.

  This requires Asterisk to send the session setup request (*SIP INVITE*) to the respective SIP trunk provider. The SIP trunk provider is responsible for delivering the request to the foreign system subscriber as well as for transferring all replies back to the Asterisk system via SIP trunk. Once a message has been passed to the SIP trunk provider, it can no longer be influenced or tracked by Asterisk.

- **Activity 5 - Call in-bound via SIP trunk**

  In-bound calls are calls where media sessions are established by foreign system subscribers with the intention to call a local subscriber.

  The session setup process as described in Figure 5.3 is applicable for in-bound calls as well. During this activity the authentication is carried out with the SIP trunk provider instead of the local subscriber. All communication with the caller is transmitted via the intermediate SIP trunk.

- **Activity 6 - Call mailbox**

  Figure 5.4 illustrates the flow of actions of a registered subscriber accessing the voice mailbox to retrieve personal messages received (UC 5). A directory number, similar to a regular subscriber's number, is assigned to the voice mail system. Once this number is called, Asterisk initiates the voice mail system which answers the call.

  This activity is initiated by subscribers through sending a *SIP INVITE* message, containing the request to set up a media connection with the directory number of the voice mail system (*step 1*) to the server. The following steps (*step 2 to step 5*) show an initial authentication, the setup of the RTP environment, and are described above in detail in **Activity 3 - Call internal**.

  While Asterisk processes *step 6*, it notices the called number's association with the voice mail system and invokes the internal function *VoiceMailMain*, which answers the call

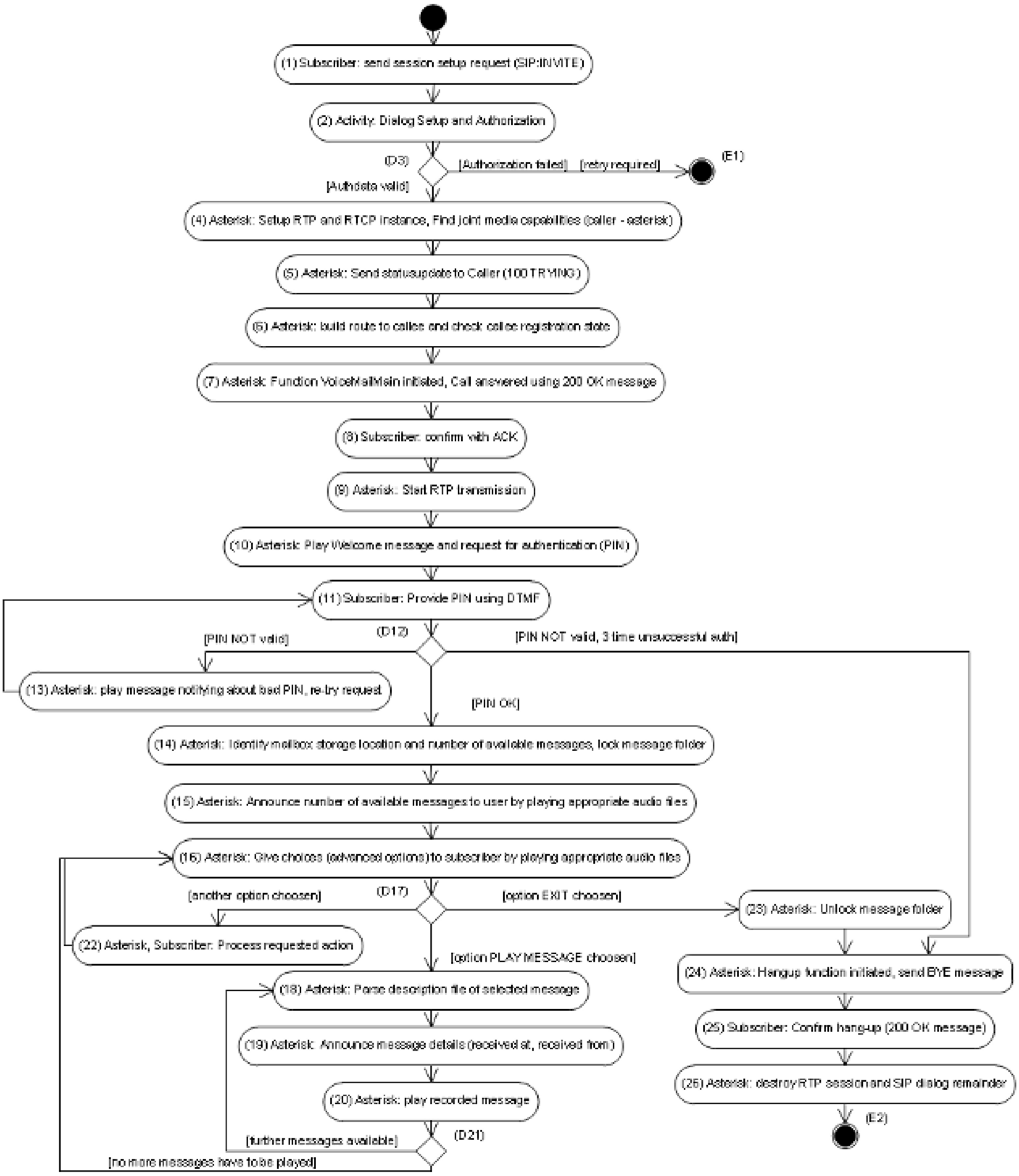
(1) Subscriber: send session setup request (SIP:INVITE)
(2) Activity: Dialog Setup and Authorization
(D3)
[Authorization failed] [retry required]
(E1)
[Authdata valid]
(4) Asterisk: Setup RTP and RTCP Instance, Find joint media capabilities (caller - asterisk)
(5) Asterisk: Send statusupdate to Caller (100 TRYING)
(6) Asterisk: build route to callee and check callee registration state
(7) Asterisk: Function VoiceMailMain initiated, Call answered using 200 OK message
(8) Subscriber: confirm with ACK
(9) Asterisk: Start RTP transmission
(10) Asterisk: Play Welcome message and request for authentication (PIN)
(11) Subscriber: Provide PIN using DTMF
(D12)
[PIN NOT valid]
[PIN NOT valid, 3 time unsuccessful auth]
(13) Asterisk: play message notifying about bad PIN, re-try request
[PIN OK]
(14) Asterisk: Identify mailbox storage location and number of available messages, lock message folder
(15) Asterisk: Announce number of available messages to user by playing appropriate audio files
(16) Asterisk: Give choices (advanced options) to subscriber by playing appropriate audio files
[another option choosen]
(D17)
[option EXIT choosen]
(23) Asterisk: Unlock message folder
(22) Asterisk, Subscriber: Process requested action
[option PLAY MESSAGE choosen]
(24) Asterisk: Hangup function initiated, send BYE message
(18) Asterisk: Parse description file of selected message
(25) Subscriber: Confirm hang-up (200 OK message)
(19) Asterisk: Announce message details (received at, received from)
(26) Asterisk: destroy RTP session and SIP dialog remainder
(20) Asterisk: play recorded message
(E2)
[further messages available]
(D21)
[no more messages have to be played]

Figure 5.4: Activity 6 - Call mailbox flow

by replying to the subscriber with a *200 OK* message (*step 7*). Once the subscriber has acknowledged the sessions setup, the voice mail system starts transmitting RTP data (*step 8 and step 9*). The RTP data Asterisk sends out is generated from audio files stored on the server's HDD. Asterisk's configuration determines the order of audio files to be played and allows to receive commands from the subscriber using in-session dialed DTMF digits transmitted by RTP [100]. This allows the subscriber to have interactive control of Asterisk's voice mail program.

The first audio file played is generally a welcome message and the request to authenticate by providing the numeric mailbox password (*step 10*). *step 11*, the subscriber providing the Personal Identification Number (PIN) using DTMF, is followed by *Decision D12*, which verifies if the provided PIN is correct and leads to following options:

1. In case the provided mailbox PIN is wrong, Asterisk continues with *step 13*, notifying the caller that the received PIN is incorrect and does not allow to access the voice mailbox. Furthermore, the system requests the caller to re-enter PIN and loops back to the beginning of *step 11*.
2. In case a wrong mailbox PIN is provided for a certain prefixed amount of times, Asterisk ends the voice mail program and initiates session tear-down at *step 24*.
3. Asterisk's voice mail program will continue its operation with *step 14* if the provided PIN is correct.

Once the subscriber's permission is verified, Asterisk identifies the storage location of the voice mailbox in question, locks it and determines the number of messages available (*step 14*). The locking mechanism identifies the mailbox as being currently in use but does not prevent from simultaneous usage. Then, Asterisk announces the number of available messages by playing the appropriate audio files (*step 15*) and gives options for actions to the user to choose from (*step 16*). *Decision D17* will determine which option is chosen by the subscriber and is in turn received by Asterisk:

1. In case the user chooses to listen to the messages, Asterisk parses the message's description file (*step 18*) to retrieve details like the time the message was left and the phone number of the message originator. Those details are announced to the user (*step 19*) and the recorded message is played (*step 20*). Once the message has been finished, *Decision D21* will decide on how to proceed:
   a. In case it was the last message which has to be played, Asterisk voice mail program will go back to the main menu and announce further potential activities for the subscriber (go to *step 16*).
   b. If there are still messages to be played, Asterisk will return to *step 18* and process the next message.
2. If another menu activity apart from playing messages or exit the voice mail system is chosen, Asterisk acts accordingly and executes the commands received from the subscriber. Possible commands are to delete respective messages or forwarding them to other subscribers. Not every command is explained in detail at this point, as they do not introduce any new way of server internal handling than already discussed and are not of crucial importance for the subsequent security analysis.
3. Asterisk releases the lock on the concerned mailbox folder (*step 23*), ends the voice mail program and initiates the session tear-down by sending a *BYE* message to the subscriber (*step 24*) if the subscriber chooses to exit the voice mail menu. Once the subscriber confirmed, Asterisk will destroy the RTP session and the SIP dialog remainder before processing the request is ended without an error (*step 25, step 26 and End E2*). In case the subscriber hangs up while interacting with the voice mail system, *step 26* is invoked automatically to allow Asterisk to make final clean-ups.

Asterisk implements additional menu options and accepts DTMF input while executing a function (at *step 20* or *step 22* for example). Such an input will cause Asterisk to interrupt its current action and to continue with the next step invoked by the input. To give a practical example, this allows to proceed with the next step in cause a long voice mail message is played.

- **Activity 7 - Log access**

  Figure 5.5 shows the flow of activities while an administrative user accesses the system's log files (UC 6). An administrative user is required to access log files for threat identification, system monitoring or for the reconstruction of unexpected system behavior which

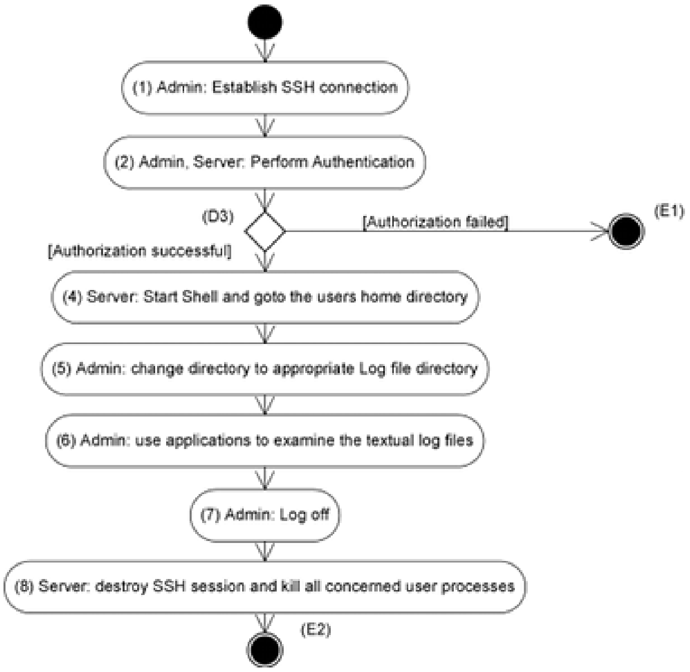
(1) Admin: Establish SSH connection
(2) Admin, Server: Perform Authentication
(D3)
[Authorization failed]
(E1)
[Authorization successful]
(4) Server: Start Shell and goto the users home directory
(5) Admin: change directory to appropriate Log file directory
(6) Admin: use applications to examine the textual log files
(7) Admin: Log off
(8) Server: destroy SSH session and kill all concerned user processes
(E2)

Figure 5.5: Activity 7 - Log access flow

has occured earlier. The administrative user initiates the connection to the server by starting a TCP session on the network port on which the server's SSH daemon listens (*step 1*). The SSH protocol offers various ways of authenticating incoming sessions and allows to identify the user in question (*step 2*) in an appropriate way. *Decision D3* determines if the authentication has succeeded properly and will end the process (*End E1*) in case it failed.

Once authentication was finished successfully and the user was identified by the server, the server will start a shell process and redirect to the user's home directory (*step 4*). From now on, the administrative user is able to work interactively on the system. During the next step, *step 5*, this user has to change to the directories where the log files are being stored. This will be achieved using the Linux command *cd*. By the use of Linux utilities like *cat*, *less* or *grep*, the administrative user is able to browse through the log entries (*step 6*). The user should not be able to modify any log files though.

The log files in question are usually located within the `/var/log` folder on the Linux file system, which is being used by the Linux kernel as well as other software for their logging entries. File `/var/log/messages` is the main logging target for the Linux kernel, while Asterisk uses folder `/var/log/asterisk` for its logging entries, unless otherwise configured. Also the CDRs are some sort of extended logging entries which can be accessed through the database's CLI commands.

Once finished, the user may log off from the system using the commands *logout* or *exit*, after which the server destroys the SSH session remainders and exits all processes concerned (*step 7 and step 8*).

- **Activity 8 - Asterisk configuration change**

  Figure 5.6 shows the flow of actions when UC 7, an Asterisk configuration change, is executed. Configuration changes are required in case the system behavior should be changed or unexpected behavior has been identified and needs to be fixed. This is done using a SSH session, whereas the first steps (*steps 1 until 4*) of this activity are similar to **Activity 7 - Log access** as described earlier in this section. *step 5* only differs in the destination folder the user is changing to. The destination folder where Asterisk's configuration files on the

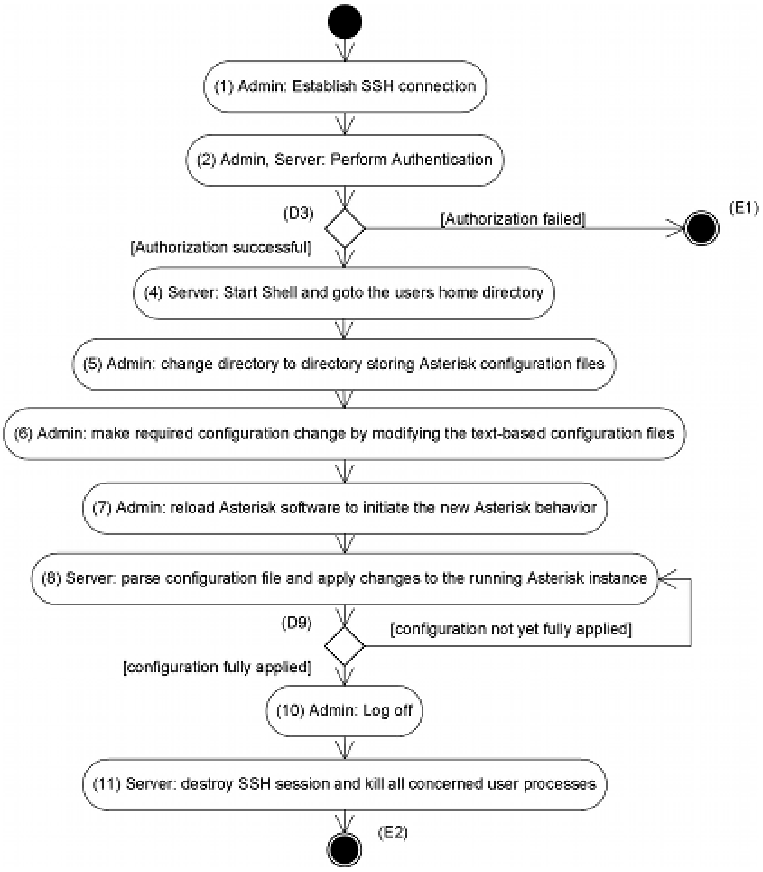
(1) Admin: Establish SSH connection
(2) Admin, Server: Perform Authentication
(D3)
[Authorization failed]
(E1)
[Authorization successful]
(4) Server: Start Shell and goto the users home directory
(5) Admin: change directory to directory storing Asterisk configuration files
(6) Admin: make required configuration change by modifying the text-based configuration files
(7) Admin: reload Asterisk software to initiate the new Asterisk behavior
(8) Server: parse configuration file and apply changes to the running Asterisk instance
(D9)
[configuration not yet fully applied]
[configuration fully applied]
(10) Admin: Log off
(11) Server: destroy SSH session and kill all concerned user processes
(E2)

Figure 5.6: Activity 8 - Asterisk configuration change flow

local file system are to be found is, in most cases, the `/etc` folder on the Linux file system. Asterisk, for example, stores its configuration files within folder `/etc/asterisk`.

By the use of utilities like *vi* or *nano*, the administrative user is now able to modify Asterisk's configuration according to the needs (*step 6*). Once finished, the changed configuration needs to be applied to the operating instance of Asterisk, being done using the Linux command *service asterisk reload* or using the Asterisk CLI command *reload* (*step 7*). Asterisk reads the first configuration file and applies it by adapting its behavior (*step 8*). This is followed by *Decision D9*, verifying that all configuration files have been applied required to run Asterisk. If the check reveals that this has not been achieved as of yet, *step 8* will be repeated with the next configuration file until all files are processed.

When finished, the administrative user logs off using the commands *logout* or *exit*. The server destroys the SSH session remainders and terminates all concerned processes (*step 10 and step 11*).

5.2 Identify Assets

Step III of the security evaluation (according to 3.4) is the determination of the system's assets. Assets identify important parts of the reference system (section 4), holding valuable data or providing crucial services. Assets are considered as targets for protection:

- The **Asterisk System Service** implements all voice services and all features of Asterisk Software. In case this service fails to deliver the expected functionality, the whole system may be affected and can not be used any longer.

  As providing voice services is the core task of the system, the correct functionality of Asterisk System Service has to be considered as an asset.

- **Configuration files** tell the Asterisk Software what to do and how to behave in predefined situations. These configuration files are not likely to be changed on a regular basis, but are required to run Asterisk as expected. Internal files allow, when altered, a change in Asterisk's system behavior. These internal files are considered as configuration files. This includes but is not limited to plain text configurations, audio prompts and executables.

  Any unauthorized change of configuration files may create unexpected system behavior and in effect will lead to lower system security. Configuration files also store sensitive data like user logon details including passwords. Hence, the Asterisk system configuration files have to be considered as an asset.

- **CDRs** contain and collect historical data of voice connections holding caller and callee details as well as time information. CDRs are mainly used to have accurate data for cost assignment and statistics, though it can be used to control user behavior as well.

  To be able to maintain the user's privacy, which is required by law [7], the CDR data needs to be considered as asset.

- **System/Asterisk Event Logs** are helpful in tracing system behavior for whatever reason. They capture detailed information about system events including the time of occurrence.

  In case of a system failure an engineer has to be able to track and trace the source of the failure to work up a detailed explanation and to be able to adjust security measures accordingly in order to avoid re-occurrences and to ensure the system's availability.

- **Environment**. The Asterisk based PBX system can only function as planned in case the underlying operating system and hardware is working properly. A fault within the hardware or operating system may limit the system's confidentiality, integrity and availability. Thus, the environment needs to be considered as an asset as well.

- **SIP trunk**s enable the local Asterisk system to establish communication with foreign systems (out-bound call) and as well allow foreign systems to establish calls with local system subscribers (in-bound call). trunks can be used to connect to PSTN providers.

  In case the SIP trunk is not working properly, the systems availability is effected considerably and this constitues why the SIP trunk is to be considered as an asset.

- The System's **Commercial Value** describe the total gain a company is able to achieve through making use of an Asterisk installation compared to not using a VoIP solution. Only in case the system behaves with integrity, the operator will be satisfied and financial gain can be maximized.

5.3 Determine Threats

This section identifies possible threats to the system according to step IV of the security evaluation approach (Section 3.4). A threat is considered to be a potential cause of an incident as described in detail in section 2.3.1.3.

Each single step carried out by Asterisk during the execution of an activity, which all have been described in detail in section 5.1, has effects on the system which may give a malicious user the opportunity to mount attacks. Considering all apparently concerned system units as well as all involved network messages for each previously identified action (presented in Figures 5.1 to 5.6), a list of possible hazards can be drawn up.

Software like Asterisk or the Linux kernel is built up of multiple thousands to millions of lines of code which are often implemented on a modular basis and are organized in libraries. A comprehensive list of threats can only be created by analyzing the source code of every involved function. Due to the extensive source code size and the caused complexity, such a detailed analysis has not been executed. Hence the threats assumed and the list presented in this section may not be exhaustive. As nowadays rapid advancement in IT infrastructure systems and software development as well as the enhancements of attack methods take place, this assumed list of threats shall be seen as a snapshot of today's situation only, which does not predict any possible future threats.

The *Threat ID* starts with key character *T* and is followed by a numbered section. The first part of the numbered section is the number of the activity with which the threat can be associated with whereas the second part is a continuous number. To be able to identify common threats which can not clearly be associated with one dedicated activity, the number 0 is used in the first part of the numbered section.

Following Tables 5.1 to 5.7 show the threats identified with their corresponding activity.

ID and **Name**	**Classification** and **Description**
T0-1 Eavesdropping	Information disclosure: An attacker retrieves common information by inspecting traffic on the network.
T0-2 Dialog acquisition	Elevation of privileges, DoS: By affecting the contents of the header field *call-id*, an attacker manages to take over and is able to continue communicating using an existing and already authenticated SIP dialog.
T0-3 Theft of data	Information disclosure: An attacker manages to gain access to the server and steals information.
T0-4 Malicious software	Reputation, DoS: An attacker manages to run introduced software or manages to call OS commands on the server which leads to unexpected system behavior.
T0-5 Killing of user threads	DoS: Threads handling SIP communication on the server are ended by an attacker. The affected communication channels become unresponsive and cause certain communications with clients to fail.
T0-6 Filling memory / disk space	DoS: System memory or disk space will be filled by an attacker. Log files, CDRs or voice mail message can cause such a shortage on the system and can prohibit others from being stored.
T0-7 Isolating the server	DoS: The attacker makes alterations to the network infrastructure with the goal of limiting the client's or server's availability. For example, the attacker may modify DNS records in a way so that the server machine is no long reachable on the network using DNS naming.
T0-8 Environmental availability	DoS: An attacker manages to take influence on the environment the system. This includes access to power supply or air-conditioning controls or, in case the server system is hosted in an virtual environment, access to the host system.
T0-9 Undesired communication	DoS: An attacker may send SPIT to annoy local system subscribers; Furthermore the system may be used by an attacker to originate SPIT messages.
T0-10 SIP dialog storage	DoS: An attacker may modify Asterisk's local storage for current SIP dialogs causing the request to fail.
T0-11 Asterisk executables	Tampering, DoS: An attacker may modify or exchange Asterisk internal executables creating an undesired system behavior or even limiting the system's availability.
T0-12 Network overuse	DoS: An attacker manages to utilize the network environment in a way that the VoIP system is not able to allocate enough bandwidth for uninterrupted and continuous operation.
T0-13 Message validity	DoS, Tampering: An attacker is able to tamper session setup and media stream messages and invalidates them by, for instance, removing single header fields.
T0-14 Log denial	DoS: The Asterisk system is denied storing log entries or CDRs due to malicious activities of an attacker.

Table 5.1: Common threats

ID and **Name**	**Classification** and **Description**
T1-1 Unauthorized registration	Spoofing: An attacker gets in possession of valid user registration details and uses them to register.
T1-2 Dictionary attack / brute force	DoS: An attacker is not in possession of valid user registration details but tries to find valid registration details using multiple attempts. Hence, the server is flooded with bogus registration requests.
T1-3 Loss of integrity	Tampering, DoS: An attacker uses unauthorized replay, deletion or insertion of messages to mount an attack.
T1-4 Data forgery	Tampering: An attacker manipulates messages to mount an attack.
T1-5 Man-in-the-Middle	Spoofing, DoS: An attacker pretends to be a client while facing the server and pretends to be a server while facing the client. This allows the attacker to gain unauthorized access causing a valid authentication on the real server's side while causing an error at the real client's side.

Table 5.2: Activity 1 - Authentication threats

ID and **Name**	**Classification** and **Description**
T2-1 Tamper registration	Elevation of Privilege, DoS: An attacker modifies an user request in a way so the media sessions are redirected to the attacker.
T2-2 Registration denial	DoS: A valid user request is modified by an attacker in a way so registration does not succeed. This causes the user not to be reachable by the Asterisk server (client denial).
T2-3 Registration DoS	DoS: An attacker sends an excessive amount of registration requests within a short period of time with the goal to cause the server to run out of resources (server denial).
T2-4 Multiple registrations	DoS: In case an attacker is in possession of valid user details these can be used to register multiple times on the server within a short period of time with the goal to limit the system's availability (server denial).
T2-5 Server impersonation	DoS, Information disclosure: The attacker pretends to be the server and manages to bring the client to register on the malicious server instead of the real one.
T2-6 Subscriber data manipulation	Tampering, DoS: An attacker is able to modify the contents of Asterisk's local subscriber storage, which may cause a registered subscriber to become unreachable.
T2-7 Registration removal	DoS, Tampering: An attacker manages to remove details about current registered users from the server.

Table 5.3: Activity 2 - Client registration threats

ID and **Name**	**Classification** and **Description**
T3-1 Invite denial	DoS: An attacker manages to invalidate a valid *INVITE* message in a way the session between caller and callee can not be established properly.
T3-2 Tampered invite	Elevation of privileges, DoS: A valid *INVITE* message is tampered so the attacker gains control of the communication instead of the initiating client user.
T3-3 Invite DoS	DoS: An attacker causes to send an excessive amount of valid *INVITE* messages within a short period of time while using valid authentication credentials. These messages can be directed to the server or to a client causing service denial on the affected side.
T3-4 Invalid requests	DoS: An attacker sends an excessive amount of malformed *INVITE* messages within a short period of time while using valid authentication credentials causing high CPU load on the server.
T3-5 Tamper media stream	Tampering: The attacker alters the media stream while the session is established.
T3-6 Tamper control messages	Tampering, DoS: An attacker manages to alter control and setup messages of the media stream. This can be done by introducing fake *BYE* or *CANCEL* messages.
T3-7 Decline codecs	DoS: An attacker manages to disable the server's ability to use certain media codecs.
T3-8 Routing disability	DoS: An attacker causes Asterisk not to be able to make routing decisions due to modified routing entries.
T3-9 Database system manipulation	DoS: In case an attacker manages to get access and manipulate the CDR database, Asterisk can be denied of storing CDR details properly.
T3-10 Database dump	Information disclosure: An attacker may retrieve CDR data from the database.
T3-11 Data manipulation	Tampering: An attacker manages to delete or modify stored records from the CDR database.
T3-12 Data insertion	Tampering: An attacker may insert bogus records into the CDR database.
T3-13 Server impersonation	DoS, Information disclosure: The attacker pretends to be the server and manage to have the client initiate the sessions via the malicious server instead of the real one.

Table 5.4: Activity 3 - Call internal threats

ID and **Name**	**Classification** and **Description**
T4/5-1 SIP trunk registration invalid	DoS: The SIP trunk provider can not authenticate due to invalid registration details.
T4/5-2 SIP trunk availability	Tampering, DoS: An attacker uses network environment vulnerabilities to limit the availability of the SIP trunk.
T4/5-3 SIP trunk replacement	Spoofing: An attacker simulates a SIP trunk and redirects all traffic via a malicious SIP trunk (MitM).
T4/5-4 SIP trunk provider availability	DoS: The SIP trunk provider may not be able to handle SIP requests properly due to several reasons including hardware failure or attacks against the provider's core network environment.
T4/5-5 SIP trunk eavesdropping	Information disclosure, Tampering: Media transfered via the SIP trunk is inspected, recorded or modified on the providers side.
All threats identified for *Activity 3* (Table 5.4) can be applied to this activity as well.	

Table 5.5: Activity 4 and 5 - Call outbound and Call inbound via SIP trunk threats

ID and **Name**	**Classification** and **Description**
T6-1 Modify announcement files	Tampering, DoS: An attacker manages to modify, rename or delete audio files used for automatic announcements (as used by the voice mailbox system, for example).
T6-2 Unauthorized mailbox modification	Tampering, DoS: An attacker manages to modify, rename or delete the recorded messages on file system level.
T6-3 Expose mailbox credentials	Information disclosure: An attacker is able to retrieve a subscriber's mailbox password which allows to retrieve mailbox messages repeatedly.
T6-4 Unauthorized message retrieval	Information disclosure: An attacker is able to retrieve mailbox messages from a subscriber but does not have permanent access to mailboxes.
T6-5 Fill disk space	DoS: The storage location for mailbox messages is filled up so the server runs out of available memory.
T6-6 DTMF signals tampered	Tampering, DoS: An attacker is able to take influence on the DTMF signals transmitted either as part of the RTP stream or as SIP messages.
All threats identified for *Activity 3* (Table 5.4) can be applied to this activity as well.	

Table 5.6: Activity 6 - Call mailbox threats

ID and **Name**	**Classification** and **Description**
T7/8-1 Exposed SSH logon credentials	Information disclosure: An attacker retrieves the administrative logon credentials to access a Linux shell on the server.
T7/8-2 SSH DoS	DoS: An attacker is able to utilize server resources for example by sending a large amount of simultaneous SSH session setup requests (TCP SYN flood).
T7/8-3 Unauthorized file access	Information disclosure: An attacker is able to view files he is not supposed to access. This includes configuration files holding user information like registration passwords or mailbox PINs.
T7/8-4 Unauthorized file modification	Tampering, DoS: An attacker modifies or deletes files on the system so Asterisk's ability to start properly is blocked or disallows Asterisk to apply configuration changes.
T7/8-5 Unauthorized file execution	Tampering: An attacker uses applications and programs which are installed on the server or are part of Asterisk's software.
T7/8-6 Uneducated changes	DoS, Information disclosure: Due to the lack of knowledge of users, changes to the Asterisk configuration can result in unexpected system behavior.

Table 5.7: Activity 7 and 8 - Log access and Asterisk configuration change threats

5.4 Determine Vulnerabilities

This subsection represents step V of the security analysis (as described in section 3.4) and aims at determining vulnerabilities of the system. Based on the security analysis proceeding steps *Analyze the Technical Background of the UCs (Section 5.1)* and *Determine Threats (Section 5.3)*, potential vulnerabilities will be evaluated.

The *Vulnerability ID* introduced in this section will start with the key character *V* and is completed by adding a sequential number.

5.4.1 Host / Environmental Based Vulnerabilities

V1 **Physical Environment**:

Every system's security is only as good as the security of its underlying environment, as *if a malicious person can gain physical access to your facility or equipment, they can do just anything they want, from destruction to disclosure and alteration* [110]. Even though the Asterisk system may be hosted using virtualization techniques, physical hardware is required non the less. This physical hardware is only able to function as expected if general requirements like an accurate temperature control, the use of uninterruptible power supplies or utilization of an appropriate physical location, are fulfilled.

The Asterisk system might also run as part of a private or public cloud service [5], which has additional implications on security like guest-to-cloud threats [128].

An attacker may be able to take advantages of flaws in the governing IT Policies (see Section 2.3.1.2) or of misimplementation thereof.

V2 **Host OS Misconfiguration**:

Flaws in the configuration of the host OS may be used by an attacker to gain access to the system. As Linux consists of multiple independent software packages, with each of them having different configuration options, the list of possible errors in the overall Linux system configuration is long (see section 2.3.2 for an introduction to Linux). Mendes et al. [72] state that *The operating system running on system hosting the web server is a fundamental component when considering security of the web server...*. This can not only be applied to web servers but also to servers offering services on the web like VoIP systems. This is why any weakness of the host system may enable malicious users to mount attacks.

The possibilities for an attacker range from simple read-only access to properties of certain applications (software packages) to full shell-based system access.

V3 **Unskilled Administrative Users**:

Kaye [60] states that *human errors or unskilled staff that lead to misconfigurations will cause a significant amount of outages*. Therefore, the administrative user is required to have extensive knowledge of the Asterisk system and its configuration files. The user has to be aware of the possible impact on the system's behavior when changing a configuration parameter.

A lack of knowledge may have only little impact on the system's behavior, but may also result in devastating down-times or other system behavior which limits the expected operation of the system.

V4 **Errors in Software**:

Although quality aspects in software development standards (as introduced in section 3.3) are of high importance nowadays, each software may have undiscovered flaws and bugs.

This also includes Asterisk as well as all other software operating on the system, including the Linux kernel. As long as those flaws and bugs are undiscovered, they do not represent an immediate danger, but once a flaw is discovered, attacks taking advantage of this newly discovered bug may spread very quickly depending on the severity of the damage it may cause and depending on the discoverer's intentions.

As early as in 1982, Lipow [65] indicated that every software has faults by proving that the number of faults per line of code increases with the overall number of lines of code of the program.

As security flaws may be discovered for every software used to operate the Asterisk based VoIP system, the boundaries of what an attacker is able to do are difficult to define. An malicious action may have only very little affect on the system or may result in serious danger up to completely disabling the Asterisk operation.

V5 **Dependency on the Security of Other Network Protocols**:

The operation of a reliable VoIP installation requires the correct and reliable operation of certain network protocols like DNS, Dynamic Host Configuration Protocol (DHCP) or ARP. These protocols build the very core of each network environment [25] as they are responsible for addressing issues and help to identify and find hosts on a network. In case an attacker is able to limit the functionality of any of these protocols, a non-malicious subscriber may not be able to communicate with the Asterisk server properly (DoS: see Section 2.4.1.2).

Interfering with these protocols may also be used to cause the client's request to be redirected to a malicious server. This might be achieved by using ARP spoofing [82] (section 2.4.1.1).

V6 **Limited Resources**:

Although memory resources are cheap nowadays compared to earlier years, the amount of available system memory can not be unlimited. Comprehending of RAM memory used for short term storage during software operation and of permanent memory in form of permanent storage devices like HDDs, memory is subject to limitation. Hence, it is possible that it will be exhausted at a certain point of time. Also other resources are subject to limitation. A network connection does have a maximum amount of data which can be transfered per second, a CPU is limited by the number of possible operations per second, to name only a few.

Not only do limits in hardware apply, limits may also be introduced by software. For example, the Linux kernel associates a set of *resource limits* to each process which specifies the amount of system resources the process can use [19].

A system misconfiguration, the misestimation of required resources during system sizing, a lack of software memory or an attack against these weaknesses will result in the system running out of resources. In turn the system's inability to behave as expected causes DoS (see section 2.4.1.2).

V7 **Use of Insecure Transport Media**:

VoIP systems and all its adjunctive devices share the network environment with other devices and also share the same core protocols (like IP) with other applications. Due to this flexibility of the network infrastructure, an attacker can easily capture or even modify traffic on the network. Therefore, all communication transmitted on a network infrastructure between the Asterisk server, its connected client devices and other devices like the termination point of the SIP trunk may be subject to eavesdropping, modification or any other kind of disturbance by others on the network [25]. For example, the insertion of spoofed *UNREGISTER* messages into the network may prohibit a valid user from receiving calls [22], which is similar to the introduction of *CANCEL* messages described in section 2.4.2.1.

An attacker can take advantage of this vulnerability when the signaling messages (using SIP) and media communication (using RTP) are unencrypted and allow media sessions to be recorded or connection profiles of system subscribers to be generated. This interferes with the system's confidentiality.

V8 **Database Misconfiguration**:

The reference system (see chapter 4) uses a database to store all CDRs generated. For each SIP session requested, one data record will be generated. Asterisk's configuration states which database needs to be used and where the records are required to be stored. Similar to Asterisk's configuration files, the database engine requires a configuration as well. Amongst others, this regulates access to the data or storage properties. Neto et al. introduced a measure for evaluating the untrustworthiness of Database Management System (DBMS) configurations which is defined as *a measure of how one should distrust a given configuration to be able to prevent the manifestation of most common security threats as real attacks* [76]. This implies that every database configuration must be considered to have configuration flaws.

In case an attacker is able to gain access to the database engine, data records may be retrieved, modified or deleted. An unauthorized modification of the database scheme can even disable Asterisk to store data as required.

5.4.2 Asterisk Based Vulnerabilities

V9 **Asterisk Misconfiguration**:

The behavior of the Asterisk system is controlled by its configuration files, instructing Asterisk how to handle predefined situations (see section 2.2). These files are also used for documentation purposes of the system's behavior and can be used by system engineers to trace the actions of Asterisk. In case of a faulty configuration the access of non-malicious users can be wrongly denied, attackers may be able the gain access to the system, requests may be handled in a not intended manner or Asterisk migth even be hindered to initiate itself correctly any longer.

It mathematician John Conway's Game of Life, an initial configuration can result in a considerable population growth period before coming to an end or can result in an apparent growth without limits, at which there is no possibility to predict the outcome in the first place. Additionally, there is no simple way to prove that the population can grow indefinitely [45]. This model can be adopted to software configurations as well: Configuration of software can result in behavior which allows attackers to mount an attack, which is somehow comparable to the Game's coming to an end result. Also, it can cause software to work as expected, which can be compared to the apparent growth without limits result. This analogy shows that the configuration might cause the software to work as expected but there is no simple way to have full proof of this and hence, the decision if the configuration has flaws can not be made easily.

The misconfiguration of Asterisk may not only open doors for an attacker to weaken the system, but also may cause a good-natured subscriber, without any bad intentions, to cause harm to the system. For example, a faulty session routing configuration may result in charges by telephony providers which could have been avoided.

As another example, an Asterisk misconfiguration could allow access to the Asterisk CLI from a remote host, which the attacker is able to access. Afterwards, the CLI could be used by the attacker to execute OS commands.

V10 **Unauthorized use of Server Resources**:

Each communication between clients and the Asterisk server is unauthorized at the beginning. Although every SIP dialog starts with an authentication mechanism, the first message from the client is unauthorized and may be created by a good-natured or a malicious client. In both cases, the initial request has to be processed by Asterisk's which causes CPU load on the server, although the created load is not very high. The system resources are used while handling the request and are released a certain time after the processing has been finished.

Hence, unauthorized clients or attackers can create CPU load on Asterisk's server system by sending an excessive amount of requests to the server during a short period of time. This may cause the limitation of available system resources on the server and may effect the server so it is not able to serve non-malicious requests any longer and is referred to as DoS (as introduced in section 2.4.1.2).

V11 **Authentication Credentials**:

Asterisk server will only accept requests from clients in case the client is able to provide authentication details. For SIP requests these credentials consist of two alpha-numeric strings of random length named *user name* and *password*. While the user name is transmitted via the network as plain text, the password will be calculated as cryptographic hash values which are transmitted via the network instead of the plain password string itself. Authentication credentials of the same kind can be used to access the server system using SSH for administrative purposes.

Furthermore, a subscriber is required to provide a password during every logon to their personal voice mailbox, which can differ from the SIP logon password. It is made up of numbers only as the subscriber needs to enter the password using the phone's number pad, is transmitted as part of the RTP stream and is referred to as PIN.

Clair et al. [31] analyzed the required demands for optimized passwords today and in the future. The chosen strings as well as their lengths have an influence on their resistance against attacks, whereas strings like *password*, *p@ssword* or *1234* are easily cracked using a dictionary attack and are examples for very weak passwords. Burnett [23] shows that user passwords can be recovered quite easily. Passwords are, according to Anderson [4], *the (often shaky) foundation on which much of information security is built* and presents concerned challenges like *difficulties with remembering the password* or *naive password choices*.

There are multiple possibilities for an attacker to retrieve those authentication strings, including a cryptographic attack on the calculated hash values or by acquiring access to the client's system. The use of this authentication method is considered as vulnerable because if exploited, an attacker is able to gain full system access identical to a regular subscriber. It is an example of *Identity Theft* (see Section 2.4.2.3).

V12 **Registration Hijacking**:

Once a client has been registered on the Asterisk server using a *SIP REGISTER* request, the Asterisk server will assign a pre-defined expiration time and store the client's details until the expiration point is reached unless the subscriber re-registers before expiration. Not only the server, but also the subscriber keeps track of the expiration period and the subscriber will renew its subscription in time. During the time a client is properly registered, the Asterisk server trusts the device and enables communication. This makes it possible for an attacker to perform *Registration Hijacking* [104].

Although the setup of new media sessions initiated by that very subscriber requires authentication, incoming calls will be routed there without further evaluation. If an attacker is able to get in control of the client device, the Asterisk server will trust the attacker as well. In case the attacker is also able to retrieve the registration details during registration hijacking, the system can be used in the same way a non-malicious subscriber is able to. Otherwise, the attacker may only receive calls until the session expires. The non-malicious subscriber, whose registration has been stolen, is no longer reachable by Asterisk for the time being due to the caused DoS (see Section 2.4.2.1).

V13 **Dependency on the Security of the External SIP Trunk Provider**:

Asterisk systems use SIP trunks to enable communication with foreign telephony systems, operated by multiple other companies. Furthermore, the SIP trunks are operated by companies (usually telephony providers) which do not allow the customer to take any influence on the SIP trunk. This causes an unavoidable dependency on the telephony provider when using SIP trunks.

There is no possibility for the local Asterisk server to verify the payload data received. For example, the originating *Caller-ID* could have been altered and transmitted over the telephony provider's network [10].

Flaws, which Asterisk server and a system designer are not able to verify, may exist in the provider's SIP trunk configuration, which can be exploited to mount an attack. The attacker may use this telephony connection while causing telephone charges or may disable the Asterisk server to use the SIP trunk properly.

Although all possible security measure on Asterisk and on the local environment might have been implemented, the SIP trunk provider could, for example, conduct eavesdropping (see section 2.4.1.3).

V14 **DoS by Unexpected User Load**:

Every message received by the Asterisk server, even if the SIP dialog was properly authenticated before, needs to be processed on the server requiring the usage of server resources. As these resources are limited, a high number of non-malicious, simultaneous requests could cause the server to run out of available resources while causing DoS (see section 2.4.1.2).

For this vulnerability to be exploited, no actions by an attacker are required. It can be caused by either a misestimation of required system resources while designing the system, or by the failure of system resources caused through hardware faults, for example.

V15 **Unwanted Communication**:

One goal of VoIP systems like Asterisk is to enable unlimited and unified communication features for its subscribers, which entails a lot of benefits for companies and organizations which make use of this technology. On the contrary, companies may use this technology with negative effects for others. For example, advertising companies may automatically initiate calls to foreign system subscribers inviting them to take part in sales promotions in which the callee is usually not interested. Such kind of communication is referred to as SPIT (as introduced in section 2.4.2.4).

SPIT messages will not generate any harm to the available features of the Asterisk system, but will annoy subscribers and may create frustration. In contrast to SPAM in eMail systems, detecting SPIT is more complicated because the content can not be known when a new request for establishing a media session is received. Therefore, Schlegel et al. [98] presents a multi-staged SPIT prevention system.

5.5 Effectiveness Matrix: Threats – Vulnerabilities

This section establishes the mapping between the previously identified threats (Section 5.3) and previously identified vulnerabilities (Section 5.4). The columns show the list of vulnerabilities whereas the rows list the identified threats. An *X* indicates that the according vulnerability can be exploited by the respective threat.

	Vulnerability														
Threat	V1 Physical Environment	V2 Host OS Misconfiguration	V3 Unskilled Administrative Users	V4 Errors in Software	V5 Dependency on the Security of Other Network Protocols	V6 Limited Resources	V7 Use of Insecure Transport Media	V8 Database Misconfiguration	V9 Asterisk Misconfiguration	V10 Unauthorized use of Server Resources	V11 Authentication Credentials	V12 Registration Hijacking	V13 Dependency on the Security of the External SIP Trunk Provider	V14 DoS by Unexpected User Load	V15 Unwanted Communication
T0-1 Eavesdropping							X								
T0-2 Dialog acquisition					X		X								
T0-3 Theft of data		X					X	X	X						
T0-4 Malicious software		X		X					X						
T0-5 Killing of user threats		X													
T0-6 Filling memory / disk space						X									
T0-7 Isolating the server					X										
T0-8 Environmental availability	X														
T0-9 Undesired communications															X
T0-10 SIP dialog storage		X													
T0-11 Asterisk executables		X													
T0-12 Network overuse						X									
T0-13 Message validity							X								
T0-14 Log denial						X									
T1-1 Unauthorized registration											X				
T1-2 Dictionary attack										X					
T1-3 Loss of integrity							X								
T1-4 Data forgery							X								
T1-5 Man-in-the-Middle					X		X								
T2-1 Tamper registration					X							X			
T2-2 Registration denial					X							X			
T2-3 Registration DoS					X					X					
T2-4 Multiple registrations										X				X	
T2-5 Server impersonation					X										
T2-6 Subscriber data manipulation		X		X											
T2-7 Registration removal		X		X											
T3-1 Invite denial					X										
T3-2 Tampered invite					X		X								
T3-3 Invite DoS										X					
T3-4 Invalid requests										X				X	

	Vulnerability														
Threat	V1 Physical Environment	V2 Host OS Misconfiguration	V3 Unskilled Administrative Users	V4 Errors in Software	V5 Dependency on the Security of Other Network Protocols	V6 Limited Resources	V7 Use of Insecure Transport Media	V8 Database Misconfiguration	V9 Asterisk Misconfiguration	V10 Unauthorized use of Server Resources	V11 Authentication Credentials	V12 Registration Hijacking	V13 Dependency on the Security of the External SIP Trunk Provider	V14 DoS by Unexpected User Load	V15 Unwanted Communication
T3-5 Tamper media stream							X								
T3-6 Tamper control messages							X								
T3-7 Decline codecs		X													
T3-8 Routing disabilities		X							X						
T3-9 Database system manipulation		X													
T3-10 Database dump		X													
T3-11 Data manipulation		X													
T3-12 Data insertion		X													
T3-13 Server impersonation					X										
T4/5-1 Trunk registration invalid							X								
T4/5-2 Trunk availability					X		X								
T4/5-3 Trunk replacement					X										
T4/5-4 Trunk provider availability													X		
T4/5-5 Trunk eavesdropping													X		
T6-1 Modify announcement files		X													
T6-2 Unauth. mailbox modification		X													
T6-3 Expose mailbox credentials				X			X		X						
T6-4 Unauth. message retrieval		X		X			X		X						
T6-5 Fill disk space						X									
T6-6 DTMF signals tampered							X								
T7/8-1 Exposed SSH logon credentials						X									
T7/8-2 SSH DoS		X													
T7/8-3 Unauth. file access		X													
T7/8-4 Unauth. file modification		X													
T7/8-5 Unauth. file execution		X													
T7/8-6 Uneducated changes			X												

Table 5.8: Threat – Vulnerability Matrix

5.6 Asset Mapping

Step VI of the security analysis (section 3.4) will assess the importance of the assets identified in step III (section 5.2) which is a precondition for the risk analysis conducted in the following step VII.

The importance of an asset is described using the levels *High* or *Medium*. Assets which would have to be rated with an importance lower than *Medium* are omitted in this section as they do not have any remarkable influence on the risks or mitigation measures presented in later sections.

Asset	Importance	Justification
Asterisk System Service	**High**	The Asterisk System Service is classified as of high importance as it is responsible for providing the features of Asterisk's software to clients. In case this service does not follow its designed behavior, a system's availability is not given and the system needs to be considered as not operational.
Configuration files	**High**	Without any or with faulty configuration files, Asterisk may not be able to initialize its functions in an expected behavior. Although Asterisk may be able to continue its expected operation in case the configuration files change while enabled, the configuration files are considered as an asset of high importance. They should represent the expected behavior of Asterisk as well as enabling Asterisk to startup in a correct manner at any time.
CDRs	**High**	CDRs are a necessity for debugging and cost allocation purposes, but they do not prevent Asterisk from providing its core functionalities like media session setup. Nevertheless, the CDR data needs to be protected due to its sensitive content and must be accurate as this data has to prove that media sessions took place, which is essential in case charges are allocated to customers.
Event Logs	**Medium**	Event logs are of medium importance as they do not prevent Asterisk from acting as expected in terms of core functionalities. Event logs are used for tracing errors while Asterisk is in operation, but do not hold highly sensitive data.
Environment	**High**	The importance of the system's environment has been classified as high because it is a necessity for an expected behavior and operation of Asterisk's system service. Asterisk's system service may only work as reliable as the environment it is implemented in.
SIP trunk	**High**	Without an operational SIP trunk, Asterisk is still able to serve all local subscribers but is without the ability to establish media sessions with a foreign system subscriber. As communicating with foreign system subscribers is essential, the asset SIP trunk is rated as highly important.
Commercial Value	**High**	Only if the system is able to generate an system-related financial benefit, the owner and operator will be satisfied. In case the system causes high unpredictable costs, the company may be damaged or harmed severely. Therefore, this asset has been assigned the highest importance rating..

Table 5.9: Asset Importance

5.7 Risk Management

The following section, dealing with step VII of the security analysis approach (section 3.4) evaluates the risk the system is exposed to. Every former identified vulnerability (section 5.4) will have assets assigned (identified and rated in sections 5.2 and 5.6) which are possibly affected in case the respective vulnerability gets exploited.

The risk will be estimated on the basis of each former identified vulnerability. Therefore, the numbering of the risks presented in this section is directly connected to the vulnerabilities identified in section 5.4. This means that the risk of vulnerability V1 is re-presented as R1, the risk of vulnerability V2 is presented as R2 and so on. The importance rating attributed to each asset is quoted in brackets.

Although many security metrics have been described so far [58], no consensual security metrics have been proposed as of yet [57]. Estimated values for the *probability of occurrence* and the *expected financial damage* are used to estimate the overall risk of the vulnerability's exploitation, whereas the *expected financial damage* is approximated by the importance of the involved asset (see Section 5.6) and the expected financial damage if exploited.

As results presented in former sections (sections 5.3 and 5.4), do not claim to be entirely comprehensive, also items considered in the following list do not claim completeness either.

R1 Risk of Insufficient Security on the **Physical Environment**:

Insufficient measures on environmental security can have an impact on **every asset** as they may lead to the physical server's disability to execute any more commands with the possible result of an unresponsive Asterisk system.

Although data center downtime can cause business losses over millions of dollars per hour [125] and the outcome of an exploitation of such a vulnerability could cause a total system outage, accomplishing such an attack still requires big efforts, like gaining physical access from the attacker. The risk is evaluated as **medium risk**.

R2 Risk of Attacks Mounted on a Faulty **Host OS Configuration**:

A faulty host OS configuration can allow attackers to gain full access to a shell of the Linux system which may have a devastating effect as the attacker might be able to mess with the system and is able to take influence on every part of the system. Hence, **every asset** may be involved in this case.

Results of the work of Eshete et al. [39] on web server security indicate that default security configuration details are way to far from the recommended security configuration settings. Although operating system configuration and used software packages may not be identical at web and VoIP servers they can be similar. Hence, the results for web servers can be adopted for VoIP servers as well. Due to the discrepancy of recommended and actual security measures on a host OS and the severe damage possible, the risk has been rated as **high risk**, although the likelihood of gaining full system access is relatively low.

R3 Risk of **Unskilled Administrative Users**:

This potential risk does not come from a malicious user but from an administrative user instead. It has a potential impact on assets **asterisk system service**, **configuration files** and **commercial value** (importance for all: High).

As the possible impact can be devastating, this risk has to be considered as **high risk**.

R4 Risk of Attacks Mounted on **Errors in Software**:

Attacks which exploit coding errors in software can affect **every asset** as basically everything on a server is software and therefore can have flaws. For example, The National Institute of Standards and Technology (NIST) reports that software which is faulty in security and reliability cost the US economy 59.5 billion USD in 2002 in breakdowns and repairs [71].

Although the flaws may be small and are yet undiscovered, they may affect every piece of the system and the consequences are unpredictable. This leads to a **high risk** rating.

R5 Risk of Attacks Mounted on the **Dependency on the Security of Other Network Protocols**:

Attacks on network infrastructure protocols may restrain a subscriber's ability to contact the Asterisk server. The asset **commercial value** (importance: High) is concerned, as the subscriber may not be able to place a call as intended. Also, the assets **environment** and **SIP trunk** (importance of both: High) are involved as the server may not be able to communicate with other hosts on the system in its usual manner.

An extended understanding of network infrastructure protocols and access to network infrastructure devices can be required to carry out this kind of attack. Pelaez et al. [82] has drawn up a possible configuration for conducting an ARP spoofing attack. A **medium risk** rating was attributed for this risk as an exploitation could completely disable all communication between subscribers.

R6 Risk of **Limited Resources**:

The limitation of available resources may have implications like the inability to permanently store data of any kind on the assets **asterisk system service** (importance: High), **CDRs** (importance: High) and **event logs** (importance: Medium). The effects on the Asterisk system service, may also influence the asset **commercial value** (importance: High).

Although the decreasing cost of memory has produced systems with increasing amounts of physical memory, memory leaks still can heavily impact a system's performance and reliability in a negative way, especially in a multi-user environment [116]. The shortage of other resources may limit the system availability too. For example, VoIP media sessions are very sensitive to problems on network media, like latency or jitter [82].

If an Asterisk server uses modern server hardware, including high-performance CPU technologies and a high amount of memory, the number of simultaneous malicious request required to cause this very incident is very high. To have a more efficient attack of this kind, the attacker can perform a DDoS attack. Although the attackers required effort is quite high a successful attack can have great influence on the whole system. Furthermore, this threat is estimated to increase over the next years with the anual number of DoS or DoS like attacks being expected to rise. According to Gartner research Gartner research [41], 76 percent of surveyed IT managers expect a little increase at least. Hence, the risk is classified as **high risk**.

R7 Risk of the **Use of Insecure Transport Media**:

Attacks mounted on insecure transport media whose usage is still necessary may have an influence on a clients capability to communicate with the Asterisk server, affecting the asset **commercial value** (importance: High). Usually, the termination point of the asset **SIP trunk** (importance: High) is located within the network and may be affected as well. In case an attacker is able to modify or introduce traffic, also the assets **Asterisk system service** and **environment** (importance of both: High) may be affected.

Eavesdropping has become easier due to widely available packet sniffing tools and constitutes quite a serious problem as a security threat for companies today [112]. In addition to eavesdropping, the insertion of messages, which can be achieved quite easily, must also be taken into account. Due to the fact that such interventions will only affect one media communication at a time, this risk has been estimated as a **medium risk** for the whole system. In case the focus is put on a single subscriber, the risk for this subscriber is rated as a **high risk**.

R8 Risk of Attacks Mounted on a Faulty **Database Configuration**:

Assets involved in this kind of attack are the **CDRs** and the **Asterisk system service** (importance for both: High). If an attacker is able to gain access to the database, the CDRs may be stolen or corrupted as well as the database scheme may be modified in a way Asterisk System Service is no longer able to store records.

Due to the architecture of the reference system, access to the database will be able from the local host only, but not from a remote network host. A big portion of the risks caused by faulty DBMS configuration are network based, which have been evaluated by Neto et al. [76] using an *untrustworthiness benchmark*. Therefore, the risk of an exposed database has been estimated as **Medium Risk**.

R9 Risk of Attacks Mounted on a Faulty **Asterisk Configuration**:

Similar to a faulty configuration of OS software (Risk of Attacks Mounted on a Faulty *Host OS Configuration* (see section 5.7, R2)), attacks mounted on a faulty Asterisk configuration have several effects. On the one hand, it may allow an attacker to access the system the same way a regular subscriber is able to or may allow subscribers to use the system in an unintended manner. Such attacks may only impact the asset **commercial value** (importance: High), akin to the risk of exposed *authentication credentials*.

On the other hand, a faulty Asterisk configuration may, amongst others, allow an attacker to gain system access and may allow the attacker to modify the configuration. This means that the attacker is able to change the behavior of Asterisk. Therefore, an unauthorized alteration of Asterisk configuration files can result in major implications with possible consequences for the assets **Asterisk system service**, **configuration files**, **CDRs**, **SIP trunk**, **commercial value** (importance of all: High) and the **event log** (importance: Medium).

Although gaining access to the Asterisk configuration files is very complex, the high severity of the possible impact leads to a rating of **high risk**.

R10 Risk of **Unauthorized use of Server Resources**:

The vulnerabilities *limited resources* (see section 5.4, V6) and *unauthorized use of server resources* (see section 5.4, V10) describe different weaknesses which can be used to harm a system. The risk of both vulnerabilities may not be identical, but highly comparable due to the fact that the common goal of both is the extinction of available system resources.

According to the risk rating of the *risk of limited resources* (see section 5.7, R6), this risk has been rated as a **high risk**.

R11 Risk of Exposed **Authentication Credentials**:

An attacker will be able to subscribe to and use the system the same way a valid subscriber may use it.

Regular usage of the system will not affect any identified asset but the **commercial value** (importance: High). Because the system is equipped with a SIP trunk uplink to another telephony system operator or provider, the attacker is able to establish calls using this SIP trunk. When using the SIP trunk to connect to the public telephony network, the SIP trunk provider will demand charges for each call made depending on the selected tariff plan. Long distance calls or overseas calls may be especially expensive, and if the incident is not discovered for a long period of time, it may generate very high costs. Therefore, the attacker will incur costs which will have to be paid by the system operator company instead of the cost originator, reducing the companies financial gain. In addition, also the operator's reputation may be corrupted too.

A regular subscription may be used by the attacker to mount further attacks against the system by exploiting vulnerabilities which are only susceptible to authenticated subscribers.

Such attacks happen and succeed quite often as shown by [48], whereas the success rate is heavily influenced by the required password strength. As it is furthermore relatively easy to perform a dictionary attack (for example with John the Ripper[34]) the risk is estimated as **high risk**.

R12 Risk of **Registration Hijacking**:

Attackers which are able to take over unexpired registrations may only have an influence on asset **commercial value** (importance: High) as they will only be able to receive calls. To be able to place calls, the attacker needs to know the authentication credentials. The corresponding risk was already discussed in *risk of exposed authentication credentials* (see section 5.7, R11).

The risk created by unexpired registrations is rated as **low risk** as an attacker is not able to take any actions other than picking up an incoming media session from another subscriber for a limited period of time. Nevertheless, the valid user from whom the session was stolen might be temporarily prevented from placing calls until re-registering [104].

R13 Risk Caused by the **Dependency on the Security of External SIP Trunk Providers**:

In case an attacker take influences on the Asterisk's servers communications with the SIP trunk termination point, the assets **SIP trunk** and **Commercial Value** (importance of both: High) are concerned.

The attack can have multiple implications for the Asterisk system. To name only a few, the attacker could disable Asterisk's ability to place or receive calls via the SIP trunk by affecting the network link or the attacker could use the SIP trunk for own purposes causing additional phone charges for the system. In addition, manipulation of the media stream or gaining access to the local Asterisk installation are possible.

Although a SIP trunk offers great benefits to a system, the untrustworthy nature of data [10] received via the SIP trunk justifies a **high risk** rating. The dependency on a telephony provider and its uncontrollable configuration adds weight to this rating decision.

R14 Risk of **DoS by Unexpected User Load**:

Similar to the risk of *unauthorized use of server resources* (see section 5.7, R10), this very risk will affect the assets **Asterisk system service**, **environment** and **commercial value** (importance of all: High) if exposed.

A server running out of resources due to a high load of authorized requests requires not as many simultaneous requests as if unauthorized request are used. Usually these requests require more server-sided efforts like media re-coding. In case the initial sizing of the system was done by a professional with regard to the expected number of clients and the expected number of concurrent media sessions, the risk can be estimated as **low risk**.

R15 Risk of **Unwanted Communication**:

The only harm which could be caused by this vulnerability is to annoy system subscribers, having an impact on the user's motivation, productivity and acceptance of the system [8]. This may influence the asset **commercial value** (importance: High).

Rosenberg and Jennings [94] conclude that SPIT calls via VoIP produce only minor costs compared to PSTN line SPIT calls. In combination with the current rise of VoIP systems requires this risk to be rated as a **medium risk**.

6 Security Measures for Asterisk

This chapter presents common terms of security of an Asterisk configuration which is followed by the evaluation of detailed mitigation strategies for the previously identified risks (section 5.7).

6.1 Common Asterisk Security Measures

This section presents security considerations concerning Asterisk software and its security while in operation.

6.1.1 Configuration Aspects

The configuration files of Asterisk are used to program expected behavior into Asterisk software. Based thereon, the key to a secure operation of Asterisk software is a consistent and well-thought-through configuration.

Know what the system does

For a designer and an administrator of an Asterisk based VoIP system it is therefore essential to know every detail of the applications configuration. Though not only knowledge about the configuration entries made is required, but also knowledge about the changes in Asterisk's behavior associated with each configuration entry is a necessity.

Before any configuration change is done, one must be aware of the goals to be achieved with this exact change and one has to be certain that the planned configuration change will comply with these goals. In case this is not given, the goals should be defined and/or research on how to best implement these goals in Asterisk needs to be completed in order to avoid a try and error approach. This would entail that uneducated configuration changes and the activation of those is followed by the observation of the outcome. This behavior denotes that the person doing the configuration change is not aware of the caused change in Asterisk behavior and may therefore cause unexpected system behavior and eventually even security problems.

As a secure operation of Asterisk will rely on a secure IT infrastructure environment following the directives presented in section 2.3.1, Asterisk's configuration must be adjusted to fit into the environment. Only detailed knowledge of both the Asterisk configuration as well as the IT infrastructure enable a system designer and an administrator to create and maintain a securely operating Asterisk within a secure environment.

Keep it Simple

One major concept which helps an administrator to understand the configured Asterisk behavior more easily is the concept of keeping the configuration as simple as possible. This also decreases the likelihood of troubles caused by security issues, which are caused by active but not used modules or configuration entries.

Due to the modularity of Asterisk it is very easy to keep track of the features the software is able to offer. Each Asterisk module implements a certain set of functionality and is available in form of a Shared Object (SO) library file, which can be loaded by an application while running. During Asterisk's startup process, the desired modules are identified using the configuration file *modules.conf*, then their appropriate SO library files are loaded into Asterisk. Modules can also be loaded from Asterisk CLI and are considered as a part of Asterisk once loaded. Preventing Asterisk from loading undesirable modules is accomplished by on the one hand ensuring that Asterisk configuration does not indicate to load those modules and on the other hand by removing the possibility of loading modules per CLI. By restricting access to CLI and by removing SO library files associated with undesired modules, dynamic loading of modules at run time can be prevented. The implemented functionality of the omitted modules will not be available in Asterisk at all. Hence, these modules can not be used to mount an attack on the Asterisk system under any circumstances.

The out of the box configuration of Asterisk has a lot of modules enabled by default. Depending on the system requirements, some of them have to be disabled and removed. In addition to unnecessary modules, also an extended amount of configuration files is included in Asterisk's out of the box configuration.

A sample Asterisk configuration [66] only requires a few configuration files. This example shows the configuration files needed for an Asterisk setup which allows communication with SIP-based subscribers and allows the use of voice mailboxes:

- *asterisk.conf* - Basic Asterisk settings: Configures basic Asterisk settings as for example the user in whose context Asterisk is executed. If this file can not be found by the Asterisk core, Asterisk uses default configuration parameters and is still able to work properly [66].
- *logger.conf* - Set up of Asterisk logging: Configures which types of messages are logged to CLI, Syslog or to discreet files.
- *modules.conf* - Module configuration: Instructs Asterisk about the modules which have to be loaded at startup.
- *sip.conf* - SIP set up: This configuration file is used to make basic SIP configuration as well as configuration for SIP entities. An entity can be a simple phone or a SIP Trunk.
- *extensions.conf* - Dialplan configuration: Extensions are groups of actions that instruct Asterisk exactly how to deal with telephone calls. Extensions typically consist of a number of priorities containing multiple actions which are grouped into contexts. Each context contains a logical grouping of actions (extensions) which make up the dialing plan.
- *voicemail.conf* - Voice mail configuration: This configuration file deals with the set up of voicemail boxes available on Asterisk.

In case Asterisk is prevented from loading undesired modules, the configuration files setting up the behavior of those modules are neither touched nor applied to Asterisk any longer. Although they have no influence on Asterisk behavior, it is strongly suggested to remove those files completely to create a transparent and simple configuration.

Access to Asterisk

Asterisk offers multiple ways for accessing and processing instructions at run time like TCP/IP based AMI or its CLI. Also web interfaces may be used to access Asterisk. These methods can change the behavior of Asterisk and, if used by unauthorized people, can be used to mount an attack on the system.

By following the *keep it simple* concept, only as few as possible of these functionalities should be enabled and for those enabled, a restrictive access control has to be applied.

6.1.2 Logging Aspects

As logging is essential for monitoring the correct Asterisk behavior as well as required for debugging purposes after incidents, a logging strategy needs to defined.

- The *degree of details* defines which data is about to be logged. This can range from logging of error messages only up to logging on debug level. While logging of error messages only may include too few information, logging on debug level is very comprehensive, may make log files unmanageable and may fill up the hard disk storage. Hence, a compromise inbetween these two options should be found.
- The *logging target* defines where to store the collected log entries.
- *Log retention* defines the maximum age of log entries which should remain available on the system. An additional software like Linux software package `logrotate` should be used to keep track of the log files.

Only if these details have been figured out and have been defined as expected behavior, the log files can create a benefit for the maintainer of the system.

6.2 Mitigation Plan

As a last step to the security analysis (step VIII, section 3.4) the presentation of strategies and procedures how the risk of exploiting vulnerabilities can be minimized is carried out. For every risk identified in step VII (section 5.7), either a mitigation strategy is presented or the risk is acknowledged and accepted. In general, risks rated as *no risk* or *low risk* will be accepted without taking any risk mitigation measures. On the contrary risks rated as *medium risk* or *high risk* will always have to have a mitigation strategy.

The result of this step will be a list of measures which will minimize an attacker's ability to cause harm to the Asterisk system.

6.2.1 Technical Measures

TM1 **Environmental Monitoring Tools** can help to identify troubles with the network and are able to send notifications. Software packages like `nagios` can be used to monitor the overall vital state of the network and all incorporated servers in it and allow to identify environmental troubles quickly.

TM2 A **Host Monitoring** tool can help to identify issues before the Asterisk system is running out of resources and may take precautions. Software package `monit` could be used for this purpose.

TM3 By the use of **VLANs** on the network, devices can be merged into a logical network segment with the sole purpose of separating VoIP enabled devices from other network devices [56]. The VLAN assignment in small network environments can be done manually. For bigger network environments techniques utilizing Link Layer Discovery Protocol (LLDP), Remote Authentication Dial In User Service (RADIUS), or IEEE 802.1x NAC can be used.

Hosts within this VLAN are only able to communicate with other hosts if there is an IP routing device in between. It is highly recommended to implement ACLs on the VLAN's border router, permitting only necessary traffic.

TM4 By the use of a **dedicated VLAN** for the **SIP trunk** an attacker's influence can be minimized as the traffic is separated from other traffic on the network [56]. Members of this VLAN are only the provider's SIP trunk termination point and an interface on the Asterisk server. This interface on Asterisk server could be a virtual one which makes use of VLAN tagging to specify the VLAN, but makes use of the same physical interface.

TM5 Next to the use of VLANs, **Dedicated DHCP or DNS** servers should be used. Dedicated network server applications, which could also be hosted on the Asterisk server, should serve only the dedicated VoIP VLAN. An own DNS sub domain, whose only authoritative server is located within the protected VoIP VLAN, helps to be independent from other network infrastructure servers which might be vulnerable in one way or another.

TM6 The introduction of **QoS Mechanisms** allows a network device to prioritize and can lower the impact of troubles on the network like fully utilized uplinks.

TM7 **Access Control** measures on network infrastructure devices, which are defined by IT policies, help to disable unauthorized access and unauthorized changes to the network environment.

TM8 Use **Dedicated Disk Partitions** for growing data. By separating the HDD areas into partitions, the file system folders which are expected to grow can be split from executables or other files and folders required for an ordinary system operation. In case the partition for growing data reaches its limits, the regular system operation may continue with only features causing this data growth not being fully functional any longer. Data like event logs or recorded mail box messages are considered as growing data.

TM9 **Dedicated Administrative User Accounts** will assign a unique name to each administrative user and will allow each user to have a personal password. A common administrative account with a password known to a group of people must not exist.

TM10 Install and operate **as little Software as Possible** on the server system. Every installed software bears the risk of configuration error which could be a door opener for an attacker.

TM11 **Application Containers** allow to hide parts of the file system and have an application's ability to restrict the access to files and folders. This is achieved through creating a new virtual file system root folder `/` on an existing physical file folder. Applications executed within this changed environment are restricted to file system access within or underneath their virtual file system root. Linux tool `chroot` is used for the creation of such a virtual root folder.

TM12 The **Configuration Files** of every software package as part of the Linux system have to describe a predictable system behavior. These software packages should be restricted to provide their duties while having, for example, unused software modules unregistered or uninstalled.

TM13 Following the concept of **Confining an Application to have as few permissions as possible to do their Job** will limit an attackers ability to do harm to the system after a weakness has been exploited successfully.

TM14 **Applying an Aligned Password Policy** will help minimizing the attackers chance of retrieving passwords using a dictionary attack. This password policy can regulate the minimum length, the maximum age after which the password needs to be changed and groups of characters like small letters, capital letters or numbers which needs to be part of every password. The policy must be enforced by the system.

TM15 A **Temporary Blacklist** can help to block undesired requests at an early stage of request processing. Although Raghavan et al. [92] presents multiple ways for detecting and mitigating DoS attacks, an exact traffic pattern can not be defined [62]. This is the reason for the shaping of incoming network traffic to be done based on its originating IP address only. This helps to reduce the required load and minimizes the impact on the system. In case multiple invalid registration request will be received from the same IP address within a certain period of time, the sending IP address will be blocked completely for a defined period of time. In case an Internet provider uses a NAT-based setup, multiple valid requests may originate from a single IP address, which must be considered while defining limits.

Asterisk software offers a possibility to blacklist subscribers based on their extension number, which requires a valid registration. As the temporary blacklist will require to block clients based on their IP address and has to function without a valid registration, this Asterisk function can not be used. Instead, `iptables` Linux firewall may be used for this purpose.

TM16 A **Database Configuration Files Change Check** script helps to identify every configuration change of the database, similar to the Asterisk configuration consistency check script presented previously.

TM17 Automated **Data Consistency Checks** based on triggers which will fire every time a record is inserted into the CDR database can assure the integrity of the database content. If unauthorized changes are detected, notifications can be sent automatically.

TM18 The **Asterisk Configuration Files** have to describe a safe and predictable behavior of Asterisk software. Measures and Guidelines how to tighten Asterisk's configuration can be found in section 6.1.

TM19 Scripts performing an **Automated Configuration Change Check** help to identify any changes of Asterisk's configuration files. Such scripts will use checksum algorithms (MD5, SHA-1) to detect changes in configuration files, and should be executed at least every few minutes. Linux scheduler software `cron` can be used to run the check frequently.

An action taken in case a configuration change was detected may be a simple notification sent to an administrator.

TM20 Asterisk's configuration allows to define the **Maxload Parameter** which tells Asterisk to accept new incoming calls only if the Linux system parameter *load average* is below a defined value. This not only makes it impossible for the attacker to overload the system, but will also result in Asterisk refusal to accept incoming session setup requests from non-malicious subscribers. The exact numerical value to be configured needs to be evaluated carefully as Asterisk may stop to process new request even though enough resources are still available.

TM21 Define the **Communication Channels** of each software. Each software package has its defined duties and may communicate with the network. An application's ability to initiate and receive network communication should be restricted to predefined ports or groups of ports. This can be enforced using network firewall software, which helps protecting the system by strictly controlling every data packet sent to or received from the network. A popular firewall software for Linux systems is the `iptables` package.

TM22 **Blacklists** can be used to ban foreign system subscribers from being able to establish sessions with local system subscribers registered on the Asterisk system. Adding a subscriber number to the blacklist is usually a manual task. This blacklist is based on the identification of a subscriber (e.g. subscriber number), whereas the mitigation measure *Temporary Blacklist* (see section 6.2, TM15) uses IP addresses instead.

TM23 **Grey Listing Mechanisms** could also be used to introduce a sort of automatic blacklisting. In case a foreign system subscriber calls the first time, a voice announcement requiring the caller to press a numbered key can help to determine if the caller is a real person or an automated dialer. Turing test as described by Bradford et al. [21] can be used to develop such a test. The session is only going to be established in case the caller passes this test, additionally the caller will be white listed which allows to bypass this test during the next call. Otherwise, the foreign system subscriber will automatically be blacklisted for a limited period of time.

TM24 **Encryption Mechanisms** help to protect payload traffic in case an attacker is able to perform eavesdropping. Calabrese [24] gives an overview of cryptographic principals which can be applied to SIP systems.

TM25 The use of a **Public-Key Infrastructure (PKI)** to verify a subscriber's identity makes the use of authentication credential strings obsolete as they are replaced by certificates [84]. This will significantly lower the number of possible attacks, although risk of theft of certificates may not be ignored and has to be mitigated separately.

TM26 By introducing additional measures on the encryption methods used, the chance of a successful exploit can be lowered. Tangwongsan et al. [112] presented an experimental approach how the usage of **Diffie-Hellman Protocol**[51] for exchanging shared keys between communicating hosts can be used to ensures that every media session will use a **Unique Encryption Key**.

TM27 A daily **Work Log**, summing up Asterisk's activities of the day could be sent by email to the person responsible. This work log contains information like the number of established connections or the number of failed registration attempts and will help to identify illegal use of the system.

TM28 By the use of **Log Rotation** mechanisms, continuously growing event log files can be avoided. Log rotation software is able to delete or archive older log files and makes sure the current log file does not grow indefinitely.

6.2.2 Organizational Measures

OM1 An **Environmental Security Procedure** is required to govern environmental security. This includes regulations regarding physical access to the server resources as well as regulations about air-conditioning systems.

OM2 A **System Monitoring Procedure** is necessary to regulate how the system and its environment are monitored. Such a procedure if implemented, allows to detect troubles early and therefore can help to minimize down times. This procedure includes detailed information about the metrics to be monitored, critical values for each metric and a procedure how and whom to alert in the case of critical values being surpassed.

OM3 The **System Access Procedure** governs who is able to access the system for administrative or log viewing purposes. This includes the regulation an assessment of technical staff on trustworthiness and skills.

OM4 **Human Resources (HR) Standards** defines a profile of qualifications of a user before administrative permissions are granted. The standard also has to include strategies as to how this qualifications can be ensured over time.

OM5 **Service Contracts** negotiated between the telephony provider and the Asterisk system operator (customer) regulate the mutual understanding of both sides duties and responsibilities. Amongst other things, this may include regulations about the maximum recovery time allowed in case of a service outage on the provider's side, regulations about fees the customer has to pay, regulations about used encryption types, if applicable, as well as regulations about authentication mechanisms and authentication details concerned.

OM6 **System Software Update and Installation Procedures** govern general issues. For example, the procedure how to apply and who is allowed to install updates on the system has to be defined. A **System Update Strategy** which controls how system software updates have to be applied is required. By assuring the latest patches are applied on a regular basis, the risk of newly discovered flaws being exploited can be minimized.

OM7 **Minimum Requirements** for the use of **Cryptographic Functions** can lower the system's liability to threats in case encryption is implemented, as suggested previously. HMAC algorithms in combination with cryptographic hash value functions which are currently considered as safe form a risk which can be accepted. Hence, policies need to assure that the security of the used Algorithms is reviewed and adopted accordingly on a regular basis.

6.3 Effectiveness Matrix: Vulnerabilities – Mitigation Measures

This section establishes the mapping between the previously identified vulnerabilities (section 5.4) and previously identified mitigation measures (section 6.2). The columns show the list of vulnerabilities whereas the rows list the identified mitigation measures. An *X* indicates that the according vulnerability can be mitigated through the respective mitigation measure, following the same scheme as at *Effectiveness Matrix: Threats – Vulnerabilities* (see section 5.5)

	Vulnerability														
Mitigation Measure	V1 Physical Environment	V2 Host OS Misconfiguration	V3 Unskilled Administrative Users	V4 Errors in Software	V5 Dependency on the Security of Other Network Protocols	V6 Limited Resources	V7 Use of Insecure Transport Media	V8 Database Misconfiguration	V9 Asterisk Misconfiguration	V10 Unauthorized use of Server Resources	V11 Authentication Credentials	V12 Registration Hijacking	V13 Dependency on the Security of the External SIP Trunk Provider	V14 DoS by Unexpected User Load	V15 Unwanted Communication
TM1 Environmental Monitoring	X				X	X				X				X	
TM2 Host Monitoring						X				X				X	
TM3 use of VLANs					X		X					X			
TM4 Dedicated Trunk VLAN													X		
TM5 Dedicated DHCP and DNS					X										
TM6 QoS Mechanisms						X	X								
TM7 Access Control					X		X								
TM8 Dedicated Disk Partitions						X									
TM9 Dedicated Admin Accounts			X								X				
TM10 as few Software as Possible		X		X											
TM11 Application Containers		X		X					X						
TM12 Software Configuration Files		X		X											
TM13 Confine Application		X		X											
TM14 Apply Password Policy			X								X				
TM15 Temporary Blacklist															X
TM16 DB Config Change Check								X							
TM17 DB Data Consistency Check				X				X							
TM18 Asterisk Configuration Files									X						
TM19 Asterisk Conf Change Check									X						
TM20 Maxload Parameter						X									
TM21 Communication Channels				X			X								
TM22 Blacklists															X
TM23 Grey Listing															X
TM24 Encryption							X								
TM25 PKI							X								
TM26 Unique Encryption Keys							X								
TM27 Work Log				X					X	X					
TM28 Log Rotation						X									
OM1 Environmental Security Proc.	X					X									

	Vulnerability														
Mitigation Measure	V1 Physical Environment	V2 Host OS Misconfiguration	V3 Unskilled Administrative Users	V4 Errors in Software	V5 Dependency on the Security of Other Network Protocols	V6 Limited Resources	V7 Use of Insecure Transport Media	V8 Database Misconfiguration	V9 Asterisk Misconfiguration	V10 Unauthorized use of Server Resources	V11 Authentication Credentials	V12 Registration Hijacking	V13 Dependency on the Security of the External SIP Trunk Provider	V14 DoS by Unexpected User Load	V15 Unwanted Communication
OM2 System Monitoring Proc.		X													
OM3 System Access Proc.			X				X								
OM4 HR Standards			X												
OM5 Service Contracts													X		
OM6 Software Update Proc.				X											
OM7 Requirements (Crypto Functions)							X								

Table 6.1: Vulnerability – Mitigation Measure Matrix

7 Reflection on Open Source Effects

This chapter will give a short overview of advantages and drawbacks of open source software solutions compared to closed source products. At first, some common issues like costs or expected maintenance effort are discussed, followed by a comparison of security concerns for open as well as closed source products.

Open source software is described as software which is freely available and can be downloaded from the Internet without incurring charges. The source code is available under any circumstances and may be accompanied by precompiled executables.

Software, for which it is required to pay is considered as closed source in this chapter although the source code may be publicly available. In many cases, companies treat open source software for which money is claimed the same way as closed software.

7.1 Comparison of Common Concerns

The following list will analyze common implications on the use of open source software.

- **Cost:**

  As per definition in this thesis, open source software can be downloaded for free, the costs for acquiring such a software are considerably low. Only charges from the provider for the download may apply. On the other hand, closed software will most certainly require a substantial financial effort.

  In addition to the acquisition costs, the costs for the initial setup has to be taken into account. When purchasing closed source software, setup support may be included in the price which decreases the required setup time. Open source products will not come with any installation support. This results in longer working time for installation and configuration and therefore causes higher setup costs as compared to a similar closed source software product.

- **Required manpower:**

  The necessary time effort to apply upgrades and to do other maintenance work on software products may be a little higher for open source software products. This is due to the fact that closed software products often come with maintenance agreements which allow easier and faster retrieval of updates.

- **How to get Support:**

  Due to the freely available source code in open source software products, troubleshooting and debugging on source code basis becomes possible. As the community supporting open source products is quite big, a lot of information is available on the Internet. Depending on the product, also the community for closed source products can be quite extensive. Many companies of closed source software products also offer support services for their products against a fee.

- **Know How Management:**

> As open source software usually does not come with setup assistance or the possibility to use a support service, more know how is required compared to closed source products.

To sum it up, open source software products may be cheaper in direct costs, but do require more know how for setup and operation of the software. This extensive know how has to come from experience which entails that the IT staff may be more expensive, further increasing the TCO for open source products. Therefore, a clear statement as to which approach is cheaper can not be given. This highly depends on a company's general IT approach and their IT staff's know how.

7.2 Security Concerns

Within the open source and free software community it is argued that making source codes available to everybody has a positive impact on security, as a large community of users and experts can pore over the code and obliterate vulnerabilities [93]. In the crypto community it is standard practice to know the design of a system, only hiding the key. On the contrary it can be argued that if software code is publicly available, a malicious user also has the opportunity to search the code for vulnerabilities. Based on this, Anderson [3] injected the question as to *whom does openness help more, attack or defense?*

Anderson [3] argues, that *in a perfect world, and for systems large and complex enough for statistical methods to apply, the attack and the defense are helped equally. Whether systems are open or closed makes no difference in the long run.*

It is also argued that the interesting questions lie in the circumstances under which this symmetry can be broken in practice. An interaction between openness of the system and its security is also a matter of functionality and the question as to *what should a secure system do?* will have different answers. While security might mean for the user the repulse of 'evil hackers on the Internet', whoever they might be, security for the vendor means growing the market and crushing the competition.

Therefore the real tensions between the open and closed system communities will be defined by struggles for power and control over standards in the future.

8 Conclusion

Despite the urge of the rapidly growing number of installed VoIP systems, implementing security features should always be a matter of course. As practice shows, organizations are often dazzled by the overwhelming advantages a VoIP system offers. As a matter of fact, security is a concern which is often forgotten or is not considered well enough nowadays.

The reasons for these disregards towards security concerns range from unskilled responsible staff, who is just not aware of or unable to identify security implications, or an responsible persons attitude of *I Don't Care About Security, Such Things Won't Happen To Us*.

Nevertheless, recent events showed that security is an important factor of success. The initial additional expenses on security measures will pay of quite fast as they will make the system more resistant against attacks of any kind, although total security can not be achieved ever.

This thesis presents an approach to evaluate the security of an reference system. It is illustrated as a process of eight steps in which the expected behavior of the system is analyzed. Based on a detailed evaluation of all system activities, assets, threats and vulnerabilities are identified. The system activities, which have been described by UCs and activity flow charts, show the expected behavior of the system, which was used to identify all possible points of contact for attackers.

On the foundation of the detailed system description, the valueable system parts to be protected, the assets, have been identified as well as threats and vulnerabilities have been derived.

As a next step, the risk of vulnerabilities getting exposed was calculated using the estimated financial damage in case of exposure and the estimated probability of an exposure. With all informations collected during the analysis so far, mitigation measures to reduce the identified risks are shown.

The identified mitigation measures have been devided into the group of technical measures and the group of organizational measures. Technical measures concern either the Asterisk configuration, the Linux environment or both. The presented measures identify well known and well established methods to reduce the risk of exploiting security issues and cover multiple areas, including system monitoring or the prevention from troubles caused by missing resources. A list of mitigation measures can be found in table 6.1.

This thesis identified a total of 7 system assets, 53 possible threats to which the system is exposed, 15 vulnerabilities and risks as well as 35 mitigation measures have been proposed.

Although there may be a lot of ways to harden a system, this thesis presented one possible approach which could be used to minimize the risk of security incidents. In the end it is important to have precautions against security threats implemented, whereat the actual approach choosen is of lesser significance.

List of Figures

List of Tables

Acronyms

ACL Access Control List. 9, 17, 59

ALG Application Layer Gateway. 59

AMI Asterisk Manager Interface. 6

ARP Address Resolution Protocol. 50

BIOS Basic Input/Output System. 14

CDR Call Detail Record. 6, 26, 39, 44, 46, 50, 54, 56, 57, 61

CEL Call Event Log. 6

CISSP Certified Information Systems Security Professional. 8

CLI Command Line Interface. 6, 26, 42

CPU Central Processing Unit. 6, 12, 13, 48, 49, 55, 59, 67

DAHDi Digium Asterisk Hardware Driver Interface. 5

DDoS Distributed Denial of Service. 55

DHCP Dynamic Host Configuration Protocol. 50, 60

DMZ Demilitarized Zone. 17, 18, 67

DNS Domain Name System. 4, 49, 50, 60

DoS Denial of Service. 8, 45–50, 55, 59

DS Differentiated Services. 18

DTMF Dual-tone multi-frequency signaling. 40, 47

GPLv2 GNU General Public License Version 2. 1, 6, 7, 12

GRUB Grand Unified Bootloader. 14

GUI Graphical User Interfaces. 6

HDD Hard Disc Drive. 26, 38, 40, 52, 62

HMAC Hash-based Message Authentication Code. 34, 35, 45, 48, 55, 57, 59, 61

HTTP Hypertext Transfer Protocol. 2, 3, 17, 18, 27, 59

HTTPS Hypertext Transfer Protocol Secure. 27

IAX2 Inter-Asterisk Exchange. 5

ID Identifier. 6, 13, 27, 44–49

IETF Internet Engineering Task Force. 3

IP Internet Protocol. 1, 2, 4–6, 18, 26, 50, 59

ISDN Integrated Services Digital Network. 5

ISO International Organization for Standardization. 2

IT Information Technology. 8–10, 23, 52, 60–63

ITIL Information Technology Infrastructure Library. 10

ITU-T International Telecommunication Union - Telecommunication. 3, 16

LDAP Lightweight Directory Access Protocol. 13

LLDP Link Layer Discovery Protocol. 59

MAC Media Access Control. 18

MBR Master Boot Record. 14

MD4 Message-Digest Algorithm 4. 55

MD5 Message-Digest Algorithm 5. 35, 45, 55, 60

MGCP Media Gateway Control Protocol. 5

MITM Man-in-the-middle Attack. 47, 48

NAC Network Access Control. 18, 59, 61

NAT Network Address Translation. 17

NIS Network Information Service. 13

NSA United States National Security Agency. 15

OS Operating System. 12–14, 26, 33, 49, 51, 52, 57, 61

OSI Open Systems Interconnection. 2, 18, 67

PAT Port Address Translation. 17

PBX Private Branch Exchange. 1, 5, 7, 44

PCI Peripheral Component Interconnect. 5

PKI Public-Key Infrastructure. 58

PSTN Public Switched Telephone Network. 4, 5, 26, 44

QoS Quality of Service. 18, 60

RADIUS Remote Authentication Dial In User Service. 59

RAM Random Access Memory. 26, 52

RPM RPM Package Manager. 26

RRD Round-Robin Database. 19

RTCP Real-time Transport Control Protocol. 3

RTP Real-time Transport Protocol. 3, 39–41, 47

SCCP Skinny Call Control Protocol. 3, 5

SDLC Systems Development Life Cycle. 8

SDP Session Description Protocol. 3, 38

SELinux Security-Enhanced Linux. 15

SHA-1 Secure Hash Algorithm 1. 35, 45, 55, 60

SIP Session Initiation Protocol. 3–5, 26–28, 34–36, 38, 39, 41, 44, 46–51, 53–58, 61, 67

SMTP Simple Mail Transfer Protocol. 2, 3

SNMP Simple Network Management Protocol. 15, 19

SPIT Spam over Internet Telephony. 49, 53

SRTCP Secure Real-time Transport Control Protocol. 3, 27

SRTP Secure Real-time Transport Protocol. 3, 27

SSH Secure Shell. 14, 33, 41, 42, 47, 48

TCO Total Cost of Ownership. 1

TCP Transmission Control Protocol. 2, 3, 6, 18, 41, 59, 61

TCP/IP Transmission Control Protocol / Internet Protocol. 2, 3, 67

TLS Transport Layer Security. 3, 27, 58

UC Use Case. vi, 20, 21, 27–34, 36, 37, 39, 41, 42, 44, 67, 68

UDP User Datagram Protocol. 2, 3, 5, 61

URL Uniform Resource Locator. 17

vCPU virtual Central Processing Unit. 26

VLAN Virtual Local Area Network. 18, 59–61

VoIP Voice over Internet Protocol. 2, 4, 5, 7, 28, 44, 49, 50, 52, 53, 59, 60

WCCP Web Cache Communication Protocol. 17

XMPP Extensible Messaging and Presence Protocol. 5

Bibliography

[1] S. Aidarous et al. *Managing IP networks: Challanges and Opportunities*. Wiley-IEEE Press, 2003.

[2] J. H. Allen et al. *Software Security Engineering - A Guide for Project Managers*. Pearson Education, Inc., 2008.

[3] R. Anderson. “Security in open versus closed systems - the dance of Boltzmann, Coase and Moore”. In: *Open Source Software: Economics, Law and Policy* (2002).

[4] R. J. Anderson. *Security Engineering, Second Edition*. Wiley Publishing, Inc., 2008.

[5] N. Antonopoulos and L. Gillam. *Cloud Computing: Principles, Systems and Applications*. Springer-Verlag London Limited, 2010.

[6] F. Armour and G. Miller. *Advanced Use Case Modeling: Software Systems*. Addison-Wesley Professional, 2001.

[7] Austria. *Datenschutzgesetz 2000 (BGBl. I Nr. 165/1999): Section 3, Datensicherheit.* 2011.

[8] M. A. Azad and R. Morla. “Multistage SPIT Detection in Transit VoIP”. In: *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on* (2011).

[9] R. Baclit et al. *Foundations of CentOS Linux: Enterprise Linux On the Cheap*. apress, 2009.

[10] V. A. Balasubramaniyan et al. “PinDr0p: Using Single-Ended Audio Features To Determine Call Provenance”. In: *Proceedings of the 17th ACM conference on Computer and communications security* (2010).

[11] C. Banerjee and S. K. Pandey. *Research on Software Security Awareness: Problems and Prospects*. 2010.

[12] M. Bauer. “Paranoid Penguin: Designing and Using DMZ Networks to Protect Internet Servers”. In: *Linux Journal* (2001).

[13] M. Bauer. “Paranoid penguin: Introduction to seLinux”. In: *Linux Journal* (2007).

[14] M. Bauer. “Paranoid penguin: Samba security, Part III”. In: *Linux Journal* (2009).

[15] M. Baugher et al. *The Secure Real-time Transport Protocol (SRTP)*. RFC 3711 (Proposed Standard). Updated by RFC 5506 Accessed on 07 July 2012. Internet Engineering Task Force, Mar. 2004. URL: http://www.ietf.org/rfc/rfc3711.txt.

[16] E. Beulen and V. Tiwari. “Parallel Transitions in IT Outsourcing: Making It Happen”. In: *Lecture Notes in Business Information Processing, 2010, Volume 55* (2010).

[17] S. G. Bhirud and V. Katkar. “Light weight approach for IP-ARP spoofing detection and prevention”. In: *Internet (AH-ICI), 2011 Second Asian Himalayas International Conference on* (2011).

[18] H. Bidgoll. *The Internet Encyclopedia - Volume 1*. John Wiley & Sons, Inc., 2004.

[19] D. P. Bovet and M. Cesati. *Understanding the Linux Kernel*. O’Reilly Media, Inc., 2006.

[20] R. Braden. *Requirements for Internet Hosts - Communication Layers*. RFC 1122 (Standard). Updated by RFCs 1349, 4379, 5884, 6093, 6298 Accessed on 07 July 2012. Internet Engineering Task Force, Oct. 1989. URL: http://www.ietf.org/rfc/rfc1122.txt.

[21] P G. Bradford and M. Wollowski. "A formalization of the Turing test". In: *SIGART Bulletin , Volume 6 Issue 4* (1995).

[22] A. Bremler-Barr, R. Halachmi-Bekel, and J. Kangasharju. "Unregister Attacks in SIP". In: *Secure Network Protocols, 2006. 2nd IEEE Workshop on* (2006).

[23] M. Burnett. *Perfect Passwords - Selection, Protection, Authentication*. Syngress Publishing, Inc., 2006.

[24] T. Calabrese. *Information Security Intelligence: Cryptographic Principles & Applications*. Delmar Learning, a division of Thomson Learning, Inc., 2004.

[25] F. Cao and S. Malik. "Vulnerability analysis and best practices for adopting IP telephony in critical infrastructure sectors". In: *Communications Magazine, IEEE* Volume: 44 , Issue: 4 (2006).

[26] J.D. Case et al. *Simple Network Management Protocol (SNMP)*. RFC 1098. Accessed on 07 July 2012. Internet Engineering Task Force, Apr. 1989. URL: http://www.ietf.org/rfc/rfc1098.txt.

[27] CentOS Project. *The Community ENTerprise Operating System*. http://www.centos.org/. Accessed on 07 July 2012. 2012.

[28] U. Chandran and D. Zhao. "SS-KTC: A High-Testability Low-Overhead Scan Architecture with Multi-level Security Integration". In: *IEEE Xplore* 27th IEEE VLSI Test Symposium, 2009. VTS '09 (2009).

[29] E. Y. Chen. "Detecting DoS Attacks on SIP Systems". In: *VoIP Management and Security, 2006. 1st IEEE Workshop on* (2006).

[30] Z. Chen et al. "Research on Man-in-the-Middle Denial of Service Attack in SIP VoIP". In: *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on* (2009).

[31] L. St. Clair et al. "Password Exhaustion: Predicting the End of Password Usefulness". In: *Lecture Notes in Computer Science* Volume 4332, Information Systems Security (2006).

[32] P. Clark. *Contingency planning and strategies*. 2010.

[33] U.S. Department of Commerce. "The Keyed-Hash Message Authentication Code (HMAC)". In: *Federal Information Processing Standards Publication 198 (NIST FIPS-198)* (2002).

[34] Openwall Community. *John the Ripper password cracker*. http://www.openwall.com/john/. Accessed on 07 July 2012. 2012.

[35] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. RFC 2246 (Proposed Standard). Obsoleted by RFC 4346, updated by RFCs 3546, 5746, 6176 Accessed on 07 July 2012. Internet Engineering Task Force, Jan. 1999. URL: http://www.ietf.org/rfc/rfc2246.txt.

[36] Digium, Inc. *Asterisk - The Open Source Telephony Projects*. http://www.asterisk.org/. Accessed on 07 July 2012. 2010.

[37] D. Endler and M. Collier. *Hacking Exposed VoiIP: Voice over IP Security Secrets and Solutions*. McGraw-Hill Osborne Media, 2006.

[38] J. Epstein. *Scalable VoIP Mobility - Integration and Deployment*. Elsevier Inc., 2009.

[39] B. Eshete, A. Villafiorita, and K. Weldemariam. "Early Detection of Security Misconfiguration Vulnerabilities in Web Applications". In: *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on* (2011).

[40] R. Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard). Updated by RFCs 2817, 5785, 6266 Accessed on 07 July 2012. Internet Engineering Task Force, June 1999. URL: http://www.ietf.org/rfc/rfc2616.txt.

[41] Forrester Research, Inc. *Changing landscape of DDos threats and protection*. http://verisigninc.com/assets/whitepaper-ddos-threats-protection-forrester.pdf. Accessed 19 June 2012. 2009.

[42] B. Foster and F. Andreasen. *Basic Media Gateway Control Protocol (MGCP) Packages*. RFC 3660 (Informational). Accessed on 07 July 2012. Internet Engineering Task Force, Dec. 2003. URL: http://www.ietf.org/rfc/rfc3660.txt.

[43] Free Software Foundation, Inc. *GNU GENERAL PUBLIC LICENSE*. http://www.gnu.org/licenses/gpl.txt. Accessed on 06 July 2012. 2007.

[44] C. Fuchs et al. "Detecting VoIP based DoS attacks at the Public Safety Answering Point". In: *ASIACCS '08: Proceedings of the 2008 ACM symposium on Information, computer and communications security* (2008).

[45] M. Gardner. *MATHEMATICAL GAMES: The fantastic combinations of John Conway's new solitaire game LIFE*. http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm. Accessed on 06 July 2012. 1970.

[46] Gartner, Inc. *Gartner Perspective: IT Spending 2010*. http://www.financialexecutives.org/fly/?code=FITJan10. Accessed on 27 June 2012. 2009.

[47] Gartner, Inc. *Press Release: Gartner says Mobile VoIP Poses a Huge Challenge for Traditional Mobile Voice Providers*. http://www.gartner.com/it/page.jsp?id=963712. Accessed on 07 July 2012. 2009.

[48] M. Gruber et al. "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*. Oct. 2011.

[49] M. Handley, V. Jacobson, and C. Perkins. *SDP: Session Description Protocol*. RFC 4566 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, July 2006. URL: http://www.ietf.org/rfc/rfc4566.txt.

[50] S. Harris. *Mike Meyers' CISSP(R) Certification Passport*. McGraw-Hill Professional, 2002.

[51] M. E. Hellman and W. Diffie. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory, Vol. IT-22, NO. 6* (1976).

[52] J. Herrera-Joancomarti, J. Prieto-Blazquez, and J. Castella-Roca. "A secure electronic examination protocol using wireless networks". In: *IEEE Xplore* Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on (2004).

[53] J. W. Hong, M. A. Bauer, and J. M. Bennett. "The role of directory services in network management". In: *ACM* CASCON '92 Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 2 (1992).

[54] J. D. Howard. "An Analysis of security incidents on the Internet 1989-1995". PhD thesis. Cernegie Mellon University, Pittsburg, Pennsylvania, 1997.

[55] B. Ingram. *FED-STD-1037c, Telecommunications: Glossary Of Telecommunication Terms*. http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm. Accessed on 07 July 2012. 1996.

[56] V. Iossifov, T. Totev, and A. Tochatschek. “Experiences in VoIP telephone network security policy at the University of Applied Sciences (FHTW) Berlin”. In: *CompSysTech ’07: Proceedings of the 2007 international conference on Computer systems and technologies* (2007).

[57] W. Jansen. *Directions in Security Metrics Re-search*. http://csrc.nist.gov/publications/nistir/ir7564/nistir-7564_metrics-research.pdf. Accessed on 07 July 2012. 2009.

[58] A. Jaquith. *Security Metrics: Replacing Fear, Uncertainty and Doubt*. Addison-Wesley Professional, 2007.

[59] K. Rafferty. “Using the Red Hat Package Manager”. In: *Linux Journal* (2000).

[60] D. Kaye. *Strategies for Web hosting and managed services*. John Wiley & Sons, Inc., 2002.

[61] J. Klensin. *Simple Mail Transfer Protocol*. RFC 2821 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, Apr. 2001. URL: http://www.ietf.org/rfc/rfc2821.txt.

[62] J. Kline et al. “Traffic anomaly detection at fine time scales with bayes nets”. In: *In Proceedings of the Third International Conference on Internet Monitoring and Protection* (2008).

[63] J. Kuch. “Technologische Methoden und Mechanismen zur Erhöhung der Vertraulichkeit von VoIP-Verbindungen”. MA thesis. Vienna University of Technology, Austria, 2008.

[64] J. Lewis. *SDLC 100 Success Secrets - Software Development Life Cycle (Sdlc) 100 Most Asked Questions, Sdlc Methodologies, Tools, Process and Business Models*. Lulu.com, 2008.

[65] M. Lipow. “Number of Faults per Line of Code”. In: *Software Engineering, IEEE Transactions on, Volume: SE-8 , Issue: 4* (1982).

[66] L. Madsen, J. van Meggelen, and R. Bryant. *Asterisk, The Definitiv Guide, Third Edition*. O’Reilly Media, Inc., 2011.

[67] makelinux.net. *Interactive map of Linux kernel*. http://www.makelinux.net/kernel_map/. Accessed on 06 July 2012. 2008.

[68] D. Mathur. “Business Process Transformation Grid: An Empirical Model for Strategic Decision Making Towards IT Enabled Transformations”. In: *Lecture Notes in Business Information Processing, Volume 37, Part 3* (2009).

[69] G. McGraw. *Software Security: Building Security In*. Addison Wesley Professional, 2006.

[70] A. M. McKenzie. *Telnet Protocol specifications*. RFC 495. Updated by RFC 562 Accessed on 07 July 2012. Internet Engineering Task Force, May 1973. URL: http://www.ietf.org/rfc/rfc495.txt.

[71] N. R. Mead. “Security Requirements Engineering”. In: *Build Security In* (2006).

[72] N. Mendes et al. “Assessing and Comparing Security of Web Servers”. In: *Dependable Computing, 2008. PRDC ’08. 14th IEEE Pacific Rim International Symposium on* (2008).

[73] L. Mui and E. Pearce. *X Window System Volume 8: X Window System Administrator’s Guide for X11 Release 4 and Release 5, 3rd edition*. O’Reilly and Associates, 1993.

[74] S. Myagmar, A.J. Lee, and W. Yurcik. “Threat Modeling as a Basis for Security Requirements”. In: *In Proceedings of the Symposium on Requirements Engineering for Information Security*. 2005.

[75] M. Nassar et al. “Holistic VoIP intrusion detection and prevention system”. In: *IPT-Comm ’07: Proceedings of the 1st international conference on Principles, systems and applications of IP telecommunications* (2007).

[76] A. A. Neto and M. Vieira. “Benchmarking Untrustworthiness in DBMS Configurations”. In: *Dependable Computing, 2009. LADC ’09. Fourth Latin-American Symposium on* (2009).

[77] K. Nichols et al. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC 2474 (Proposed Standard). Updated by RFCs 3168, 3260 Accessed on 07 July 2012. Internet Engineering Task Force, Dec. 1998. URL: http://www.ietf.org/rfc/rfc2474.txt.

[78] W. Odom. *CCNA 640-802 Official Cert Library, Updated Third Edition*. Pearson Education, Inc., 2012.

[79] Cabinet Office. *ITIL® glossary and abbreviations*. ITIL® is a registered trade mark of the Cabinet Office. Crown copyright 2012. All rights reserved. Material is reproduced with the permission of the Cabinet Office under delegated authority from the Controller of HMSO. 2011.

[80] A.L. Opdahl and G. Sindre. “Experimental comparision of attack trees and misuse cases for security threat identification”. In: *In Press* Information and Software Technology (2008).

[81] Packet Storm. *Files – Packet Storm*. http://packetstormsecurity.org/DoS. Accessed 17 July 2012. 2012.

[82] J. C. Pelaez et al. “Misuse Patterns in VoIP”. In: *Proceedings of the 14th Conference on Pattern Languages of Programs* (2007).

[83] R. Petersen. *Ubuntu 9.04 - System Administration and Security*. 2009.

[84] J. Peterson and C. Jennings. *Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)*. RFC 4474 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, Aug. 2006. URL: http://www.ietf.org/rfc/rfc4474.txt.

[85] R. Petrusel and P. L. Stanciu. “Making Recommendations for Decision Processes Based on Aggregated Decision Data Models”. In: *Lecture Notes in Business Information Processing, 2012, Volume 117, Part 9* (2012).

[86] T. Porter. *Practical VoIP Security*. Syngress Publishing, Inc., 2006.

[87] J. Postel. *Internet Protocol*. RFC 791 (Standard). Updated by RFC 1349 Accessed on 07 July 2012. Internet Engineering Task Force, Sept. 1981. URL: http://www.ietf.org/rfc/rfc791.txt.

[88] J. Postel. *Transmission Control Protocol*. RFC 793 (Standard). Updated by RFCs 1122, 3168, 6093 Accessed on 07 July 2012. Internet Engineering Task Force, Sept. 1981. URL: http://www.ietf.org/rfc/rfc793.txt.

[89] J. Postel. *User Datagram Protocol*. RFC 768 (Standard). Accessed on 07 July 2012. Internet Engineering Task Force, Aug. 1980. URL: http://www.ietf.org/rfc/rfc768.txt.

[90] PostgreSQL Global Development Group. *PostgreSQL*. http://www.postgresql.org/. Accessed on 07 July 2012. 2012.

[91] N.R. Prasad. “Threat Model Framework and Methodology for Personal Network”. In: *Communication Systems Software and Middleware, COMSWARE*. 2007.

[92] S. V. V. Raghavan and E. Dawson. *An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks*. Springer India Pvt. Ltd. 2011, 2011.

[93] E. S. Raymond. *The Cathedral and the Bazaar*. http://www.catb.org/~esr/writings/homesteading/cathedral-bazaar/. Version 3.0, Accessed on 25 July 2012. 2000.

[94] J. Rosenberg and C. Jennings. *The Session Initiation Protocol (SIP) and Spam*. Request for Comments: 5039. 2008.

[95] J. Rosenberg et al. *SIP: Session Initiation Protocol*. RFC 3261 (Proposed Standard). Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141 Accessed on 07 July 2012. Internet Engineering Task Force, June 2002. URL: http://www.ietf.org/rfc/rfc3261.txt.

[96] P. Saint-Andre. *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*. RFC 3921 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, Oct. 2004. URL: http://www.ietf.org/rfc/rfc3921.txt.

[97] M.B. Sarder, K. Rogers, and E. Prater. "Outsourcing swot analysis for some us industry". In: *In: Technology Management for the Global Future, PICMET 2006* (2006).

[98] R. Schlegel et al. "On Spam over Internet Telephony (SPIT) Prevention". In: *Communications Magazine, IEEE* Volume: 46 , Issue: 8 (2008).

[99] M. D. Schoeder and J. H. Saltzer. "A hardware architecture for implementing protection rings". In: *ACM* (1972).

[100] H. Schulzrinne and S. Petrack. *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*. RFC 2833 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, May 2000. URL: http://www.ietf.org/rfc/rfc2833.txt.

[101] H. Schulzrinne et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550 (Standard). Updated by RFCs 5506, 5761, 6051, 6222 Accessed on 07 July 2012. Internet Engineering Task Force, July 2003. URL: http://www.ietf.org/rfc/rfc3550.txt.

[102] B. Schwarz. *Asterisk open-source PBX system*. 2004.

[103] Z. Seils and J. Christner. *Deploying Cisco Wide Area Application Services*. Cisco Press, 2008.

[104] L. Shan and N. Jiang. "Research on Security Mechanisms of SIP-based VoIP System". In: *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on, Volume: 2* (2009).

[105] IEEE Computer Society. *Port-Based Network Access Control*. IEEE Standard for Local and metropolitan area networks. 2010.

[106] R. Sparks. "SIP: Basics and Beyond". In: *Queue, Volume 5 Issue 2* (2007).

[107] M. Spencer et al. *IAX: Inter-Asterisk eXchange Version 2*. RFC 5456 (Informational). Accessed on 07 July 2012. Internet Engineering Task Force, Feb. 2010. URL: http://www.ietf.org/rfc/rfc5456.txt.

[108] A. Stango, N.R. Prasad, and D.M. Kyriazanos. "Threat Analysis Methodology For Security Evalutaion And Enhancement Planning". In: *IEEE Xplore* Emerging Security Information, Systems and Technologies, Third International Conference on (2009).

[109] J. Steven and G. Peterson. "Security lessons learned from Société Générale". In: *IEEE Security and Privacy* (2008).

[110] J. M. Stewart, M. Chapple, and D. Gibson. *CISSP: Certified Information Systems Security Professional Study Guide*. John Wiley & Sons, Inc., 2012.

[111] F. Swiderski and W. Snyder. *Threat Modeling*. Microsoft Press, 2004.

[112] S. Tangwongsan and S. Kassuvan. "A highly effective model for security protection against eavesdropping exploits". In: *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (2009).

[113] Tenable Network Security, Inc. *Tenable Network Security - Nessus 5*. http://www.tenable.com/products/nessus. Accessed on 06 July 2012. 2012.

[114] L. B. Torvalds. *Free minix-like kernel sources for 386-AT*. http://groups.google.com/group/comp.os.minix/msg/2194d253268b0a1b?pli=1. Accessed on 07 July 2012. 1991.

[115] R. Tracy. *LPIC-1/CompTIA Linux+ Certification*. 2012.

[116] T. Tsai, K. Vaidyanathan, and K. Gross. "Low-Overhead Run-Time Memory Leak Detection and Recovery". In: *Dependable Computing, 2006. PRDC '06. 12th Pacific Rim International Symposium on* (2006).

[117] H. Tsunoda et al. "A simple response packet confirmation method for DRDoS detection". In: *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference* (2006).

[118] International Telecommunication Union. *Information Technology - Open Systems interconnection - The Directory: Public-key and attribute certificate frameworks*. X.509 (Recommendation). Accessed on 07 July 2012. International Telecommunication Union, Nov. 2008. URL: http://www.itu.int/rec/T-REC-X.509-200811-I/.

[119] International Telecommunication Union. *Infrastructure of audiovisual services – Systems and terminal equipment for audiovisual services*. H.323 (Recommendation). Accessed on 07 July 2012. International Telecommunication Union, Dec. 2009. URL: http://www.itu.int/rec/T-REC-H.323-200912-I/en.

[120] International Telecommunication Union. *Management framework for Open Source Interconnection (OSI) for CCITT applications*. X.700 (Recommendation). Accessed on 07 July 2012. International Telecommunication Union, Sept. 1992. URL: http://www.itu.int/rec/T-REC-X.700-199209-I/.

[121] International Telecommunication Union. *Open Systems Interconnection – Connection-mode protocol specifications*. X.224 (Recommendation). Accessed on 07 July 2012. International Telecommunication Union, Nov. 1995. URL: http://www.itu.int/rec/T-REC-X.224-199511-I/en/.

[122] International Telecommunication Union. *Security architecture for Open Systems Interconnection for CCITT applications*. X.800 (Recommendation). Accessed on 07 July 2012. International Telecommunication Union, Mar. 1991. URL: http://www.itu.int/rec/T-REC-X.800-199193-I/.

[123] VMWARE Inc. *About VMware (VMW), the Global Leader in Virtualization Solutions*. http://www.vmware.com/company/. Accessed on 07 July 2012. 2012.

[124] J.A. Wang et al. "Security Metrics for Software Systems". In: *Proceeding ACM-SE 47 Proceedings of the 47th Annual Southeast Regional Conference* (2009).

[125] M. Wiboonrat. "An Empirical Study on Data Center System Failure Diagnosis". In: *Internet Monitoring and Protection, 2008. ICIMP '08. The Third International Conference on* (2008).

[126] P. Wouters. "Designing a Safe Network Using Firewalls". In: *Linux Journal* (1995).

[127] T. Ylonen and C. Lonvick. *The Secure Shell (SSH) Authentication Protocol*. RFC 4252 (Proposed Standard). Accessed on 07 July 2012. Internet Engineering Task Force, Jan. 2006. URL: http://www.ietf.org/rfc/rfc4252.txt.

[128] H. Yu et al. "Cloud Computing and Security Challenges". In: *ACM-SE '12: Proceedings of the 50th Annual Southeast Regional Conference* (2012).