Thesis (UAS)

Information Technology

European Computer Science

2011

Karsten Brauer

AUTHENTICATION AND SECURITY ASPECTS

in an international multi-user network

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS (UAS) │ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | European Computer Science

17.05.2011 | 59 pages

Advisor:

Dr. Vesa Torvinen

Karsten Brauer

AUTHENTICATION AND SECURITY ASPECTS in an international multi-user network

Access control and authentication are elementary principles to ensure security in information systems. To achieve these mechanisms, the functional principles and use of directory services and AAA are investigated and analyzed. By means of the Lightweight Directory Access Protocol (LDAP), a centralized user management and access management is going to be evaluated and designed. In a further practical part, the deployment of a LDAP directory service in an international organization is outlined to implement and improve organization's security requirements.

Furthermore, the fundamentals of information security with facilitating best practices and techniques for successful security engineering are explained, including common basic goals in computer and network security like confidentiality, integrity availability and authenticity. Subsequently, access control models and techniques, as well as cryptographic principles and standards are also discussed. To top the subject off a brief plan of risk management and security policies is presented to achieve the best possible accurate protection and security of assets, infrastructure and information in a technology environment.

KEYWORDS:

AAA, Access Control, Accountability, Authentication, Authorization, Certificate, Cryptography, Directory Service, Encryption, Hash Function, Identification, Identity Management, Information Security, LDAP, Public Key Infrastructure, RADIUS, Risk Management, Security Policy, User Management, X.500

TABLE OF CONTENTS

FIGURES

TABLES

1 Introduction

During the past 20 years, information systems and telecommunication became more and more important if not even essential in a globalized and interconnected world. These new techniques brought many advantages for employers and employees, companies and clients, but also some often still underestimated risks.

1.1 Reasons and Motivation

Computer systems simplified, quickened and improved work and production processes in many ways. On the other hand, their interconnections with each other opened the doors for inquisitive or malicious attempts of gathering or even worse sabotage to harm or interrupt normal operations.

The IT systems of an organization beginning from the physical network cablings to end-user applications running on the servers need to be protected and reliably maintained.

1.2 Threats

Several basic threats and dangers like the following examples threaten an IT environment. In addition to it, the following chapters will outline an overview about information security principles and techniques, concerning the following threats of computer security.

Failure

Reliable working information and communication systems are vital in today's working life. The loss of network connectivity through network problems, hardware defects on either workstations or servers, or even more often software errors can lower productivity drastically.

Loss of data

Data is generic term for all kind of information, which includes files like pictures or documents, financial calculations or construction plans. The loss of data could happen accidentally for example by human error or a wrong configuration, in the worst case by a malicious attack from inside or outside the network.

Theft of information

Innovative companies run a higher risk of becoming a victim of economic crimes. Possible opponents may try to get to know company secrets, which could include latest and future developments, economic details or information about production processes. Eavesdroppers and intruders may threaten from inside or outside while listening on a company's communication or stealing valuable information on flash drives, or from distance through the internet.

1.3 Aims and Goals

Some threats can be easily minimized, for example by establishing a security policy or adequate risk management. However, a main part of this thesis is going to examine the threats of information and identity theft, with the focus on access control and authentication.

After exploring the principles and best practices of information security, a practical section deals with the question of a centralized user management for an international organization. Using the example of LDAP, directory services will be investigated and explained. Furthermore, an LDAP directory service will be deployed to manage users between multiple sites and services.

2 Information Security

> Preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can also be involved (ISO/IEC 27002, 2005)

Information security signifies the protection of information and information systems from unauthorized access, modification or destruction.

Computer security and information assurance are both sub-areas of information security, whereas computer security defines practices and procedures how to protect information on computer systems and networks from theft, corruption or natural disaster as already mentioned in the beginning. Information assurance, however, is an approach of managing risks related to the use, processing, storage, and transmission of information or data.

The three main components and goals of those interrelated fields are to protect and ensure the confidentiality, integrity, and availability of information or data.

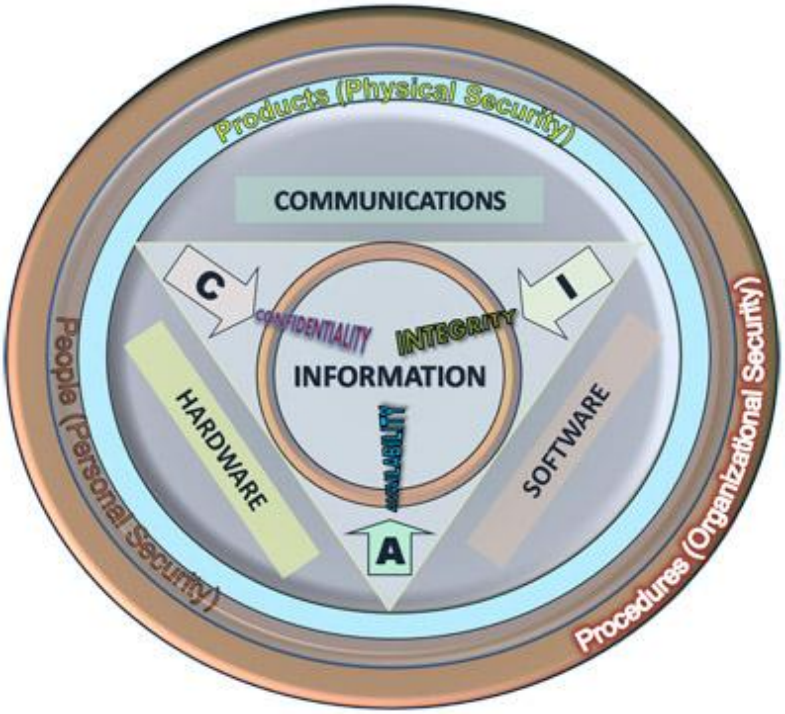

Figure 2.1. CIA triad - Information Security Components[1]

[1] John Manuel, The Information Security triad: CIA., 2009-12-26; (Wikipedia, 2011)

The CIA triad (confidentiality, integrity, availability) has represented the key principles of information security for many years, although there is a continuous debate of extending these three points. Security experts have agreed on a few additions. On the other hand, other points have been denied and some points do not fit well with the CIA core concept.

An alternative model the “Parkerian hexad” (Parker, 2002) has been proposed at the beginning of the new century. Nevertheless, the six atomic elements of information, named confidentiality, possession, integrity, authenticity, availability, and utility, are still the subject of debate amongst security professionals.

The following sections explain the generally accepted key concepts of information security.

2.1 Confidentiality

> Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes (ISO/IEC 27001, 2005)

Confidential information must only be accessed, used, copied, or disclosed by authorized users. A confidentiality breach occurs if unauthorized persons or systems access or disclose information they are not allowed. To prevent disclosure of confidential data like a credit card number from eavesdroppers, the transmission must be encrypted. In addition, the number must be protected wherever it will be processed or stored (e.g., databases) to prevent unauthorized access.

2.2 Integrity

> Integrity is the property of safeguarding the accuracy and completeness of assets (ISO/IEC 27001, 2005)

In information security, integrity means that information cannot be altered or tampered without being detected. It ensures the correctness of a message and protects against unauthorized modification. If information has been changed, the hash value of a file or the message authentication code (MAC) of a message would change, too. Thus, a modification would be recognized when comparing the current against the original information.

2.3 Availability

> Availability is the property of being accessible and usable upon demand by an authorized entity (ISO/IEC 27001, 2005)

Availability assumes that information systems and services, as well as the information itself, is available and operating as expected when needed or requested. It could be also considered as the degree to which a system or equipment is operable.

2.4 Authenticity

> Authenticity is the property that an entity is what it claims to be (ISO/IEC 27000, 2009)

Authenticity proves that all parties involved in an action are who they claim to be by validating their identities. In information security, Message Authentication Codes (MAC) or digital signatures are used to ensure the authenticity of data, transactions, communications or, documents, i.e., that the information is genuine and authentic.

2.5 Non-repudiation

> Non-repudiation is the ability to prove the occurrence of a claimed event or action and its originating entities, in order to resolve disputes about the occurrence or non-occurrence of the event or action and involvement of entities in the event (ISO/IEC 27000, 2009)

In information technology and communications, non-repudiation assures that a sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data. In electronic commerce, digital signatures are used to establish authenticity and non-repudiation. (Wikipedia, 2011)

3 Access Control

> Access Control means to ensure that access to assets is authorized and restricted based on business and security requirements (ISO/IEC 27000, 2009)

Access control polices and regulates access to systems, information or data. In most cases, access must be always restricted to individuals or computer systems that are authorized to access. Therefore, it usually follows the stages of identification, authentication and authorization to control access considering privileges. A superior process of accountability can accomplish the responsibility of an entity for its actions, for example, by providing a log.

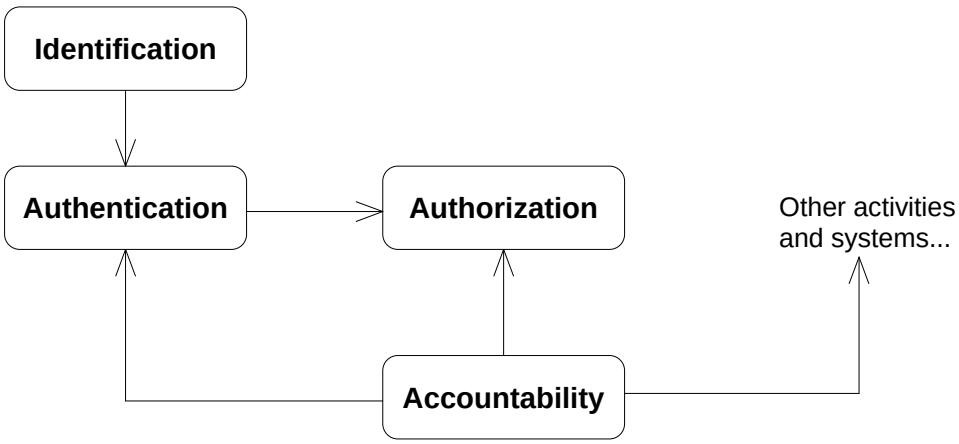

Figure 3.1. Access control steps

3.1 Identification

Identification is the action of identifying or recognizing a person or system. During the identification, an identity will be claimed which may or may not be true. Usually, the subject will provide a public piece of information, like a username or an identification number.

$$Sender_{Bob} \rightarrow Hello, I\ am\ Bob.$$

$$Sender_{Alan} \rightarrow Hi\ Bob, I\ am\ Charlie.$$

Figure 3.2. True and false identification

3.2 Authentication

The International Organization for Standardization (ISO) defines authentication as the "provision of assurance that a claimed characteristic of an entity is correct" (ISO/IEC 27000, 2009). In other words, authentication verifies a claimed identity and proves that an individual or computer system is who or what it claims to be.

$$Sender_{Bob} \rightarrow I\ am\ Bob.\ Here\ is\ my\ ID\ card.$$

$$Sender_{Alan} \rightarrow Hi, I\ am\ Alan.\ My\ PIN\ is\ 1234.$$

Figure 3.3. Authentication with ownership and knowledge

In information security, the identification of a user usually insists of a username (public) and a password (private information). With the username, an identity will be claimed; the password will be matched against a stored user password, to verify the user's identity. If the username and passwords correspond, the user is authenticated.

Further options for identification and authentication are biometric information, such as a fingerprint or electronic systems like RFID tokens or smart cards. Different identification methods distinguish from each other in effort, reliability and security. A combination of various methods (multi-factor authentication) may increase security and reduce risk of identity theft; for example, the loss of a RFID token could open all doors, however, an additional PIN code (Personal Identification Number) would still prevent unauthorized access (two-factor authentication).

The three general characteristics (i.e., factors), used to prove an identity, are as follows (Harris, 2002)

1. Something the user owns / has (such as a token or smart card)
2. Something the user knows (a passphrase or PIN)
3. Something only the user can present (e.g. biometric identification).

An additional characteristic can be (Bishop, 2004)

4. Where the user is (for example using a particular terminal or workstation).

Basically, all identification methods compare the entered or read data with the stored samples to ensure that a user is who he claims to be. As a result, data has to be exchanged with databases or directories where communication and storage is also to be considered confidential and must be protected.

3.3 Authorization

Authorization is the process that determines and approves the privileges for an authenticated access (Bhattacharya, Sandip; et al., 2003). Moreover, it defines what information an identified and authenticated person or system is permitted to access and which actions he or it is allowed to perform.

3.3.1 Access Control Models

Access control models are used to enforce the rules and objectives of an established security policy and to define how subjects can access objects. The three most common access control models used today are explained briefly below.[2] (Vacca, 2009)

The current less convenient and popular Discretionary Access Control (DAC) allows the owner or creator of an object, for example, resources like a file, to define who is or who is not allowed to access an object. That is why DAC is sometimes also called identity-based access control (IBAC).

Mandatory Access Control (MAC), however, uses classifications to determine what the subject (user) needs to know. The subject will be able to access all objects (data or information) where its clearance level is higher or equal than the object's classification. It is occasionally referred to as a rule-based access control.

The most widespread model, Role-Based Access Control (RBAC), uses roles or groups to assign permissions to a subject. A user will be able to access the resources his group(s) or role(s) are allowed to. For example, an administrator could create a job position or department-related permissions as a group and assign the associated employees to this group. As a benefit, administrative effort decreases, because mostly only the role and not the users themselves need to be modified.

All access control models can be also used together or in combination to implement an organization's security requirements. (Harris, 2002)

[2] Caballero, Albert; Information Security Essentials for IT Managers: Protecting Mission-Critical Systems

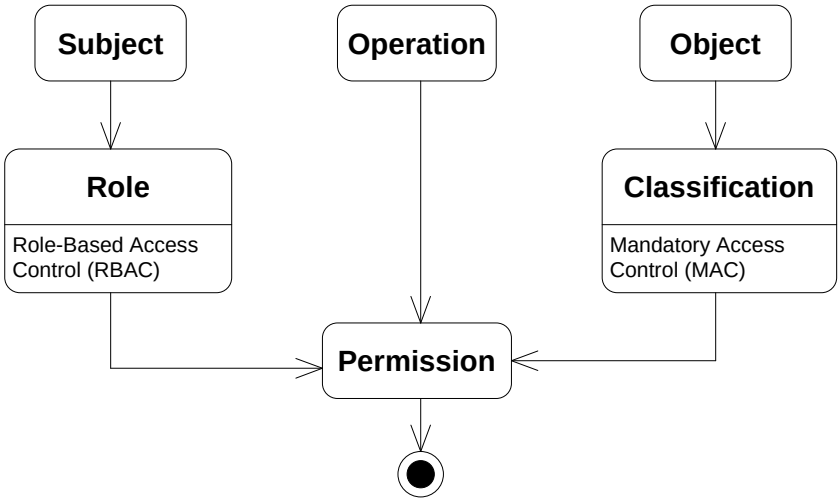

Figure 3.4. Access Control Models and Techniques

In general, permissions depend on the subject – for example, a user; the object - the resource the user wants to access, and an operation, for example, a user wants to rename a file. According to the access control model(s) used other aspects also need to be considered, such as the role of the user (RBAC), or the classification of an object (MAC). The intersection of these parts reveals the permissions and privileges of an access control policy.

3.3.2 Access Control Techniques

The *access control matrix* is a mechanism to associate access permissions of a subject to an object. It is one of the more frequently used techniques in terms of access control. The rows are constituted of the user's *capability table*; on the other hand, the columns reflect the resource's *Access Control List* (ACL). An Access Control List is a method of determining the user's individual access rights and privileges to resources, like files or folders, on a system. Common privileges in an operating system and file system context are (Gattiker, 2004):

- Read – to read a file or the content of a directory
- Write – to create or update files / directories
- Execute – to run a file, for example a program

Table 3.1. Example of an Access Control Matrix

User / Role	File Server	Repository
Admin	full control	full control
Accounting	read, write, execute	no access
Team leader	read, write, execute	read, write
Programmer	read, execute	read

Content-Dependent Access Control is another access control technique used. Access will be controlled depending on the content which means that different access levels, with increasing permissions, exist. For example, the receptionist at a bank is able to see the client's name and his account number, whereas a bank employee can also look up the current account balance. However, the bank manager is also able to take a look at the bank statements of the last year. This is a widely used approach in institutions or organizations with a need of a certain degree of confidentiality, for example, medical records or personnel files.

Some further common control techniques among many others depend on the

- Time of day (allow access only at specified times, from - until)
- Transaction type (what operation is allowed)
- Logical location (what IP address)
- Physical location (which terminal).

3.4 Accountability

Accountability ascertains the responsibility of an entity (like a person) for its actions and decisions (ISO/IEC 27000, 2009). For this purpose, all relevant activities events and operations on a system, e.g., failed and successful authentication attempts, are recorded in a log. An audit trail, also referred as information audit, is a chronological record of system activities to enable the reconstruction and examination of a sequence of events.

4 Cryptography

Cryptography most times refers to encryption, the process of converting plain information (plaintext) into unintelligible ciphertext, i.e., to encrypted information. However, cryptography covers a broader range of useful methods and functions today.

4.1 Hash Function

A hash function is a deterministic procedure to prove the integrity of data, i.e., that a file or message has not been altered or corrupted. A hash word reduces an amount of data, like a file, to a given length of bits through calculation of cryptographic hash algorithms, whereas a good algorithm should create a unique hash value. This presumes that a hash function is resistant to collisions. A collision arises if different input data result in the same hash value. In addition, a hash function is “one way” only; this means that it is almost impossible to derive the original data or message from a given hash.

$$MD5(\text{"}Hello\ World!\text{"}) \Rightarrow \text{ed076287532e86365e841e92bfc50d8c}$$

$$SHA1(\text{"}Hello\ World!\text{"}) \Rightarrow \text{2ef7bde608ce5404e97d5f042f95f89f1c232871}$$

Figure 4.1. Example of MD5 and SHA-1 hash function

Hashes are widely used in information security to prove the integrity of information, for example to verify the completeness of a data backup or to ensure that a software program has not been manipulated. They are also an important element in digital signatures and certificates as can be read in the following sections.

Today’s most common hash functions are the Secure Hash Algorithms (SHA-1[3] and SHA-2[4]) by the NIST[5] and NSA[6]. By the time of writing, the NIST holds a competition of

[3] FIPS 180-1: Secure Hash Standard (SHS) [1995]
[4] FIPS 180-2: Secure Hash Standard (SHS) [2002]
[5] National Institute of Standards and Technology (NIST)

hash algorithms, to elect a succeeding SHA-3 hash standard in 2012. This new hash algorithm will eliminate several known security flaws existing in the current algorithms, which might be used generating collisions[7]. They might be, therefore, considered insecure in the near future.

The following table shows recommendations of hash algorithms to be used in digital signatures and their transitions by the German *Federal Network Agency* (Bundesnetzagentur, 2010) and the NIST (NIST, 2011). The recommendations are in relation with the hash algorithm's (expected) security strength.

Table 4.1. Hash function transitions for digital signatures

Hash Algorithm	Length in Bits	Recommendations *Bundesnetzagentur / NIST*
MD5[8]	128	Disallowed
SHA-1	160	Acceptable through 2008 / 2010
RIPEMD-160	160	Acceptable through 2010 / not a FIPS
SHA-224 (SHA-2)	224	Acceptable through 2015 / Acceptable
SHA-256 (SHA-2)	256	Acceptable through 2017 / Acceptable
SHA-384 (SHA-2)	384	Acceptable through 2017 / Acceptable
SHA-512 (SHA-2)	512	Acceptable through 2017 / Acceptable

4.1.1 Key Derivation Functions

Modern computer chips and cloud computing made it easier and easier to calculate hash values in advance, which are stored in databases known as *rainbow tables*. These also purchasable huge password collections allow attackers to work out the corresponding password of a hash value; more precisely to find a collision having the same hash result.

If passwords are stored "as encrypted hash values, attackers could [..] use brute force to try and derive the original passwords. Using a modern graphics card, the time

[6] National Security Agency (NSA)
[7] Collision attack against SHA-1 with a theoretical complexity of 2^51 hash function calls (Manuel, 2008)
[8] RFC 1321: The MD5 Message-Digest Algorithm [1992]

required to crack a six character password is only 9 minutes. For eight characters, the required computing time is already 300 days. However, this time can be reduced by hiring cloud servers to crack the passwords. For example, using Amazon's Elastic Computing Cloud (EC2) to crack an eight-character password with brute force would cost about 600 Euros. With twelve characters, it would already cost more than 15 billion Euros. This means that passwords of 11+ digits can currently be regarded as safe, as the cost required to crack them would be greater than a criminal's potential earnings."[9] (Heise Media UK Ltd., 2011)

$$Hash(Password + Salt) \Rightarrow \text{HashWord}$$

$$MD5(\text{"}Hello\ World!\ SALT\text{"}) \Rightarrow \text{a62b8d5fb2a99a88eb8e8b31ba90845c}$$

Figure 4.2. Hash function with salt

An efficient way to increase effort and expenses is the use / implementation of salts. A *salt* has the purpose of producing a large set of keys corresponding to a given password. Therefore, cryptanalysis and statistical processes will not be able to find out frequently used weak passwords easily. For example, an individual random salt[10] will be created or calculated for each password stored in a database. The result is a linear intensification of computing effort, which finally requires more computing time or higher financial means for better equipment.

Even better are *key derivation functions*, which produce a derived key from a base key and parameters like salt value and iteration count. A well known example is PBKDF2 (Password-Based Key Derivation Function) that applies a pseudorandom function, such as a cryptographic hash, cipher, or HMAC to an input password along with a salt and repeats the derivation process many times (in general 10.000 to 100.000 times) to produce a derived key. The additional computational costs make password cracking much more difficult, and are known as *key stretching*. (Kaliski, 2000)

[9] Attack and intrusion of Sony's PlayStation Network (PSN) with data piracy of 77 million users in April 2011
[10] A salt length of at least 64 bits is recommended

4.2 Encryption

Encryption, sometimes also referred as encoding, is the process of converting cleartext into unreadable ciphertext. Its history goes back thousands of years wherever information needed to be hidden and protected. Even more today, encryption is needed and used by governments, military, and enterprises to keep secret information confidential. Though encryption can protect the confidentiality of messages, other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a Message Authentication Code (MAC) or a digital signature.

4.2.1 Symmetric Encryption

The most common form of cryptography is symmetric encryption. The same key or secret is used to encrypt and decrypt data.

$$Data_{enc} \coloneqq Encrypt(Data, Key)$$
$$Data \coloneqq Decrypt(Data_{enc}, Key)$$

Figure 4.3. Symmetric encryption with shared key

The advantages of symmetric encryption are its ease (for example, implementation or requirements) and performance. However, the key, e.g., a password, must be known by all parties involved, so it may need to be shared, for example by e-mail. More secure and often used are specific key exchange methods like Diffie-Hellman or a Public Key Infrastructure (PKI).

A simple historic example of symmetric encryption is the algorithm ROT13, which rotates the letters of the alphabet 13 places to the left; for example, “Hello World!” will be encrypted to “Uryyb Jbeyq!”. If another person wants to read the examined secure message, he has to move 13 letters to the right again. This knowledge of how to encrypt and decrypt the data is called a secret. Nowadays, this is considered as insecure since modern cryptanalysis and (network) computing performance are able to break much harder ciphers within seconds.

Simplified, it can be separated between two types or approaches of symmetric encryption, between stream and block algorithms. Stream ciphers encrypt symbol by symbol, whereas block ciphers encrypt a block, i.e., a group of symbols by time. The advantages of the one are the disadvantages of the other and vice versa. Block ciphers are more secure, for example, because of their higher diffusion. On the other side, stream ciphers may be less prone to errors and might be faster.

However, most modern encryption methods are nowadays block algorithms (for security reasons) and use keys like passphrases to calculate their ciphertexts. The table below lists a few chosen popular encryption algorithms and their characteristics.

Table 4.2. Symmetric encryption algorithms

Algorithm	Block	Key Length	Remarks
DES	64	56 (effective)	Considered insecure
3DES	64	56, 112, 168	DES successor, three rounds of DES
AES	128	128, 192, 256	Developed as Rijndael
Blowfish	64	32 - 448	Designed by Bruce Schneier
Twofish	128	128, 192, 256	Successor of Blowfish, AES finalist
Serpent	128	128, 192, 256	Finalist in AES contest (2nd place)
IDEA[11]	64	128	Intended replacement for DES
RC4	*Stream*	40 - 2048	Used in SSL and WEP

The best-known and common encryption algorithms have their motivations in the protection of United States government's computer and information security. As a consequence, they have been also standardized by American authorities, first of all the *National Institute of Standards and Technology (NIST)*, former *National Bureau of Standards*, which announces *Federal Information Processing Standards (FIPS)*. These standards are to be used in computer systems of all non-military U.S. government agencies and their contractors. In the following sections, the three major standards are briefly described.

[11] International Data Encryption Algorithm [1991]

Data Encryption Standard (DES)

The Data Encryption Standard was firstly standardized in 1976 by the work of IBM and NSA, reasoned by the need of a governmental standard for encrypting confidential information. The first standard was published as *FIBU PUB 46* in 1977. DES is a shared secret block cipher with a shortened key length of 56 bits and a block size of 64 bits. Because of its short key length and advanced techniques of cryptanalysis, it is considered insecure today.

Triple DES (3DES)

The DES successor, Triple Data Encryption Algorithm (TDEA), was published as *FIPS 46-3* in 1989. It uses the DES algorithm three times for each data block, usually by encryption and decryption with three different keys, as shown in the following example.

$$ciphertext = Encrypt(k3, Decrypt(k2, Encrypt(k1, plaintext)))$$

Figure 4.4. Triple Data Encryption Algorithm (TDEA)

The NIST designates TDEA to have only 80 (2TDEA)[12] to 112 (3TDEA) bits of security, depending on the independent generation of *k3*. (NIST, 2007 p. 61) In comparison, AES is assessed, in relation to the key length, at a security strength of at least 128 bits.[13]

Advanced Encryption Standard (AES)

AES is today's standard for secure governmental and organizational encryption of confidential (unclassified) information. AES is based on the "Rijndael" algorithm of the two Belgian cryptographers Vincent Rijmen and Joan Daemen, which has been chosen in a 5 years standardization process and contest of 15 competing algorithms by the NIST. Its advantages are the still unbroken high security and good performance, which allows software, as well as hardware implementations. In late 2001, AES was finally published as *FIPS PUB 197* and is still state-of-the-art in computer cryptography.

[12] Assuming that an attacker has access to approximately 2^40 (plaintext, ciphertext) pairs, using the same secret key
[13] Compare NIST Special Publication 800-131A; (NIST, 2011)

4.3 Public Key Cryptography

One of the most important fields in cryptography is the public key cryptography, which relies not just on a single key, but rather on separated keys for encoding and decoding. Such keys are commonly called *public*, available for everyone, and *private* which is only known by the creator of the key pair.

A public key cryptosystem must meet the following conditions (Bishop, 2004):

1. Easy to encipher or decipher a message with the appropriate key given
2. Infeasible to derive the private key from the public key
3. Infeasible to determine the private key from a chosen plaintext attack.

These conditions generally assure that an encrypted information can only be deciphered using the appropriate private key.

There are three commonly used public key algorithms today. *Diffie-Hellmann (DH)*, named after its publishers Whitfield Diffie and Martin Hellman [1976] is a protocol designed for key exchange, for example, to transfer passwords securely over a public or shared medium, like the Internet. The *Digital Signature Algorithm (DSA)*[14], as its name implies, is used for digital signatures, whereas *RSA*[15] is applicable for digital signatures and encryption purposes.

Table 4.3. Public key algorithms

Algorithm	Remarks and based Problems
DH	*Finite Field Cryptography (FFC)*: Diffie-Hellman problem (discrete logarithm problem) Key exchange protocol
DSA	*Finite Field Cryptography (FFC)*: discrete logarithm problem For signatures only (sign - verify)
RSA	*Integer Factorization Cryptography (IFC)*: large number factoring problem (RSA inversion) Cryptosystem for encryption and authentication

[14] FIPS 186: Digital Signature Standard (DSS) [1994]
[15] Rives, Shamir, Adleman [1977]

All these asymmetric algorithms rely on certain mathematical problems[16], like discrete logarithm problems or the problem of integer factorization. The newest protocols and standards also support *Elliptic Curve Cryptography (ECC)*, which assumes that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is unfeasible. The size of the elliptic curve determines the difficulty of the problem. It is accepted that the same level of security could be reached with a much smaller elliptic curve group compared to standard RSA which results in smaller key sizes and reduced storage and transmission requirements.[17] (Wikipedia, 2011)

4.3.1 Asymmetric Encryption

Another form of encryption is public key encryption, also known as asymmetric encryption. The first name refers to its essential component, the public key which is used for the encryption of information or data. On the other side, a 'private key' only known to the receiver is used for the decryption process. Similarly to the general conditions for public key cryptography, the following fundamental rules apply.

- The sender knows the encrypting key of the receiver (public key)
- The decrypting key (private key) must not be derived from the encrypting key

The following figure illustrates the encryption process, using a public key, and the deciphering using the associated private key.

$$Sender_B \rightarrow Encrypt(Message, Key_{pub^A})$$

$$Receiver_A \leftarrow Decrypt(Message, Key_{priv^A})$$

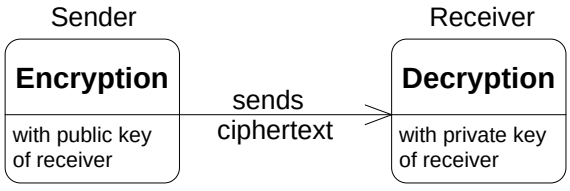

Figure 4.5. Asymmetric encryption

[16] Number-Theoretic Reference Problems cf. (Menezes, et al., 2001)
[17] FFC and IFC key size 2048 bit, ECC equivalent 224 bit; security strength of 112 bit (NIST, 2007)

Today, RSA is the most widely known and used cryptosystem offering asymmetric encryption as well as signing capabilities. However, public key encryption or decryption is substantially slower than common symmetric encryption. Therefore, in practice, public key encryption is most times only used for the transport of symmetric encryption algorithm keys or passwords. This is sometimes also referred to as *hybrid encryption*, because of the asymmetric key exchange and the symmetric encrypted payload or data transfer. Popular examples are IPSec, SSL/TLS or E-Mail encryption with PGP/GPG.

4.3.2 Signature

A digital signature ensures authenticity, integrity and non-repudiation of a message or a document, therefore, it is most likely used in combination with a hash function. In general, a hash value of a document will be created and encrypted with the senders own private key. The receiver decrypts the received signature using the sender's public key and verifies it against the local document hash word. If the values/signatures equal, the message has not been altered and is authentic. Besides, the sender has also been approved or respectively authenticated, because only he could know his 'secret' private key.

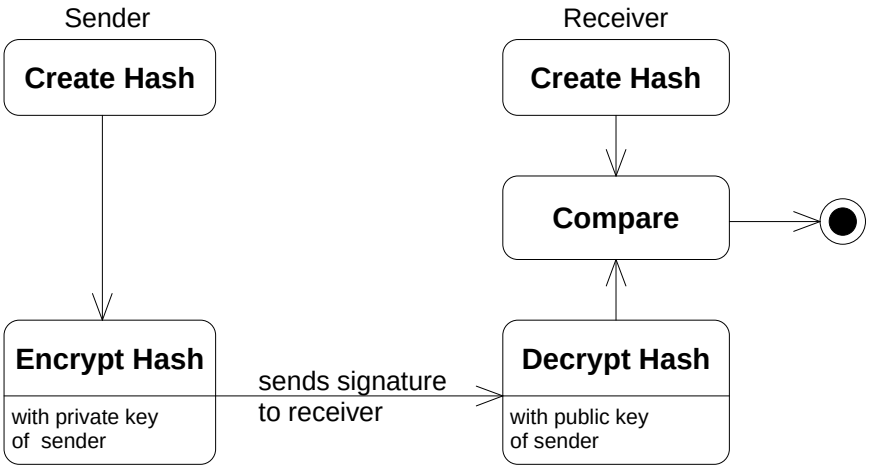

Figure 4.6. Digital signature of a message

Primarily, a digital signature must meet the following two conditions (Pfleeger, 2006):

1. Unforgeable – It must be impossible that the same signature for a message is produced by anyone else (protection of the signature).
2. Authenticity – The receiver can verify that the message has been signed by the sender and only he could have created the signature (protection of the receiver)

The key-user association, to verify a public key owner, is commonly done using a Private Key Infrastructure (PKI) which is described in greater detail in the next section. Another important, if not the most important, fact is that the private key must remain private, in other words, secret. The holder of a (stolen) private key would be able to produce every kind of signature in the bearer's name.

The most commonly used digital signature algorithms are DSA (and its elliptic curve variant ECDSA) or are based on RSA signature schemes.

4.3.3 Public Key Infrastructure (PKI)

A Public Key Infrastructure is the overall structure of processes, servers, services and involved persons needed to manage and maintain digital certificates.

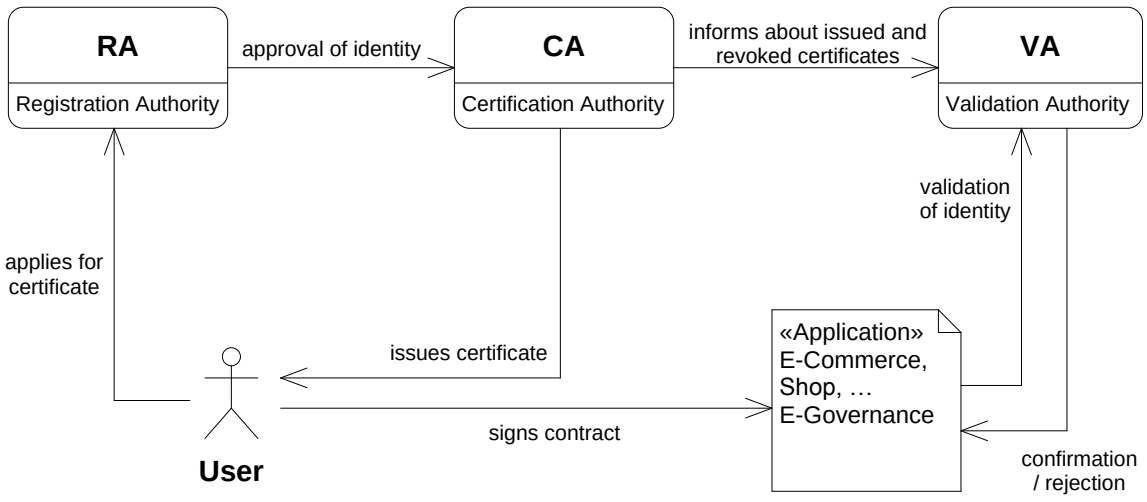

Figure 4.7. Concept of a Public Key Infrastructure

Digital certificates are an up-to-date still common way of managing trust in the variety and anonymity of the Internet. *Certification Authorities (CA)* issue certificates for clients (for example, e-mail addresses) or servers, like SSL encrypted web pages. The common approach of CAs is to check the applicants identity information submitted with the *Certificate Signing Request (CSR)*. Vendors offer several classes of identity validation, which can include the review of commercial registers or as simple as an e-mail address, to which an approval mail will be sent. However, an inexperienced Internet user would not be able to see any difference, since all classes of certificates offer the same shallow security and vendor browser seals.

On the technical site, the certification procedure follows the following simplified steps:

1. Applicant creates private and public key pair.
2. Applicant applies for a certificate (Certificate Signing Request) with applicant's public key and other contents[18].
3. Certificate Authority issues certificate (signs certificate after approval of identity)

The validation of certificates takes place through the certificate signature by the CA, which can be proved with the certification authority's public key. These public keys of best-known root certification authorities are commonly stored and included in the operating system or clients, such as web browsers. If a certificate is issued and signed by an intermediate certification authority the whole certification tree must be validated until a trusted (root) CA is found. Other mechanisms are *Certification Revocation Lists (CRL)*, where revoked and, therefore, invalid certificates are listed[19], as well as the verification of the validity dates[20].

Nevertheless, public key infrastructures as well as *web of trust* approaches are controversy discussed, not least because of several gaps in their trust model. On the other hand, commercial vendors market their certificates at quite high costs considering that they have little effort and the fact that an alternative identity validation is not available for today's clients. This leads to the enforcement of SSL certificates by security audits and the business own needs, avoiding security alerts in a potential customer's web browser, if using self-signed certificates.

[18] Common contents of X.509 certificates: serial number, issuer, validity, subject (+public key), certificate signature, etc.
[19] Certificate revocation with the aid of their serial number and maintained CRL's at the CA.
[20] The date the certificate is first valid from and the expiration date.

The weaknesses and complexity of the system have recently been demonstrated by a certificate theft at Comodo[21]. A 21-years old programmer claims to be behind the unauthorized creation of illegitimate SSL certificates for the web servers of various major web service providers.[22] He accessed to the certification server's API after decompiling a software library at a reseller's website used for the submission of certificate signing requests. In the sources, the hacker finally found access credentials for the reseller's Comodo and GeoTrust accounts. (Heise Media UK Ltd., 2011)

In the light of these points, alternatives are wanted. A future success could be the Domain Name System Security Extensions (DNSSEC)[23], an extension of the DNS protocol that allows authenticity and integrity of DNS transactions. A working group (DANE[24]) has been already convoked to figure out how to put SSL certificates into DNSSEC.

[21] Comodo Group, Inc. - Comodo is Creating Trust Online®; http://www.comodo.com

[22] An attacker generated forged certificates for login.live.com, mail.google.com, www.google.com, login.yahoo.com, login.skype.com and addons.mozilla.org. [2011]

[23] RFC 4033: DNS Security Introduction and Requirements [2005]

[24] Using Secure DNS to associate Certificates with Domain Names for TLS; http://www.ietf.org/id/draft-ietf-dane-protocol-06.txt [2011]

5 Risk Management at a Glance

> Coordinated activities to direct and control an organization with regard to risk; Risk management generally includes risk assessment, risk treatment, risk acceptance, risk communication, risk monitoring and risk review. (ISO/IEC 27000, 2009)

Risk management has a major impact on modern Information Technology security, as well as many other disciplines. It usually begins with the identification and classification of the information assets that need to be protected. In the next step, risk assessment examines the probability (threats and vulnerabilities) and impact (costs) of undesired events.

Risk treatment generally consists of countermeasures like, firewalls and anti-virus software, or policies and procedures such as regular backups and configuration hardening. In addition, training such as security awareness education is a preventive solution. The cost and benefit of each countermeasure is carefully considered. Thus, the aim of risk management is not only to eliminate all risks, but also to manage them in the most cost-effective way.

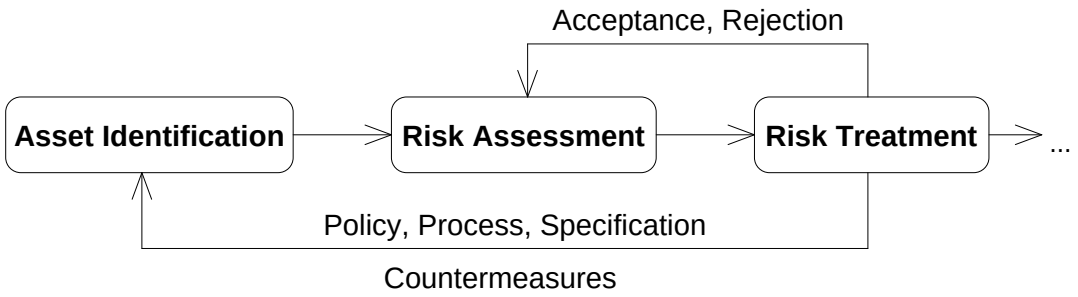

Figure 5.1. Taxonomy of Risk Management

Risk assessment and risk management plans are continuously revised and improved based on data experienced from regular tests and evaluation.

5.1 Security Policy

An effective and important part in an organization's computer and data security are policies. Policies provide a formal doctrine of guiding principles for all Information Technology-related activities. They are an operational framework of best practices, defined technology standards and governance practices for all entities of an organization and should be reviewed, revised, and approved on a regular, e.g., annual basis. In other words, policies provide the user or an employee with rules and guidelines for handling assets or data, and even more importantly, regulations for what they are not allowed to do. Therefore, policies form a legal guideline and may make users liable for bad or wrong behavior.

There are many different subjects and *policies*, which can be joint or separate documents. The following are a few common policy examples:

Change Management, Data Back Up, Data Security, Disaster Recovery, Information Classification, Internet Security, Network Security, Operations Activity, Passwords and Data Privacy, Purchasing, Remote Network Access, Security Audit, Server Security, Software Development Life Cycle and many others.

The following example of a *password policy* regulates how passwords are used and computer security is established within an organization.

- Passwords will be composed of at least two of these symbol subsets, and will have a length of not less than 8 characters: Upper- and lower-case alpha, numeric digits, punctuation and special characters.
- User passwords must not contain the user's name or ID.
- Generic accounts and group passwords are not allowed so that individual accountability can be maintained at all times.
- Passwords cannot be re-used for a minimum period of 1 year.
- Incorrect password attempts allowed before password is suspended is 4.
- All user-level passwords (e.g., email, web, desktop computer, etc.) must be changed at least every 90 days. The recommended change interval is monthly.
- Password-protected screensavers should automatically activate within 10 minutes of system idle.
- Passwords must not be inserted into e-mail messages or other forms of electronic communication that leave the network/infrastructure in an unencrypted channel.

As another example, a *data security policy* regulates how sufficient data security can be established. The usage of external storages as well as the connection of any kind of devices to computers or networks might be prohibited. Another often-unpracticed point is how to deal with old or defective storage devices, like hard disk drives. Hard disks should normally be erased and overwritten following defined standards.[25] However, flash drives and solid state hard drives (SSD) may be still readable using these common procedures. In these cases, special sanitization processes may need to be used. (Wei, et al., 2011)

5.2 Disaster Recovery

A disaster recovery plan or disaster recovery policy ensures that the technology environment operates and performs at normal after a natural or human-induced crisis or disaster.

Recovery goals in a disaster are commonly to minimize disruption of critical business functions, to maintain overall management functions, and to maintain the security and integrity of assets, data, and infrastructure.

A disaster recovery plan goes hand in hand with risk management and risk reduction, whereas a definition of a disaster must be declared, for example, as an event that significantly reduces the availability and operability of Information Technology services for more than twenty-four hours. In general, critical systems and data must be protected within reasonable and financial means while understanding that systems and data are not equally critical for everyone. One important measure to safeguard data is a *backup plan*, which describes the process, storing and verification of data backups. Another preventive countermeasure is the introduction of *High Availability* systems (HA) that are set up in a redundant or replicated manner for continuous access and operation in case of failure.

[25] By the time of writing most US GO's and military suggest device destruction instead of purging/sanitizing

Very simple and efficient measures to prevent system and hardware failures are

- Redundant Array of Independent Disks (RAID), e.g., RAID-1 mirroring protects in case of an single hard drive failure
- Redundant power supply and Uninterruptible power supply (UPS)
- Data backups - frequent backups allow to restore lost or damaged data
- External backups - off-site backups protect data in case of disasters; for example, copy on tape, DVD, USB drives or using network replication.

Backups and especially external off-site data storage, however, challenge again the question of data security. Therefore, in practice, at least encryption and access control measures must be taken into consideration.

6 An International Organization

The needs of organizations vary depending on size, structure, workflow, and, of course, their security requirements. In small to medium enterprises, IT security is still under-valued, even if contracts and agreements declare high contractual penalties. The threats are often unseen, benefited by a trustful or familial work atmosphere, or worse that persons are not responsible.

The supporting enterprise in this project is a successful small company specializing in automation technology as well as hardware and software development. Due to its growth and continuing competition, a branch in Bangkok, Thailand has been established, which is independent but a major factor in daily business operation and project development.

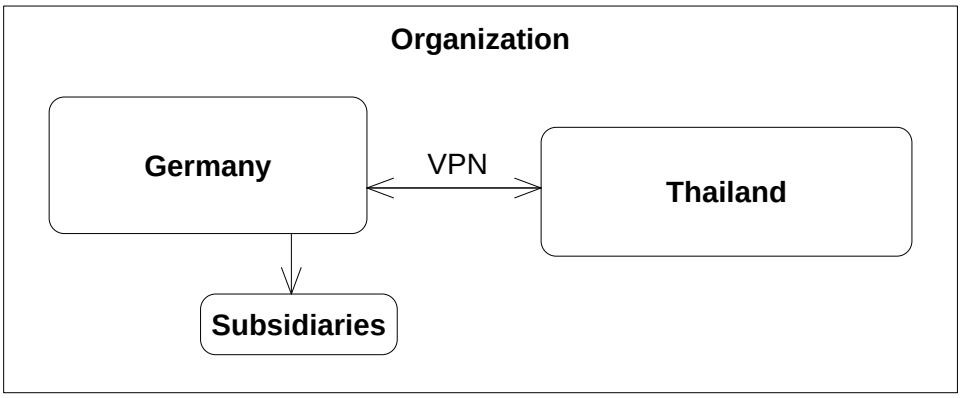

Figure 6.1. Organizational structure of an international company

6.1 IT Situation

At the time of writing, user administration and management became more and more time consuming and difficult. For a growing successful company, new employees and users still had to be entered into several systems manually. Each step was susceptible to mistakes and administrative competences were not clearly arranged.

The figure below shows an incomplete overview of widely used services within the whole enterprise. Adventitiously, some of these services and servers exist at both branches; others however, are maintained by one site only, but are also accessible through the company's virtual private network (VPN).

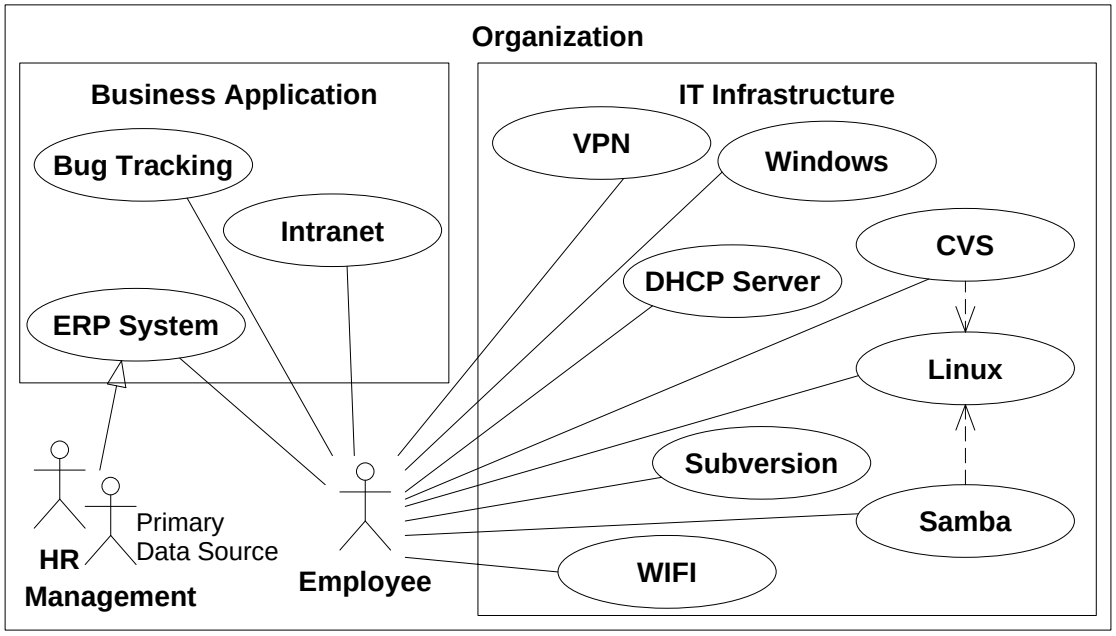

Figure 6.2. Overview of systems and services

The disadvantages and inefficiency of the current handling can be easily seen. Each system and service requires own information, user accounts and passwords. Maintaining this system(s) will be a challenge for further growth.

ERP System

However, there is one single centralized point where employee and user information is maintained consistently. The proprietary enterprise resource planning and project management system contains all employees as well as product and project information that are necessary to use this data source for a centralized user database and management.

The simplified core functions of this management system are outlined in the use case diagram below.

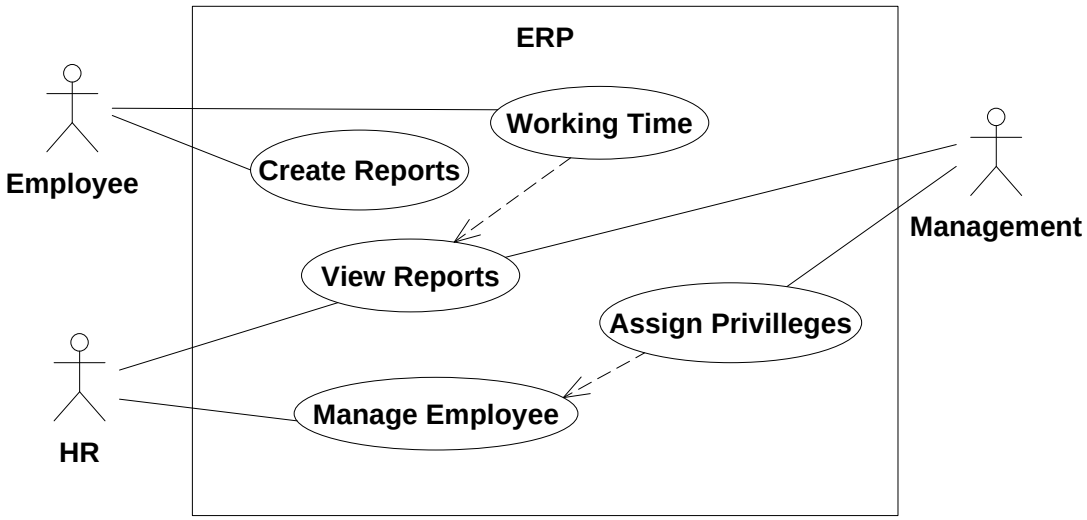

Figure 6.3. Simplified use case of the ERP system

6.2 IT Environment

The IT assets are typical for a small enterprise, including Samba file servers and various repositories. The general server operating system is Linux: in several ‘flavors’ like Ubuntu or Debian. Additionally, VMware ESXi servers act as a platform for virtual Windows and Linux guest machines used for development or as virtual servers.

ERP System

A virtual machine with Windows XP is used as server for the intranet and the company’s own proprietary Enterprise Resource Planning system. The system is based on PHP 5.2 and a MySQL 5.0 database as backend and data source and is running through XAMPP on an Apache 2.2 web server at all company sites.

Other Servers

It should be mentioned that the Debian Linux DHCP and DNS servers (Bind 9) are also hosted in a virtual machine today.

Among many unlisted others, additional servers are based at the German headquarters, which are also partially synchronized with the Thai branch.

6.3 Requirements

The previous statements let us make up the following wish list or better requirements for a companywide centralized user management:

1. Operating system independent (Client / Server)
2. Inexpensive, preferably open source software
3. Reliability and performance, as appropriate also redundancy and replication
4. Import of available inventory data

A cost-effective reliable multiplatform and multiuser solution is preferred. The possibility to integrate/import existing employee and user information is essential. Data replication might be used to establish independent authentication points at each branch to ensure availability in case of failure between both sites and to decrease access time.

6.4 Solutions and Specifications

The requirements could be met by several different approaches. Each one has its own advantages and disadvantages. Even if the given objective is to establish a general directory service, some other (pragmatic) solutions are mentioned in the following sections.

The following table shows selected services and their available authentication methods.

Table 6.1. Supported authentication methods

Service Name	Current Authentication	Alternative(s)
ERP System	MySQL	LDAP (through PHP)
Intranet	Password file	
Bug Tracking	MySQL	LDAP
Linux	Password file	PAM (LDAP, MySQL, RADIUS), NIS
CVS	Linux Account	
SVN	Password file	LDAP, MySQL, RADIUS (w. Apache)
Samba	Password file	LDAP, MySQL, RADIUS
WIFI	User password	RADIUS
Windows	User password	Samba as Domain Controller (PDC)
VPN Appliance	User password	LDAP, Active Directory

6.4.1 MySQL

Due to its popularity, many services, especially under open source Linux systems, are nowadays able to authenticate against MySQL[26] databases.

This very simple solution requires only effort in installing and configuring MySQL authentication modules for applications or services used. Necessary settings are server host, database username and password, database name, and table name. In addition, the table's attribute names for the username and password are needed. Additional information of how the password is stored, for example as MD5 hash value, is sometimes also required. With this given information, the authentication module will query the database. A non-empty result (only 1 row expected) normally means that the authentication is successful.

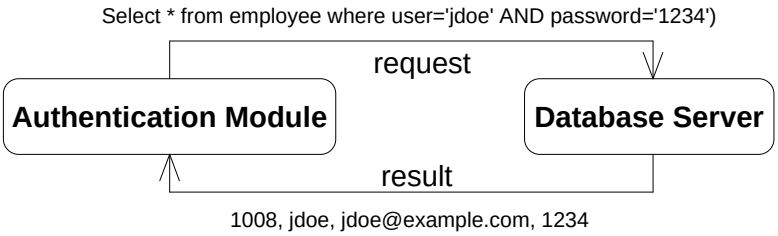

Figure 6.4. MySQL database authentication

However, as a drawback, redundancy and replication are not easy to implement and transport layer encryption could only be reached by tunneling the whole database traffic.[27]

[26] MySQL – The world's most popular open source database, Oracle; http://www.mysql.com
[27] For example, SSH tunnel or a Virtual Private Network (VPN)

6.4.2 RADIUS

RADIUS (Remote Authentication Dial In User Service) is a protocol defined in RFCs 2865[28] and 2866[29] that provides authentication, authorization and accounting (AAA) for network services. RADIUS is widely used to manage Internet and network access at Internet Service Providers (ISP) or enterprises, for example, in access points for wireless networks, port-based authentication[30] in network switches or Virtual Private Networks (VPN).

6.4.2.1 AAA System

In network security, authentication, authorization and accounting are also referred to as AAA. Beside RADIUS (Remote Authentication Dial In User Service) there exist a few other protocols providing AAA services. Common examples are Diameter[31], which is a successor to RADIUS, TACACS[32] (Terminal Access Controller Access-Control System) and also the Cisco Systems proprietary TACACS+ protocol. All protocols provide the following three common functions (Aboba, et al., 2003):

Authentication, as the "the act of verifying a claimed identity, in the form of a pre-existing label [...] or as the end-point of a channel (entity authentication)"

Authorization, as "the act of determining if a particular right, such as access to some resource, can be granted to the presenter of a particular credential"

Accounting, as "the act of collecting information on resource usage for the purpose of trend analysis, auditing, billing, or cost allocation"

[28] RFC 2865: Remote Authentication Dial In User Service (RADIUS) [2000]
[29] RFC 2866: RADIUS Accounting [2000]
[30] IEEE 802.1X-2004 - Port Based Network Access Control; http://www.ieee802.org/1/pages/802.1x-2004.html
[31] RFC 3588: Diameter Base Protocol [2003]
[32] RFC 1492: An Access Control Protocol, Sometimes Called TACACS [1993]

6.4.2.2 RADIUS Protocol

The RADIUS protocol operates between a client, in most cases a network device and a server. It runs at the application layer, using UDP.

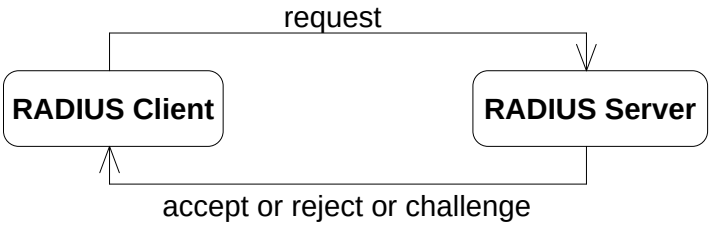

Figure 6.5. RADIUS authentication and authorization

The client sends an access request to the server, who authenticates the given credentials and will, as a response, reject, challenge, or allow access.

In detail, it is even more complicated. In the following example, a user wants to use a commercial wireless network (WIFI) for internet access. After he has connected to the hotspot (access point), the remote access server (RAS) asks for his username and password. On the other side, the RAS sends an access request to the RADIUS server, which processes the information by using its own user files or external sources like databases or directories. After authentication, the server will send its response to the Network Access Server (NAS), which will finally reject or allow access to the Internet. In rare cases, the user could be also challenged, i.e., asked for additional information, like a one-time password, to verify his identity.

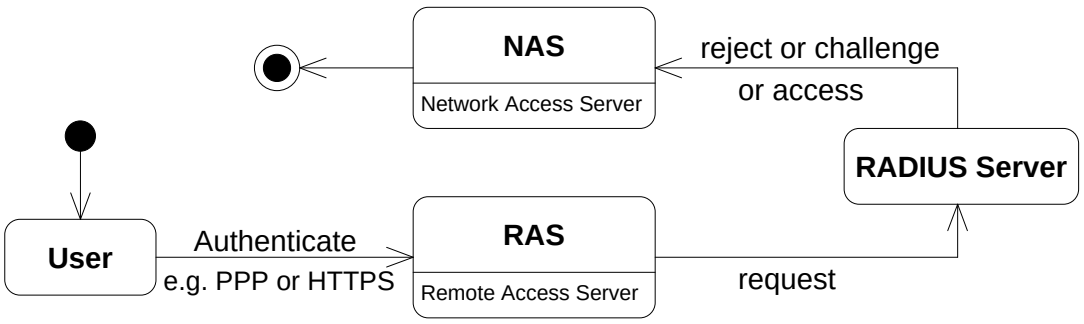

Figure 6.6. RADIUS component flow

Just as the MySQL solution in the section earlier, RADIUS authentication modules or clients need to be installed where available. In this case, however, most network devices will already have RADIUS support, with the result that we could cover a broader range of services and devices out of the box. As a last step, server settings need to be applied to all clients, so that they are able to communicate with the RADIUS server.

A RADIUS server is able to store user accounts and information locally or, more often, by enquiring a directory, for example, LDAP. The following chapters will explain directory services, and how a centralized user management can be implemented.

7 Directory Services

A directory service is a specialized database (Howes, et al., 2003) that uses hierarchical structures to store and process information. International standards define a directory as a tree-like structure, where data is statically stored. Each entry could contain arbitrary attributes, values or children.[33] Its optimized hierarchical tree like structure makes it valuable for lookup and search operations on huge datasets. A practical example, besides address books, is user authentication, which is often done between a client and a centralized directory service.

7.1 X.500 Directory Service

The early beginnings of directories evolved in the late 1980s by a predecessor of the International Telecommunication Union (ITU). The resulting X.500 standards consist of a series of ISO standards and recommendations (ITU-T, 2008). As a consequence, X.500 directories rely on a large suite of protocols, including DAP, DSP, DOP, and DISP[34]. Some of the X.500's strengths are the flexible and complete information model, its adaptability and openness. On the other hand, first the vendor implementations were flawed, not interoperable and did not perform or scale well. Another drawback is the very extensive and complex standard, of which still no complete implementation of X.500 exists (Apple, et al., 1997). Furthermore, X.500 was based on the OSI network protocol, which never prevailed against the simple and economic TCP/IP protocol. Nevertheless, it is possible to run X.500 over TCP/IP today. Finally, its top-down architecture made it less popular during times of an expanding internet. Internet growth (from bottom-up) and interconnected organizations with own independent deployments made global public (directory) service providers obsolete.

Notwithstanding, the LDAP designers have adopted many of the best ideas of X.500, while removing unneeded complexity.

[33] ISO/IEC 9594-1:2008 – [...] The Directory: Overview of concepts, models and services

[34] ITU-T X.519: Protocol Specifications describes among others, Directory Access Protocol (DAP), Directory System Protocol (DSP), Directory Operational Binding Protocol (DOP), and Directory Information Shadowing Protocol (DISP)

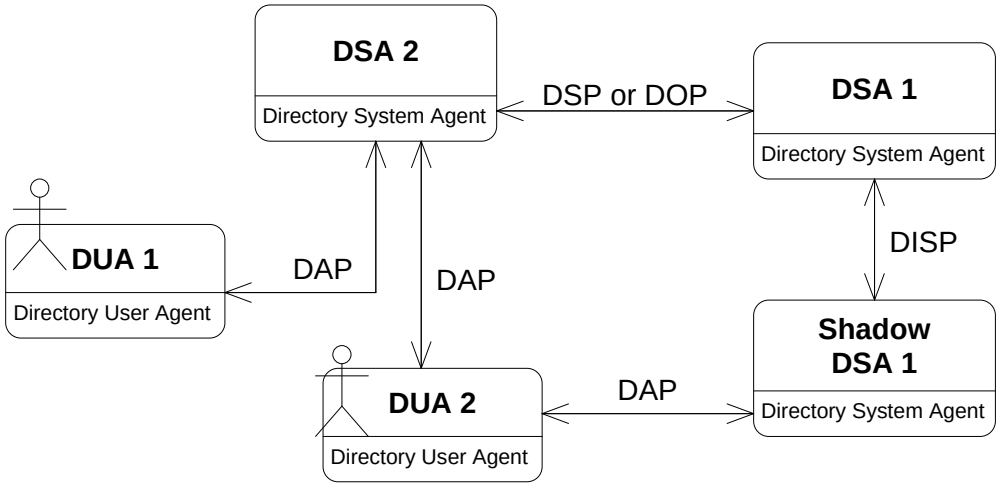

Figure 7.1. Components and Protocols of an X.500 Directory Service

7.2 Lightweight Directory Access Protocol (LDAP)

During early 1990's engineers at the University of Michigan and the Internet Engineering Task Force (IETF) researched and produced a lightweight directory access protocol for X.500 directories to improve and simplify directory services. Subsequently, a set of Internet standards called Request for Comments (RFC) has defined the Lightweight Directory Access Protocol. The first LDAP specification was published as RFC 1487[35] in 1993. In 1995, the first popular and widely used version of LDAPv2 followed as RFC 1777[36]. The latest version, LDAPv3 was published as RFC 2251[37] in 1997, and lastly updated as RFC 4510[38] in June 2006.

In general, LDAP simplifies the heavyweight X.500 DAP protocol in four important areas, to simplify implementation and improve performance (Howes, et al., 2003).

1. Functionality – almost same functionality at a much lower costs
2. Data representation – data carried as simple text strings
3. Encoding – only a subset of X.500 encoding rules is used
4. Transport – runs directly over the TCP/IP stack

[35] RFC 1487: X.500 Lightweight Directory Access Protocol [1993]
[36] RFC 1777: Lightweight Directory Access Protocol [1995]
[37] RFC 2251: Lightweight Directory Access Protocol (v3) [1997]
[38] RFC 4510: Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map [2006]

Early LDAP standards and implementations focused on directory client access problems, improving interoperability, performance and simplifying implementations; for example, by using a fast and simple C language API. The API of University of Michigan's first LDAP implementation (1992) became most widely a standard in RFC 1823[39]. On the server side, ldapd an LDAP–to–X.500 DAP protocol translator, became the interface between LDAP clients and X.500 directories.

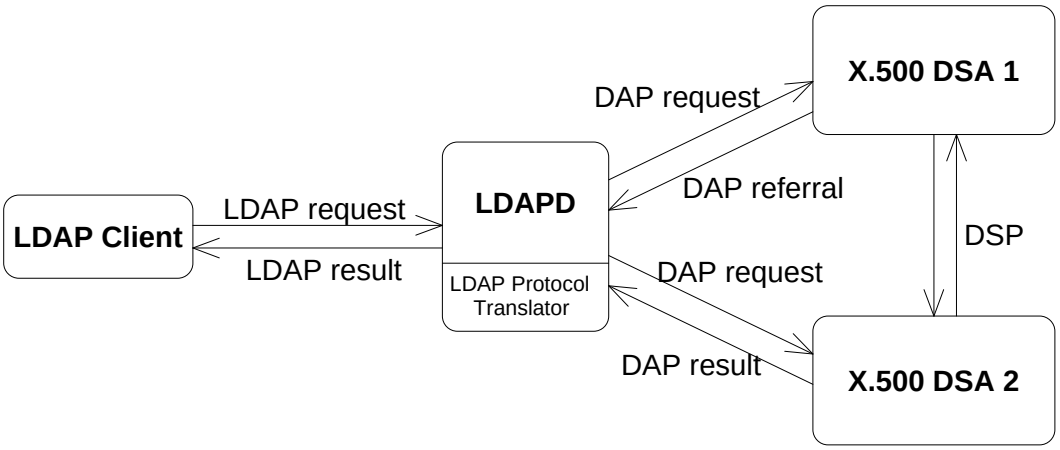

Figure 7.2. Components of a LDAP Directory System

Further development became again initiated by the University of Michigan, which exposed that more than 99 percent of their X.500 directory access came through LDAP. This statistical knowledge and the fact that server implementations and systems were still large and complex, led to the development of a standalone LDAP server, eliminating the necessity of X.500 directories and an intermediate ldapd server. This standalone LDAP server is designated as slapd, for standalone LDAP daemon, and was first released in U-M LDAP 3.2[40] in December 1995 (Howes, et al., 2003).

After this point, the success of LDAP was almost unstoppable. Its sophisticated open standard, free open source implementations, as well as support of many vendors and software companies[41] made it to the de facto standard for (internet) directory services.

[39] RFC 1823: The LDAP Application Program Interface [1995]
[40] University of Michigan LDAP implementation
[41] Netscape led a group of 40 software companies to establish LDAP as the directory service protocol of choice [1996]

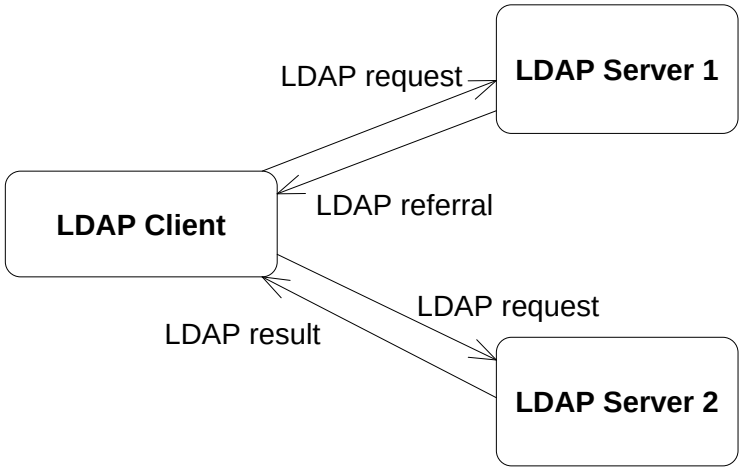

Figure 7.3. LDAP Standalone Directory Service (slapd)

In 1997, the latest LDAP version 3 became published. It enhances the prior LDAP versions in the following areas. (Howes, et al., 2003), (The OpenLDAP Project, 2011):

- Internationalization – using UTF-8[42] allows to store or process almost all languages by LDAPv3 servers or clients
- Security – Support for Simple Authentication and Security Layer (SASL)[43] and Transport Layer Security (TLS)[44]
- Referrals and Continuations – A new mechanism for returning referrals to other servers was added
- Extensibility – LDAPv3 is extensible for new functions and features
- Schema discovery – all servers publish their protocol versions and other useful information, to improve collaboration and interoperability

Today, LDAPv3 is supposed to be the last standardized major LDAP version, because of its flexibility and extensibility. This means that there is no need for further versions today.

Summarizing, the success of LDAP is well-founded by its many advantages. It is widely supported by many vendors and applications, but also inexpensive with high reliability,

[42] UCS Transformation Format 8; RFC 3629: UTF-8, a transformation format of ISO 10646 [2003]
[43] RFC 4422: Simple Authentication and Security Layer (SASL) [2006]
[44] RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 [2008]

performance and scalability. Moreover, LDAP directories are still simple and easy to understand, making it the universal, general-purpose directory service of choice.

7.3 Directory Information Tree

A Directory is a tree-like hierarchical data structure, saving directory entries. In LDAP, it is called a Directory Information Tree (DIT). Each entry in this tree could be a container, contain data itself or both at the same time.

The Distinguished Name (DN) identifies each object within the tree; it is read from the bottom to the top. Some RFCs and best practices suggest using Domain Name System (DNS)[45] names for the top most levels of the distinguished name, to make it globally unique for Internet use.

The following example of a tree shows the hierarchical namespace of a company (example.com) with the aid of its domain name. It is divided into two organizational units, people and groups, which contain the employees and users, as well as their belonging groups.

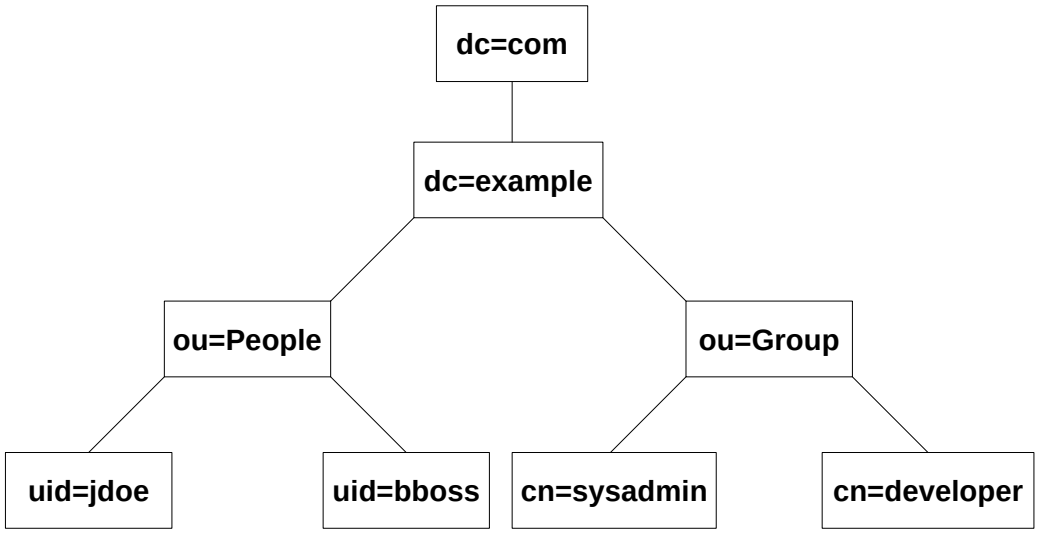

Figure 7.4. LDAP Directory Information Tree

[45] RFC 1034 & RFC 1035: Domain Names - Concepts and Facilities [1987] & Implementation and Specification [1987]

In the example above, the very left leave (the entry with user id ‘jdoe’) has the following Distinguished Name (DN), which should be and is globally unique.

```
dn: uid=jdoe,ou=People,dc=example,dc=com
```

An entry within the directory information tree consists of a set of attributes, which are defined in a schema. An attribute has a name and one or more values. The following example in LDAP Data Interchange Format (LDIF) shows an exemplary entry for the user ‘jdoe’ already mentioned in the figure above.

```
dn: uid=jdoe,ou=People,dc=example,dc=com
 uid: jdoe
 givenName: John
 sn: Doe
 telephoneNumber: +1 888 555 6789
 telephoneNumber: +1 888 555 1234
 mail: john.doe@example.com
 manager: uid=bboss,ou=People,dc=example,dc=com
 objectClass: inetOrgPerson
 objectClass: organizationalPerson
 objectClass: person
 objectClass: top
```

Figure 7.5. Directory entry in LDAP Data Interchange Format

LDIF is a plain text data interchange format for representing LDAP directory content and update requests, such as add, modify or delete operations. The following example adds a telephone number attribute to an existing entry from the example tree above.

```
dn: uid=bboss,ou=People,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: +1 888 555 9876
```

Figure 7.6. Modifying a directory entry with LDIF

A LDAP schema is a set of rules that describes what kind of data is stored. It helps to maintain consistency and quality of data. Schemas also provide applications a consistent interface to the data. They are assigned by one or more objectClass attributes. A schema contains among other details:

- Required attributes
- Allowed attributes
- How to compare attributes
- Limit what the attributes can store, e.g. restrict to a data type like integer

Furthermore, object classes are used to group information. They could also be used to estimate which users are allowed to access a certain type of account, for example, Samba (objectClass: sambaSamAccount). Entries can have multiple object classes, but must at least have one, a special class called top, to which all classes extend. In other words, objectClass is the only required attribute.

7.4 LDAP Protocol

The LDAP protocol operates at the application layer and runs over TCP/IP (by default on TCP port 389). LDAP clients are able to request the following major operations to servers:

- Bind – authenticate to a directory server
- Search – search and retrieve entries
- Compare – check if a named entry contains a given attribute value
- Add – add a new entry
- Delete – delete an entry
- Modify – modify an entry
- Modify Distinguished Name (DN) – move or rename an entry
- Start TLS – uses the LDAPv3 Transport Layer Security (TLS) extension

7.5 LDAP Directory Services

Today there are many commercial and free LDAP directory service implementations available. Their main differences are found in system inclusion, extend, support and costs. The following table lists a few chosen vendors with their released LDAP compliant directory service products.

Table 7.1. LDAP compliant directory services

Vendor	Name	Remarks
Microsoft	Active Directory	Requires Windows platform Advanced management features, also Kerberos for authentication
Novell	eDirectory	Uses a relational database Several interfaces (e.g., SOAP, ODBC etc.)
Oracle	Internet Directory	Stores and integrates into Oracle databases
IBM	Tivoli Directory Server	Built-in proxy server Integrates with IBM middleware Web-based user interface Focused on enterprise security
RedHat	389 Directory Server	GPL license Advancement of Netscape Directory Server Java-based GUI
OpenLDAP	slapd server	Free, open source license OS independent Lightweight program suite

7.6 LDAP Search Filters

Search operations and data lookups are the most common functions used in directories. A search operation needs three parameters: filter, scope and the search base.

The LDAP *filter* uses a powerful set of rules (Smith, et al., 2006) defined as RFC 4515[46], and defines the properties of the object(s) searched; for example, to find an entry with user id ‘jdoe’.

(uid=jdoe)

A more complex example returns all entries with the surname ‘Doe’, which also have a phone number.

(&(sn=Doe)(telephoneNumber=*))

The *scope*, however, defines where to search within the directory tree. It can be either base, one, subtree, or children; where base matches only the entry with provided DN, one matches the entries whose parent is the provided DN, subtree matches all entries in the subtree whose root is the provided DN, and children matches all entries under the DN (but not the entry named by the DN). (The OpenLDAP Project, 2011)

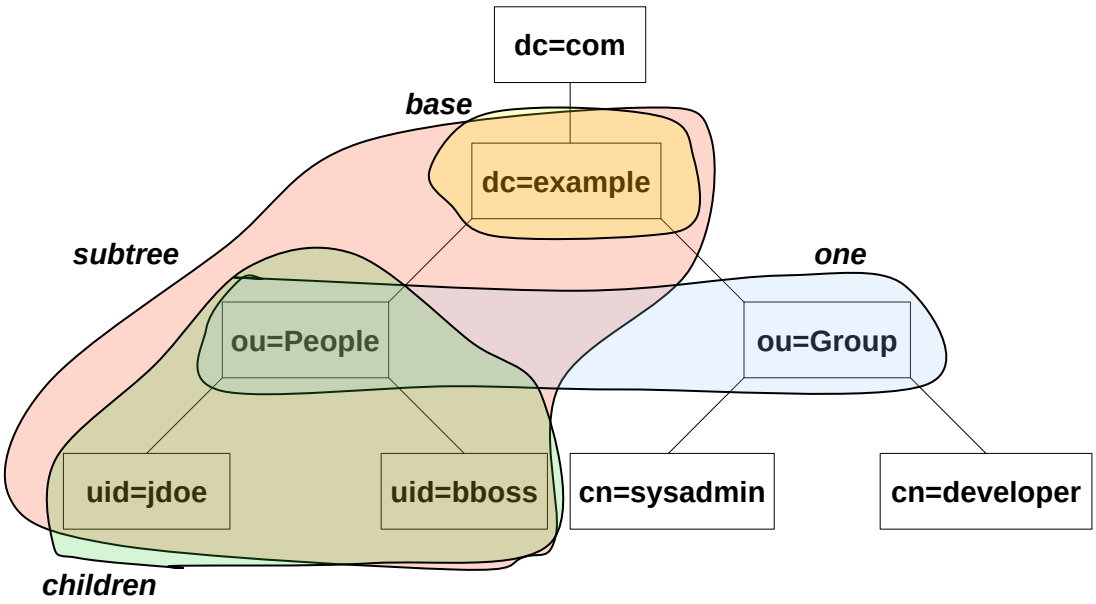

Figure 7.7. LDAP search scope for (dn: dc=com,dc=example)

[46] RFC 4515: Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters

The last parameter, *search base*, provides the Distinguished Name (DN) and this is where search operations are started and performed.

Filter, scope and search base are mighty tools and often used in LDAP and its configurations, for example, for access control and access lists.

8 Identity and Access Management

> Identity management is the centralized policy-based management of all information required for access to corporate systems by a person, machine, program, or other resource. (Panko, 2010)

Directory services like LDAP were the first attempt for unification and centralized management of user information and data. However, it is hardly possible to consolidate all information into one database or directory. For example, human resource records need a special protection, whereas phone numbers should be visible by everyone.

Identity management is an approach of centrally managing data between several (independent) data sources or information systems. For example, changes at a dataset in the HR records are synchronized with the identity management system, which again distributes the changes to other systems that are involved and need to be informed.

Summarizing, identity management is a centralized point for managing identities and their permissions. It acts as an interface between multiple systems and information sources, including access management.

Access management manages access privileges, administers security policies or enables single sign-on.

8.1 Single Sign-On (SSO)

> Single sign-on is designed to reduce the number of logins that a user has to remember and thereby create a more efficient and streamlined work environment for the user (Novell)

One advanced form of authentication is called single sign-on. SSO allows the user to authenticate only once at the beginning of his session, whereby no further visible authentications are needed for all integrated systems and services.

Single sign-on promises among higher productivity and easier administration (password management), enhanced security features. Phishing attacks can be recognized, since the SSO manages all identification and authentication processes, and the varying use of secure credentials lowers the risk of breaches and theft.

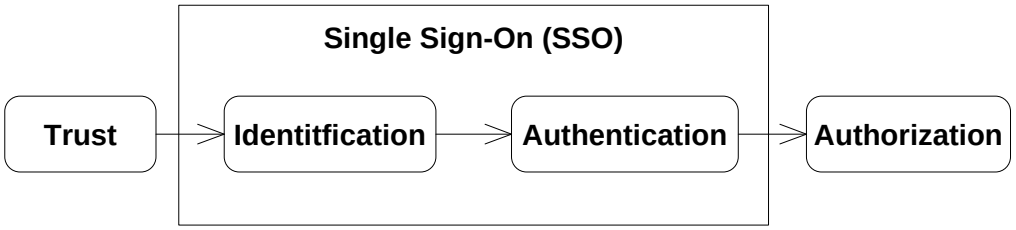

Figure 8.1. Classification of single sign-on

There are multiple approaches for single sign-on, which differentiate by type, if locally or stored in a centralized directory, as well as implementation using tickets (e.g. Kerberos), cookies or a Public Key Infrastructure (PKI). Local storage limits the user to a single workstation whereas a directory storage will make the user able to authenticate from anywhere in the organization's network. In addition, a combination of both systems offers more flexibility to mobile and remote users.

9 Establishment of a centralized User Management

After the evaluation of the requirements, a centralized user management using a directory service, has been implemented. Of all available LDAP products, OpenLDAP has been chosen because of its free and operating system independent implementation, which is also available as a pre-compiled package for most Linux distributions. Other aspects are OpenLDAP's flexibility, replication features and extensive documentation.

The following sections simply and shortly describe the conception, implementation, test and use of a LDAP directory service within the organization.

9.1 Directory Design

The real world constitution of the company is already a simple but very useful structure for a directory hierarchy and is transferred to and represented by the following model.

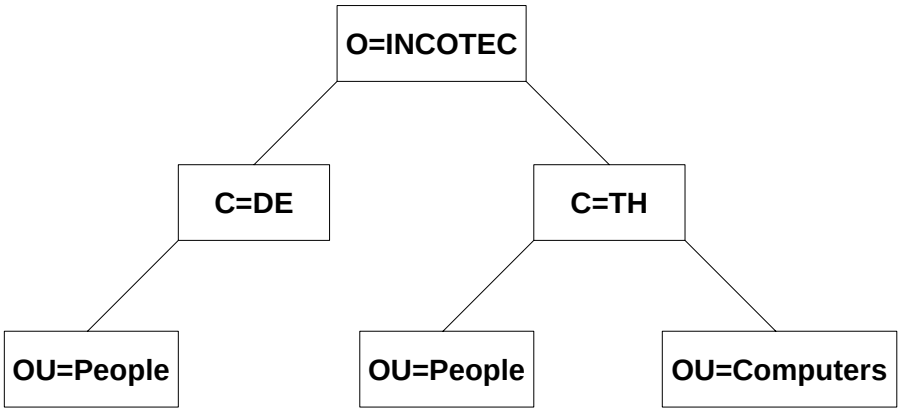

Figure 9.1. Organization's Directory Information Tree

The top is the organization followed by the German and Thailand sites. Both sites maintain their *People*, in other words employees and/or users. In addition, the Thai branch also manages their *Computers*, which include all network devices that need to be addressed by a DHCP server.

9.2 User Management

The main goal is to create and manage employees from a single point (ERP system). As mentioned earlier, the ERP system has already a broad knowledge for project and other reasons. However, this information is not accessible and used by other systems, yet.

9.2.1 ERP Interface

To provide the ERP system information as a data source for a LDAP directory, they must be exported somehow. A system modification to communicate with a LDAP server using the LDAP protocol seemed to be too expensive, therefore a synchronization approach was chosen. A newly created PHP script reads the ERP system database and provides all directory entries as a single LDIF file which can be easily imported to slapd, e.g., using the tool *ldapadd*.

The LDIF file contains three parts: employees, groups and computers. The former will be used for user authentication or to provide public information, like an (e-mail) address book. Groups will be used to authorize access and privileges for employees. At last, the DHCP server will use the computer information, to provide statically defined IP addresses to employee hosts.

9.2.2 Linux User Accounts

As a prerequisite, in this case, the Linux user IDs had to be harmonized between Germany and Thailand. As a consequence of decentralized management user ids at both sites were not corresponding and duplicate user ids assigned. In short, the assignment was not ideal for a centralized use. Therefore, a unique employee ID has been introduced, which is globally available for current and former employees. Based on that, all Linux user account user IDs needed to be temporarily replaced with their corresponding new ones. Last, the owners of all files and directories, as well as access control lists, needed to be changed to their new values, too. This was automatically done with a Python[47] script on the German and Thai file servers.

[47] Python Programming Language; http://www.python.org

9.3 Implementation

The installation and configuration consists of several parts, beginning with the LDAP server, followed by client software which will access the directory service.

9.3.1 OpenLDAP

The LDAP standalone server can be easily installed under Linux. In this example, the Debian[48] aptitude packages of OpenLDAP (slapd) have been used. The server is almost ready to run out of the box. Only a few properties, like the directory suffix, root dn and password needed to be set. For further reading the OpenLDAP Administrator's Guide (The OpenLDAP Project, 2011) is highly recommended.

9.3.1.1 Data Integration

Next, the employee information needs to be imported to the directory. In this case, a daily cron[49] job will execute the following tasks to update the directory database:

1. Download LDIF file from ERP system
2. Shutdown slapd (LDAP daemon)
3. Delete old directory database
4. Insert LDIF data to directory with ldapadd command line tool
5. Start slapd

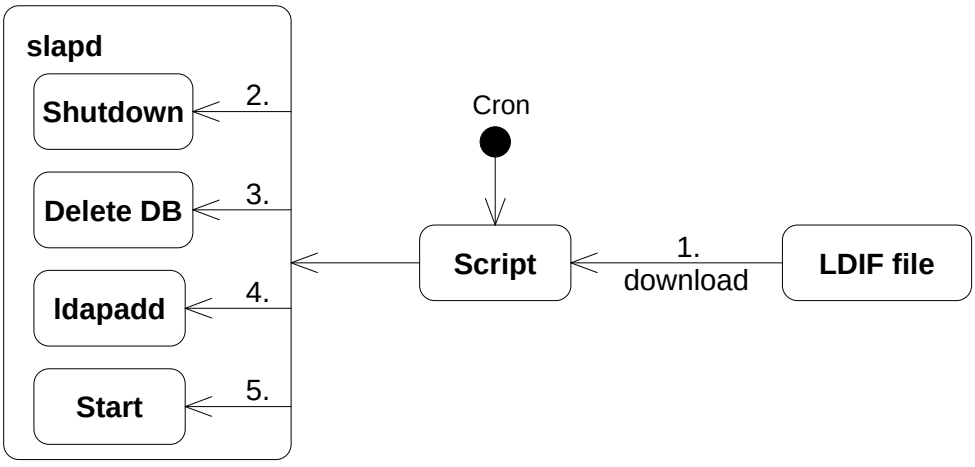

Figure 9.2. Periodically updating the LDAP directory

[48] Debian - The Universal Operating System; http://www.debian.org
[49] Cron is a time-based task scheduler in Linux operating systems

This simple solution has the drawback that during the data import the directory will be unavailable. However, in this case, it is neglible, because nobody works during nighttime; in addition, it takes a few seconds only to authenticate employees with a second sign-on. More seriously, in this procedure there is a long update delay because new users and data changes will be automatically applied at fixed times only.

9.3.2 Linux Authentication

The Linux Pluggable Authentication Modules (PAM) mechanism allows easy LDAP integration. The only task is to load the LDAP-PAM modules and to add LDAP to the System Databases and Name Service Switch (NSS) configuration.

9.3.3 DHCP Server

Next, the used DHCP server (dhcpd) needed to be enabled for the use of our LDAP directory's instead of its local configuration files. Therefore, the dhcpd-ldap module/patch was installed from the Debian Linux package repositories and configured with the corresponding LDAP server credentials. Another important point is the installation of the dhcpd schema (dhcpd.schema) in OpenLDAP which allows managing and maintaining DHCP server configurations.

9.4 Test

In theory, test is an very important stage in all kind of deployments and life cycles. In practice, however, testing took place in virtual machines at initial stages, for slapd configuration and test, as well as Linux and file server integration. Thus, three virtual Linux machines were involved which simulate a small ideal world test environment for first experiments and complex configurations.

After the initial stage, LDAP has been installed on a working server environment, adding productive systems consecutively. This can be seen as a major gap, regarding sophistication and accuracy. However, the time budget for these tasks was limited due to other functions and responsibilities.

10 Conclusion

Successful security engineering depends on many predictable as well as unpredictable factors. Access control and authentication are just keystones on the long way to a protected and secured Information Technology infrastructure, as seen in this thesis. Furthermore, many disciplines and work fields overlap and experts of many areas need to work together for a successful outcome. Even specialists offering their services for much money are not safe of security breaches and attacks against their own institutions.

Although many facts and details of these incidents will never become public, basic rules may have prevented successful intrusion. This is proved by, e.g., the mentioned certificate theft at Comodo, in March 2011. Strict access control with limited API access, to the registration server only, might have hindered the intruder, even if he knew the credentials. Of course, also these measures are not to last, but they are simple and efficient steps for best adequate security.

Other victims in year 2011 were Sony, with outdated vulnerable software on their PSN servers, and RSA[50] due to a flaw in Adobe's Flash Player, where the intruders may have managed to obtain the seeds and serial numbers of SecurID tokens. Last but not least, HBGary and Barracuda Networks[51], also security firms with good reputation, were targets during the first months of the year.

10.1 User Management with LDAP

As a consequence of the increasing threats for the security of assets, networks and information, the implemented user management using a LDAP directory service is a fundamental but vital part of the organization's (future) security strategy.

Simplified central management capabilities, as well as consistent users and permissions on almost all machines allow an easy accurate operation, whereby also security drastically increased by using a continuous access control.

[50] http://www.h-online.com/security/news/item/RSA-break-in-it-was-the-Flash-Player-s-fault-1221057.html
[51] http://www.h-online.com/security/news/item/Data-theft-at-network-security-firm-1226663.html

10.2 Future Development and Extension

After the first stage, a few extensions and long-term goals for further development can be already collected.

First of all, the permission management could be improved, introducing work projects as access levels in the server systems and their access lists. Thus, a more detailed security distinction could be used limited to the user who is only able to see and do what he needs to.

Secondly, a Radius server with an LDAP backend could be established. In this way, network devices could authenticate against the centralized user database. As there are switches and wireless access points, to secure network (port) access.

Entirely, the whole development is benefiting the organization’s security and work processes. Even if a few adaptations need to be made in case of changing requirements and future processes, the system is a valuable and expandable part of daily business and IT operations.

Acknowledgements

Concluding this thesis, written in several, much more than expected, months of discontinuous work, some very own findings came up.

One of the most important lessons learned is that it is very hard to arrange family life, full time work and scientific writing in one person's daily routine; much harder than I ever thought. I had to realize that there would not be enough time for all tasks waiting to be done. Some decisions and prioritization needed to be done, either seeing how the kids were growing up, or earning the money to keep the kids growing healthy in Bangkok, Thailand's capital of extremes...

The popular saying "time is money" seems to be right, but the reverse looks infeasible.

Summarizing, just humanly not scientifically approved, something or someone always falls by the wayside. In this case however, these pieces of typed (electronic) paper, valuable or not, and a few persons who believed in me; expected that I am able to get the job done.

I still do not know why. I am incessantly looking for something I could have done better, a reason. Why it did not work out?

Is there any answer?

After all, it is a busy narrow one-way road...the pursuit of happiness.

Still well done?

For my family and friends, wherever you are.

Rattanaphon, Juthathip & Korvin

I also would like thank some special people, accompanying my studies and sophisticated life for quite a while. On the study side, especially for their labor, support and patience, to get this almost endless work finally done.

Prof. Dr. Wolfgang Gerken and my university *HAW Hamburg*,

My supervisor *Dr. Vesa Torvinen* and *Turku University of Applied Sciences*,

Prof. Jean-Yves Antoine and *Université François-Rabelais*.

Last but not least, *Poppy Skarli* for a still unforgotten time in Finland.

Bangkok, 17.05.2011

APPENDIX

Table A.1. Common LDAP Abbreviations

Acronym	Long Form
C	Country
CN	Common Name
DAP	Directory Access Protocol
DC	Domain Component
DIT	Document Information Tree
DN	Distinguished Name
DSA	Directory System Agent
DUA	Directory User Agent
L	Location
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
O	Organization
OID	Object Identifier
OU	Organizational Unit
RDN	Relative Distinguished Name
SN	Surname
ST	State
UID	User ID

REFERENCES

Aboba, B. and Wood, J. 2003. Authentication, Authorization and Accounting (AAA) Transport Profile. Request for Comments: 3539. [Online] 2003. http://tools.ietf.org/html/rfc3539. RFC 3539.

Anderson, Ross. Security Engineering: A Guide to Building Dependable Distributed Systems. First Edition. s.l. : Wiley. http://www.cl.cam.ac.uk/~rja14/book.html. ISBN 0470068523.

Apple, C. and Rossen, K. 1997. X.500 Implementations Catalog-96. Request for Comments: RFC 2116. [Online] 1997. http://www.rfc-editor.org/rfc/rfc2116.txt.

Arkills, Brian. 2003. LDAP Directories Explained: An Introduction and Analysis. s.l. : Addison Wesley, 2003. ISBN 0-201-78792-X.

Bhattacharya, Sandip; et al. 2003. Professional Apache Security. s.l. : Wrox Press, 2003. ISBN 1-86100-776-0.

Bishop, Matt. 2002. Computer Security: Art and Science. s.l. : Addison Wesley, 2002. ISBN 0-201-44099-7.

—. 2004. Introduction to Computer Security. s.l. : Addison-Wesley, 2004. ISBN 0-321-24744-2.

Bundesnetzagentur. 2010. Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen). Mainz, Germany : s.n., 22. 12 2010.

Gattiker, Urs E. 2004. The Information Security Dictionary. s.l. : KLUWER ACADEMIC PUBLISHERS, 2004. ISBN 1-4020-7927-3.

Hannebohm, Martin. 2011. Konzeption und Umsetzung einer unternehmensweiten einheitlichen Benutzerverwaltung. s.l. : HAW Hamburg, 2011.

Harris, Shon. 2002. Mike Meyers' CISSP Certification Passport. s.l. : McGraw-Hill Osborne Media, 2002. ISBN 0072225785 .

Heise Media UK Ltd. 2011. Attack on the PlayStation Network: what customers should now watch out for . The H Security. [Online] Heise Verlag, April 28, 2011. http://www.h-online.com/security/news/item/Attack-on-the-PlayStation-Network-what-customers-should-now-watch-out-for-1233905.html.

—. 2011. Single hacker claims responsibility for Comodo certificate theft. The H Security. [Online] Heise Verlag, March 28, 2011. http://www.h-online.com/security/news/item/Single-hacker-claims-responsibility-for-Comodo-certificate-theft-1216417.html.

Howes, Timothy A., Smith, Mark C. and Good, Gordon S. 2003. Understanding and Deploying LDAP Directory Services. s.l. : Addison Wesley, 2003. ISBN 0-672-32316-8.

ISO/IEC 27000. 2009. Information technology - Security techniques - Information security management systems - Overview and vocabulary. 2009. ISO/IEC 27000:2009(E).

ISO/IEC 27001. 2005. Information technology - Security techniques - Information security management systems - Requirements. 2005. ISO/IEC FDIS 27001:2005(E).

ISO/IEC 27002. 2005. Information technology - Security techniques - Code of practice for information security management. 2005. ISO/IEC 27002:2005(E).

ITU-T. 2008. Recommendation X.500 (11/08). X.500 : Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services. [Online] 11 2008. http://www.itu.int/rec/T-REC-X.500/e. ITU-T X.500.

Kaliski, B. 2000. PKCS #5: Password-Based Cryptography Specification Version 2.0. [Online] September 2000. http://tools.ietf.org/html/rfc2898. Request for Comments: 2898.

Kissel, et al. 2006. Guidelines for Media Sanitization. [Online] September 2006. http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf. NIST Special Publication 800-88.

Krawczyk, H., Bellare, M. and Canetti, R. 1997. HMAC: Keyed-Hashing for Message Authentication. [Online] December 1997. http://www.faqs.org/rfcs/rfc2104.html. Request for Comments: 2104.

Manuel, Stephane. 2008. Classification and Generation of Disturbance Vectors for Collision Attacks against SHA-1. CRI - Paris Rocquencourt. [Online] 2008. http://eprint.iacr.org/2008/469.pdf.

Menezes, Alfred J., Oorschot, Paul C. van and Vanstone, Scott A. 2001. Handbook of Applied Cryptography. Fifth Printing. s.l. : CRC Press, 2001. http://www.cacr.math.uwaterloo.ca/hac/. ISBN 0-8493-8523-7.

NIST. 2007. Recommendation for Key Management – Part 1: General (Revised). National Institute of Standards and Technology. [Online] March 2007. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf. NIST Special Publication 800-57.

—. 2011. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. National Institute of Standards and Technology. [Online] January 2011. http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf. NIST Special Publication 800-131A.

Novell. Single-Sign-on: Finding The Best Fit. White Paper. [Online] http://www.novell.com/products/securelogin/single_signon.pdf.

Orman, H. and Hoffman, P. 2004. Determining Strengths For Public Keys Used For Exchanging Symmetric Keys. Request for Comments: 3766. [Online] April 2004. http://www.faqs.org/rfcs/rfc3766.html. RFC 3766.

Panko, Ray. 2010. Corporate Computer and Network Security. Second Edition. s.l. : Pearson Prentice-Hall, 2010.

Parker, Donn. 2002. Toward a New Framework for Information Security. The Computer Security Handbook (4th ed.). [Online] 2002. http://www.computersecurityhandbook.com/CSH4/Chapter5.html. ISBN 0471412589.

Peltier, Thomas R. 1998. Information Security Policies and Procedures: A Practitioner's Reference. s.l. : Auerbach, 1998. ISBN 0-8493-9996-3.

Pfleeger, Charles P. 2006. Security in Computing. Fourth Edition. s.l. : Prentice Hall, 2006. ISBN 0-13-239077-9.

Rivest, Ronald L. and Kaliski, Burt. 2003. RSA Problem. MIT Laboratory for Computer Science. [Online] December 10, 2003. http://people.csail.mit.edu/rivest/RivestKaliski-RSAProblem.pdf.

SECG. 2009. SEC 1: Elliptic Curve Cryptography. Standards for Efficient Cryptography. [Online] 2009. http://www.secg.org/download/aid-780/sec1-v2.pdf.

Shim, Jae K., Qureshi, Anique A. and Siegel, Joel G. 2000. The International Handbook of Computer Security. s.l. : The Glenlake Publishing Company, Ltd., 2000. ISBN 0-8144-0579-7.

Slone, Skip. 2004. Identity Management. The Open Group. [Online] 2004. White Paper. http://www.opengroup.org/projects/idm/uploads/40/9784/idm_wp.pdf.

Smith, M. and Howes, T. 2006. Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters. Request for Comments: 4515. [Online] 2006. http://tools.ietf.org/html/rfc4515. RFC 4515.

Stallings, William. 2005. Cryptography and Network Security Principles and Practices, Fourth Edition. s.l. : Prentice Hall, 2005. ISBN 0-13-187316-4.

The OpenLDAP Project. 2011. OpenLDAP Software 2.4 Administrator's Guide. [Online] 2011. http://www.openldap.org/doc/admin24/.

Vacca, John R. 2009. Computer and Information Security Handbook. s.l. : Morgan Kaufmann, 2009. ISBN 978-0-12-374354-1.

Wei, Michael, et al. 2011. Reliably Erasing Data From Flash-Based Solid State Drives. [Online] 2011. http://www.usenix.org/events/fast11/tech/full_papers/Wei.pdf.

Wikipedia. 2011. Wikipedia, the free encyclopedia. [Online] 2011. http://en.wikipedia.org/wiki/.

—. 2011. Elliptic curve cryptography. Wikipedia, the free encyclopedia. [Online] 2011. http://en.wikipedia.org/wiki/Elliptic_curve_cryptography.

—. 2011. Information Assurance. Wikipedia, the free encyclopedia. [Online] 2011. http://en.wikipedia.org/wiki/Information_assurance.

—. 2011. Information security. Wikipedia, the free encyclopedia. [Online] 2011. http://en.wikipedia.org/wiki/Information_security.

—. 2011. PBKDF2. Wikipedia, the free encyclopedia. [Online] 2011. http://en.wikipedia.org/wiki/PBKDF2.

Zeilenga, K. 2006. Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map. Request for Comments: 4510. [Online] OpenLDAP Foundation, 2006. http://tools.ietf.org/html/rfc4510. RFC 4510.