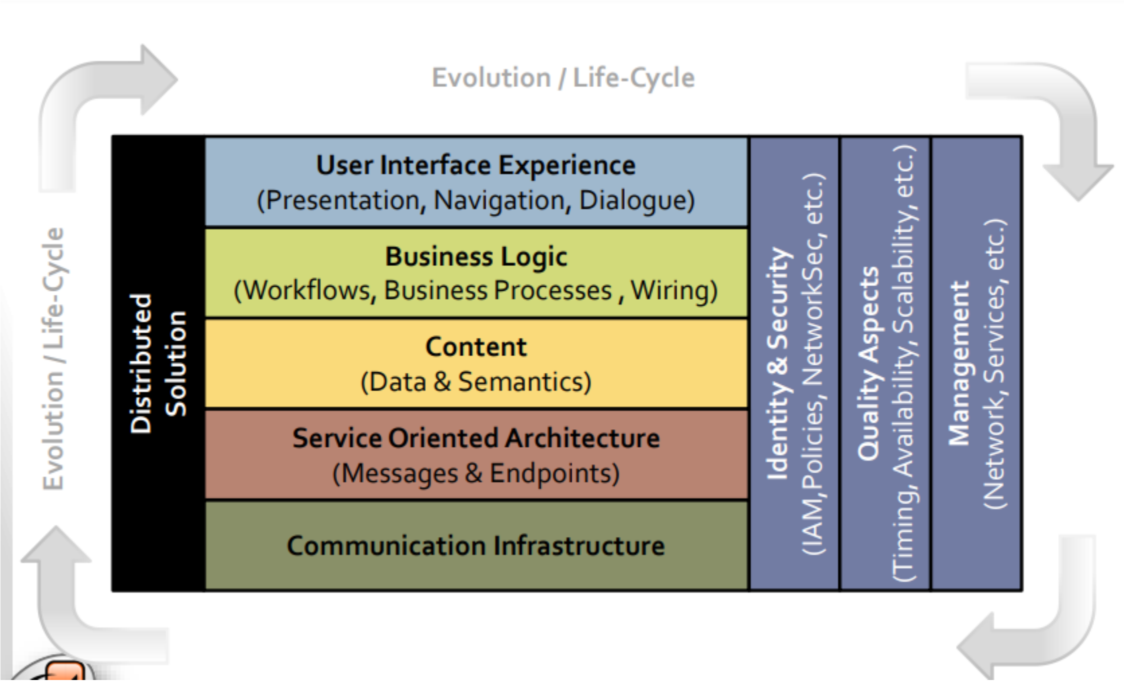
SVS - Lectures

eo shiru

July 11, 2019

Contents


Evolution / Life-Cycle
Evolution / Life-Cycle
Distributed Solution
User Interface Experience
(Presentation, Navigation, Dialogue)
Business Logic
(Workflows, Business Processes , Wiring)
Content
(Data & Semantics)
Service Oriented Architecture
(Messages & Endpoints)
Communication Infrastructure
Identity & Security
(IAM,Policies, NetworkSec, etc.)
Quality Aspects
(Timing, Availability, Scalability, etc.)
Management
(Network, Services, etc.)

The increasing decentralization of public networks by deregulation of telecommunications markets leads to increasingly extensive use of the Web and increasing usage of the open & decentralized Internet. Back then networks were mostly closed and managed centrally, while the Internet was used as a pure research and didn't have worthwile targets. Because of the recent evolution of the internet and the world wide web security mechanisms are becoming an indispensable part of modern communication systems. Security must be considered in a comprehensive & integrated way, taking new aspect into account: identity and privacy.
But what is **security**? Security refers to the ability to avoid being harmed by any risk, danger or threat (Cambridge Dictionary of English). In practice and regarding IT infrastructure this is an *unreachable goal*. Therefore your software is never a 100% secure.

0.0.1 Security Goals

Our focus is on actions to achieve security goals.
Mnemonic for security goals: "**CIA**":

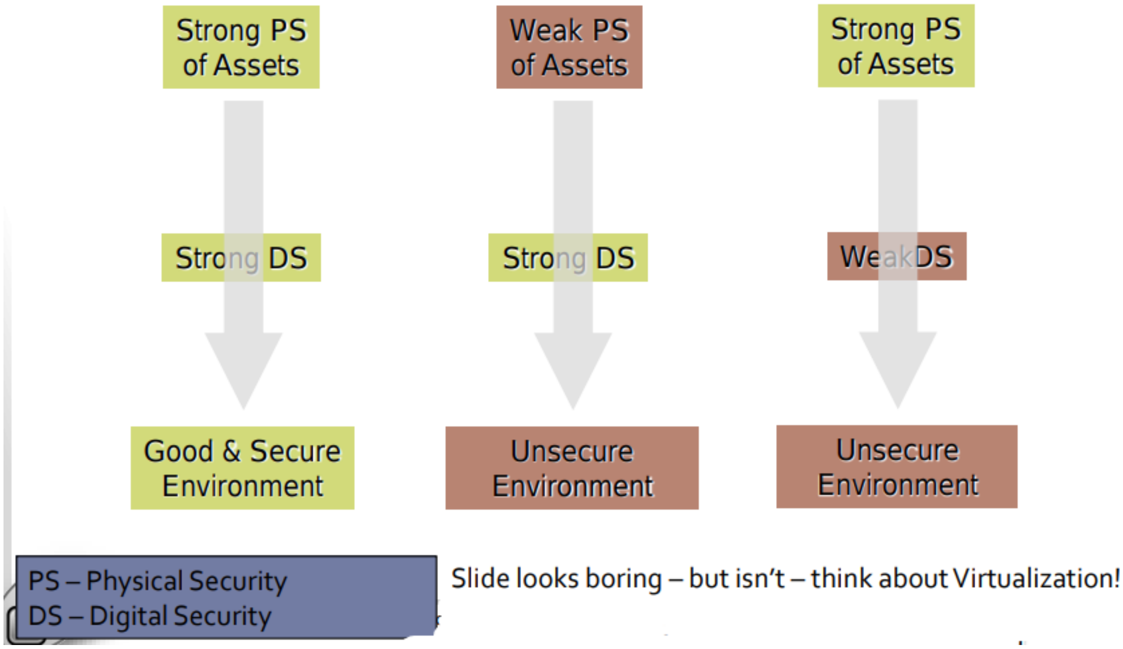
- Confidentiality
  - data secrecy
- Integrity
  - data intactness
- Authenticity

  - secure data origin

- additional (soon-to-be) major goals:
  - Liability (Non-Repudiability)
    - non-repudiation (repudiation = Zurückweisung, Nichtanerkennung) of data origin
    - important for contracts or in the fight against SPAM
  - Identity
    - verification of an individual entity
    - nowadays identity is of increasing significance

In this lecture the generic term **Assets** denotes things worth protecting eg data or services (business applications for example).

A strong physical security is the foundation to protecting assets and achieving the security goals. Physical vs Digital Security:



Strong PS of Assets

Strong DS

Good & Secure Environment

Weak PS of Assets

Strong DS

Unsecure Environment

Strong PS of Assets

Weak DS

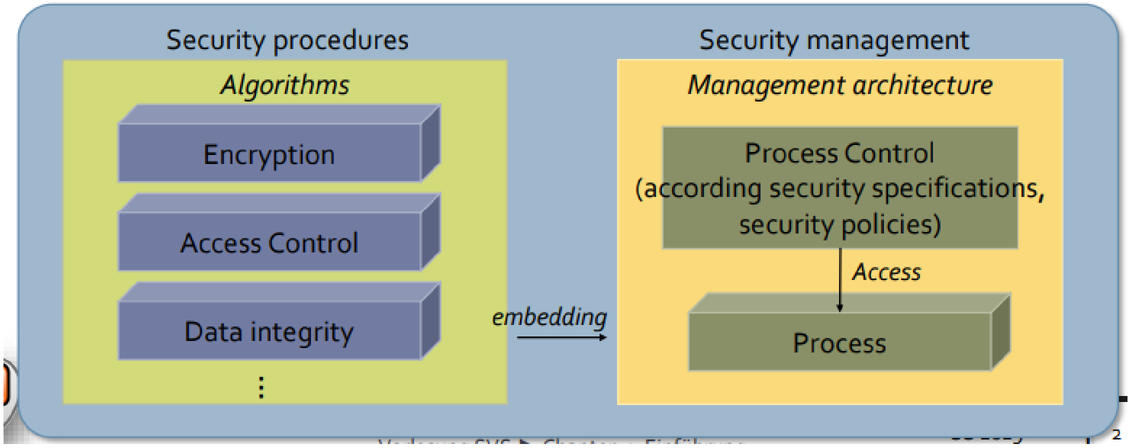
Unsecure Environment

PS – Physical Security
DS – Digital Security

Slide looks boring – but isn't – think about Virtualization!

We can achieve the security goals mentioned in the previours lecture by:

- information encryption
- implementation of authentication
- establishment of security activities
- monitoring of the system or the network in terms of attacks
- continous reduction of weak spots
- etc



In the data transfer model (2 users communicating) we can distinguish for example two types of attackers:

- passive attacker
  - can only listen, not manipulate
  - confidentiality threat
- active attacker
  - can listen, change, delete, duplicate
  - threat for confidentiality, integrity and authenticity

The difference between authenticity and liability lays in the focus between internal and external relationships. Here's an example: Authenticity means that Bob is sure the data comes from Alice (internal) - Liability means that Bob can prove this to third parties (external).

Here are some types of threats: Interception of transmitted data

- Modification of transmitted data
  - Change
  - Delete
  - Insert
  - Reorder data blocks
- Masquerade
  - Faking a false identity
  - Sending messages with a false source address
- Unauthorized access to systems
  - Keyword „Hacking"
- Sabotage (Denial of Service)
  - Causing an overload situation (including hardware)

  - "Destroying" protocol instances by illegal packets

And here are *some* attack techniques: Tapping cables or radio links

- Interposing (man-in-the-middle attack)
- Replaying of intercepted messages (replay attack) (e.g. replay of login messages for the purpose of unauthorized access)
- Selective changing / swapping of bits or bit strings (without being able to decrypt the message)
- Break-in by taking advantage of errors (buffer overflows)
- Break-in by means of active components (trojans, worms, backdoors)
- Breaking cryptographic algorithms
- Social Engineering (e.g. through direct contact and social web)

And some countermeasures:

- Don´t use self-made algorithms, use only proven algorithms that are considered safe!
- Use safe methods and replace old algorithms
- Behaviour (Pattern) analysis
- Use Social Web the right way
- Know your enemy
- limit the attack surface
- limit identity properties
- distribute attack surface
- apply encryption everywhere

Security should be considered in an integrated way. This means:

- consideration of all assets
- based on risk assesment
- use adequate security approaches and services (often a mix of different techniques)

All in all it is almost impossible to achieve 100% security. Therefore one has to clearly define what has to be protected and how high the according security requirements should be.

0.1 Excourse: GDPR

What is a data subject?

- any person whose personal data is being collected, held or processed

What are the data subject's rights?

- Individuals/Citizen (data subjects) have the right to:
  - information about the processing of your personal data
  - obtain access to the personal data held about you
  - ask for incorrect, inaccurate or incomplete personal data to be corrected
  - request that personal data be erased when it's no longer needed or if processing it is unlawful
  - object to the processing of your personal data for marketing purposes or on grounds relating to your particular situation
  - request the restriction of the processing of your personal data in specific cases
  - receive your personal data in a machine-readable format and send it to another controller ('data portability')
  - request that decisions based on automated processing concerning you or significantly affecting you and based on your personal data are made by natural persons, not only by computers; You also have the right in this case to express your point of view and to contest the decision

What is personal data and what not?

→ Personal data is any information that relates to an identified or identifiable living individual. Different pieces of information, which collected together can lead to the identification of a particular person, also constitute personal data.

Personal data that has been de-identified, encrypted or pseudonymised but can be used to re-identify a person remains personal data and falls within the scope of the law.

Personal data that has been rendered anonymous in such a way that the individual is not or no longer identifiable is no longer considered personal data. For data to be truly anonymised, the anonymisation must be irreversible.

Examples of personal data:

- name and surname, home adress, email adress
- identification card number
- location data
- IP adress
- cookie ID

Examples of data not considered personal data:

- a company registration number
- an email adress such as info@company.com

- anonymised data

What is a data controller?
The controller or data controller is simply the organization (a legal person, agency, public authority, etc.) or the natural person which, alone or depending on the organization and personal data processing activity, in collaboration with others defines what needs to happen with the personal data (and also collects personal data) and obviously is key in personal data protection. Formal definition (Article 4):
'controller' means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data; where the purposes and means of such processing are determined by Union or Member State law, the controller or the specific criteria for its nomination may be provided for by Union or Member State law

What is a data processor?
The processor or data processor is a person or organization who deals with personal data as instructed by a controller for specific purposes and services offered to the controller that involve personal data processing (remembering that processing can be really many things under the GDPR). The formal definition of the processor as you can read it in the GDPR Articles (GDPR Article 4):
Processor means a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller. The main difference to data controllers is that the GDPR has a really different stance with regards to data processors whereby they have duties and responsibilities that are directly applicable and can be directly enforced and GDPR compliance is a shared obligation as you will discover.

1 Attacks on End Systems

Attacks on end systems via

- computer viruses
- computer worms
- trojan horses
- exploits
- cracking systems

might focus on

- unsecured computer systems
- exploiting programming errors
- bad security measures
- weak passwords

Computer Virus

- based on biological model
- infects resources of the host system to replicate itself
- malicious functions

  - load generation
  - data corruption
  - spying

- various types
  - boot sector viruses
  - file viruses
  - macro viruses
  - script viruses
  - composites

- self-defense mechanisms of viruses:
  - stealth
  - modification
  - cryptographic methods
  - polymorphism
  - retroviruses (against anti-virus programs)

- passive distribution: by embedding into other programs and execution by the host system

Computer Worm

- based on biological model

- uses resources of the host system and of the network to spread over to other systems *automatically* in order to execute its malicious function there

- malicious functions
  - load generation
  - data corruption
  - spying
  - spamming
  - DDoS

- various types
  - email worms (social worms, file attachment, active content)
  - interactive worms (ask the user "please press OK" to use exploits)
  - instant messaging worm (sending of malicious software / links to all chat partners)
  - IRC worms (usage of scripting in IRC programs)
  - P2P worms (at file-sharing sites: tempting names → download)
  - cell phone worms (distribution via Bluetooth, MMS, etc)

- often in combination with other forms of malware, eg viruses droppers, backdoors, trojans

Dropper (virus dropper, DDoS dropper)

- executable program that acts as a carrier program for malware
- is usually terminated after the virurs has been installed

Injector

- similar to dropper, but the malware will only be "installed" (injected) in memory

Backdoor

- part of a program that allows users to gain acces to the machine / system bypassing the normal access security
- variants: default passwords (BIOS); specially equipped passwords / routines / servers that allow access (sometimes subsequently installed programs)
- closely linked to trojans and droppers

Trojan (trojan horse)

- similar to the well-known story..
- program that executes a potentially harmful function without user′s knowledge
- attention: often mixed up in the context of rootkits and backdoors

Rootkit (administrator toolbox)

- collection of software tools for concealment and stealth intrusions of malicious software
- examples: hiding backdoors by hiding processes, logs, log-ins

Exploit

- a program (including scripts & macros) that exploits the weaknesses or failures of a system or another application to obtain privileges or to use it for DoS attacks

Malware as generic term refers to malicious or unwanted software or programs.
Buffer Overflow

- application reserves a buffer to store some input values
- the length of the input is larger than the buffer but the whole input still gets processed
- memory space outside the buffer gets overwritten/accessed

2 Attacks on Infrastructures

Attacks on infrastructures via

- attacks on signaling mechanisms
- distributed denial of service (DDoS)
- attacks on WLAN hotspots and routers
- break-ins (password theft, bugs, exploits)

might focus on

- unsecured intermediate system
- overload situations
- unsecure data storage
- weak passwords

Attacks on signaling mechanisms

- ICMP: fake control messages
- RSVP: fake resource allocation

Attacks on router

- attacks on routing protocols
- distribution of false routes
- WLAN, Bluetooth etc

Attacks on Hardware, eg virtual server

- usb-attacks

Denial of Service Attack

- the targeted weak spot is the overload of the network component
  - may result in loss of service or entire computer systems
- attack possibilities
  - basic principle: large amount of requests sent to the target service or target system
  - requests must be designed in a way that they lead to an overload situation (more efficient use of exploits)
- examples:
  - ping of death = fake "echo request" information leads to a crash
  - smurf = broadcasting of an ICMP "echo request" with false return address (address of the victim)

- special forms
  - **Distributed** DoS = coordinated attack with a large number of computers
    - closely linked with trojan / droppers infected systems that can be used as a remote-controlled attack network (BotNets)
    - BotNets - Malware starts its DDoS attacks after being distributed via a dropper

WLAN Attacks

- the targeted weakspot is the transmission medium as well as utilizing encryption techniques
- attack scenario
  - capture data packets of a protected WIFI network
  - "attack" on encryption → search for a key
  - use the found key for further attacks in the protected network
- examples: wepcrack, weplab etc.

Break-in

- the targeted weakspots are routers, proxys, computers and services in a network as well as weak passwords, poor and faulty security mechanisms
- attack scenarios
  1. *host scanning* → which computer / router / proxy exist in close proximity of the target (broadcasts, routing list, traffic, sniffing, DNSpredict/Google)? → list of target systems
  2. *scanning the target system* → type of systems (by means of fingerprints, traffic analysis, Google, whois, etc.) which services (IP/TCP/UDP) are available or vulnerable (portscanning & ICMP etc)
  3. *attack* → exploiting bugs, backdoors, exploits, password scanners/lists, dropper, Google-HackingDB
  4. *successfull breach* → read password lists, install droppers, backdoors, keyloggers, proxy monitor, rootkit, etc
     - start attacks from the compromised system
     - remove traces
- examples:
  - GHDB → default SSID and passwords of WIFI routers
  - NBTEnum → search for other Windows systems
  - Network Monitors → traffic analysis (eg TTL field observations) with respect to transparent bridges or dangers arising from IDS (not to attract attention)
- break-in via exploits for example toolkits, known exploits, zero day exploits

3 Attacks on Data / Protocols

Attacks on data / protocols via

- communication interception
- information manipulation
- attacks on protocols and core mechanism

Focus on

- protocol weaknesses
- (lack of) communication weaknesses
- focus on manipulating algorithms and protocols (eg via "contributions" to open source projects)

Examples:

3.0.1 Address Resolution Protocol

- **weak spot** of the ARP is that it is a stateless protocol and therefore it is possible to send ARP-Replies without any requests
- **ARP-Spoofing** (ARP Request Poisoning, ARP Poison Routing)
  - sniffing = collecting network information
  - poisoning = targeted sending of wrong ARP packets (ARP-Reply with MAC adress for a foreign IP adress) to caches
  - data packets will now be sent to attacker (address in the cache) which manipulates/spies on the data packets before they are sent to their real destination → this faked association enables Man-in-the-Middle attacks
- there are various tools to simplify attacks eg ARP Video

3.0.2 Internet Protocol

- **weak spot** of the IP is that IP-packets are not protected
- **attack possibilities**
  - reading IP-packets is simple
  - checksums for integrity checking are not safe
  - no protection of IP-PCI (IP Header) → manipulation of the protocol header is simple
  - liability is unsafe because authenticity of addresses is not provable
- **attack scenario**
  - target system is protected by IP-sender adresses (meaning that only systems with registered IP addresses are alllowed to use the target system)
  - sniffing: spying of systems that exchange data with the target system (can also be encrypted)
  - connecting to the target system using spied out IP addresses

3.0.3 Transmission Control Protocol (TCP)

- **weak spot** of TCP is that a large number of ACK messages leads to high load on the firewall control
  - ACK = acknowledgement; TCP is an acknowledgement-based protocol; when computers communicate via TCP, received packets are acknowledged by sending back a packet with ACK bit set
  - some firewalls check incoming home network internet traffic insufficiently
  - verification only for SYN messages, ACK messages are all let through
    - SYN = synchronize message via which a client requests a connection from the server
    - part of the TCP three-way handshake (SYN → SYN-ACK → ACK)
- **attack possibilities**
  - incorrect ACKS are used to implement exploits (rather unproblematic)
  - ACK-Tunneling = ACK is used for data transport → Trojan/Dropper acts as an ACK server and reads fata from the ACK (problematic)
  - SYN flood
- **attack scenario**
  - intrusion into the target system and installation of an ACK server, which acts as a remote shell, or dropper, etc
  - target system can now be controlled remotely (until replacement by a better firewall)

3.1 Web-based attacks

3.1.1 SQL Injection

- **weak spot**: web applications that use databases and without properly sanatizing etc
- **attack possibilities**
  - transfer of input data to the database (Form, URL) in a way to spy, change, delete data and execute code

3.1.2 XSS - Cross Site Scripting

- **weak spot**
  - possibility of executing script code in the browser
  - weak user input checks
- **attack possibilities**
  - identify weak spots in web applications (eg possible user input via URL) that allow execution of script code → construct URL with script code
  - other variants possible: <img>, <iframe>, etc and send those to potential targets (spam)
- **attack scenario**
  - URL queries cookies and sends those to a script → script calls the current application with the stolen cookies and uses the application under false idenity (session hijacking)

3.1.3 Cross Site Request Forgery (CSRF)

- exploiting the functionality of a web applications where victims have accounts
- submit manipulated HTTP requests
  - embed link or images in e-mails
  - cross-site scripting
  - malware

4 Attacks by Communication Partner

Attacks of the communication partner by

- faking identities
- trust abuse
- attacks on the data
  - listening to the data (sniffing)
  - manipulating data
  - decrypting protected data

Focus on

- misuse of trust, eg social engineering

5 Web-based Attacks: GHDB

- exploits are known and possibly even the corresponding targets, tanks to search engines such as the Google Hacking Database or other databases where there are plenty of where attackers might get user ids, passwords and other identity properties from

6 Social Engineering

- phone
  - call the victim or services of the victim
  - example: Apple's password reset - procedure
- trash of the victim (Harddisc, CD, USB-Sticks)
  - searching for sensitive data
  - lots of examples exists in the media
- confidence tricks
  - all kinds of scams
- online databases
  - social sites → check news about victim at typical user sites
- U3-USB-Stick

Not so much related to rest of lecture:

6.0.1 OWASP

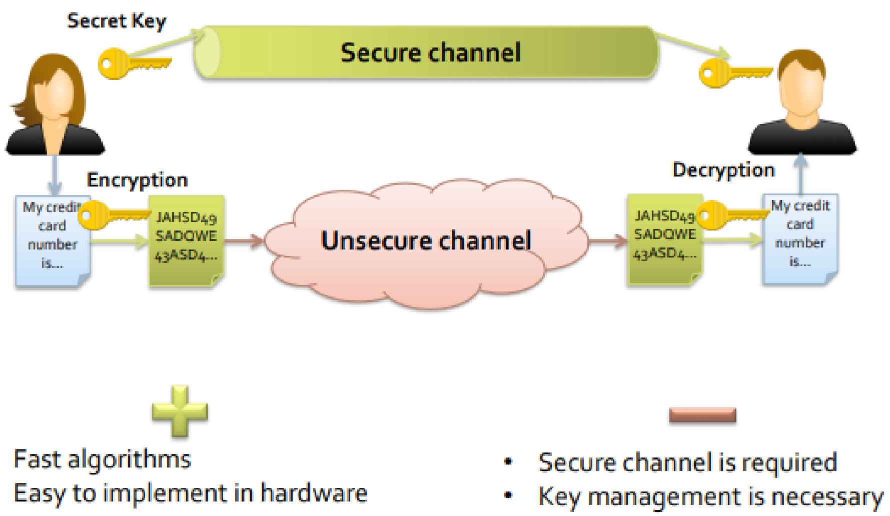
The Open Web Application Security Project is a worldwide not-for-profit charitable organization focusing on improving the security of software, which issues software tools and knowledge-based documentation on application security

7 Security Mechanisms for Distributed Software

7.1 Cryptography

Cryptography is a broad field, which is only briefly touched in this lecture. The methods we'll use in this lecure are:

- one key (symmetric algorithms)

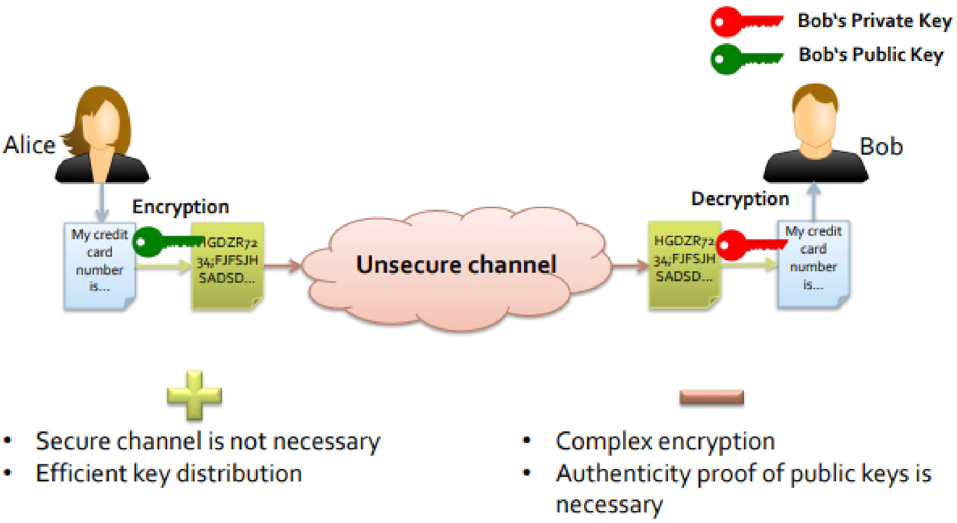
  - 

    +
    - Fast algorithms
    - Easy to implement in hardware

    –
    - Secure channel is required
    - Key management is necessary

  - both participants use the same key (for de- and encryption)
  - the key therefore has to be transmitted aswell (risk)

- two keys (asymmetric algorithms)

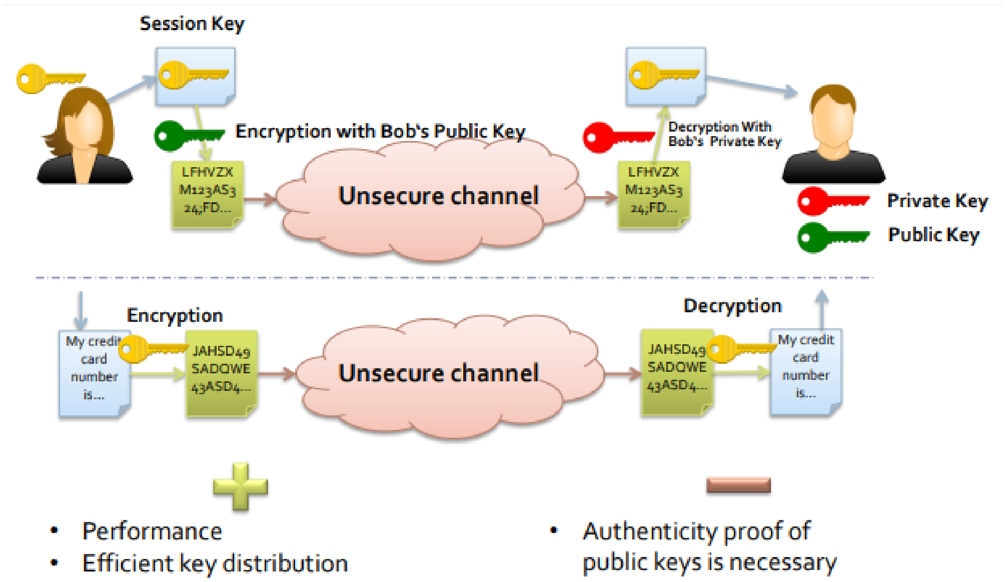
  - 
Bob's Private Key
Bob's Public Key
Alice
Bob
Encryption
My credit card number is...
HGDZR72 34;FJFSJH SADSD...
Unsecure channel
Decryption
HGDZR72 34;FJFSJH SADSD...
My credit card number is...

    +
    - Secure channel is not necessary
    - Efficient key distribution

    –
    - Complex encryption
    - Authenticity proof of public keys is necessary

  - a public key is used to encrypt a message which can only be decrypted with the according private key → private key is not submitted (thus more secure)

- hybrid methods

  - 
Session Key
Encryption with Bob's Public Key
LFHVZX M123AS3 24,FD...
Unsecure channel
LFHVZX M123AS3 24,FD...
Decryption With Bob's Private Key
Private Key
Public Key
Encryption
My credit card number is...
JAHSD49 SADOWE 43ASD4...
Unsecure channel
JAHSD49 SADOWE 43ASD4...
Decryption
My credit card number is...
• Performance
• Efficient key distribution
• Authenticity proof of public keys is necessary

  - session key is encrypted with public key and transmitted and then gets decrypted with private key
  - session key is used to encrypt data/message and now the receiver can decrypt it with the earlier decrypted session key

- one-way hash functions

  - **compression**
    - ∗ inputs of arbitrary length are mapped to outputs with fixed length
  - **irreversibility** (surjective function)
    - ∗ input can not be inferred from the output
  - **collision-resistant**
    - ∗ a hash function $h()$ is called collision resistant - if it is hard to find to find two inputs a and b such that $h(a) = h(b)$ and $a \neq b$

Public key cryptography visualized
Challenge-Response with Public Key:

1. Client sends identifier ID to server

2. Server sends generated random number R

3. Client signs R with a private key & sends the result

4. Server verifies the result using the public key of the client

7.2 Public Key Infrastructure (PKI)

- Challenge: management of public keys
- binding the key to its owner
  - certificate = digital certificate of public key assignment to a (legal) person (eg X.509 Certificate)
  - certification authority (CA) = provides certificate issugin services; the certificates are usually signed with the private key of the CA
    - reduces the problem of authentic key distribution to distribution of authentic keys of CAs
  - service users must identify themselves to the CA

CA services require the use of a computer which is suitably protected against improper use. In particular, it is recommended to use a computer without any network connection to protect it physically.
The secret keys of the CA must be adequately protected and may not be given to third parties. The responsibility lies with the administrators of the CA, who are, therefore, advised to use external peripheral devices (eg smart card, floppy disk).
The secret signature key of the CA must only be used to sign CA- or Enduser keys or revocation lists (CRLs) or to create cross-signed certificates.
Each CA must generate its asymmetric key pairs by themselves. Asymmetric key pairs of the CA for signature generation must have a minimum length of 2048 bits RSA (or equivalent). In case CA generates asymmetric key pairs fo the end user, CA has to perform it on a dedicated CA computer. All data obtained during certification must be treaded as confidential by the CA staff. CA legal data protection regulations are to be complied with.

In summary the operation of a Certificate Authority is mainly influenced by the *technical requirements*, the *legal requirements* and the *organizational requirements*.

8 SSL/TLS

Wiki: Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communications security over a computer network. Several versions of the protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). Websites can use TLS to secure all communications between their servers and web browsers.

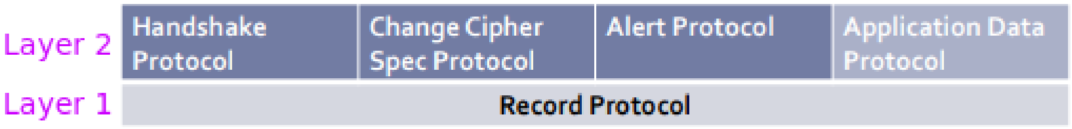
The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications.[2]:3 When secured by TLS, connections between a client (e.g., a web browser) and a server (e.g., wikipedia.org) should have one or more of the following properties:

- The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session (see § TLS handshake). The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted (see § Algorithm below). The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).

- The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).
- The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

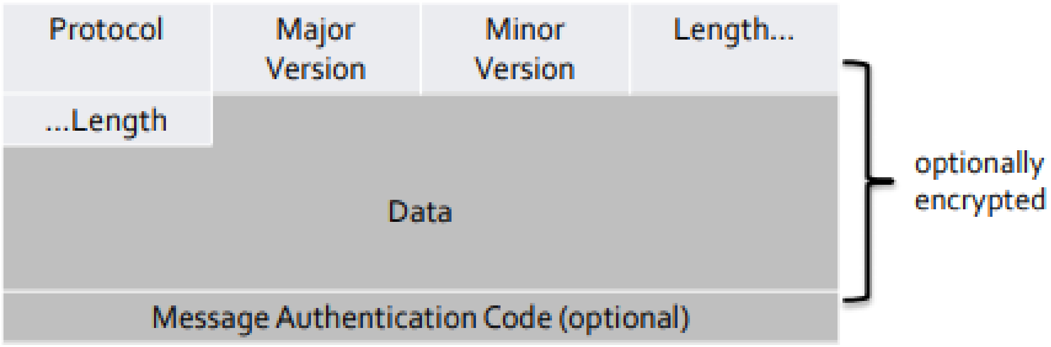
In the OSI-model SSL/TLS belongs in layer 6, in the TCP/IP model it belongs above the transport layer (ie TCP,..) and below the application layer (ie HTTP,..).

The SSL/TLS Architecture basically consists of of 2 protocol layers:



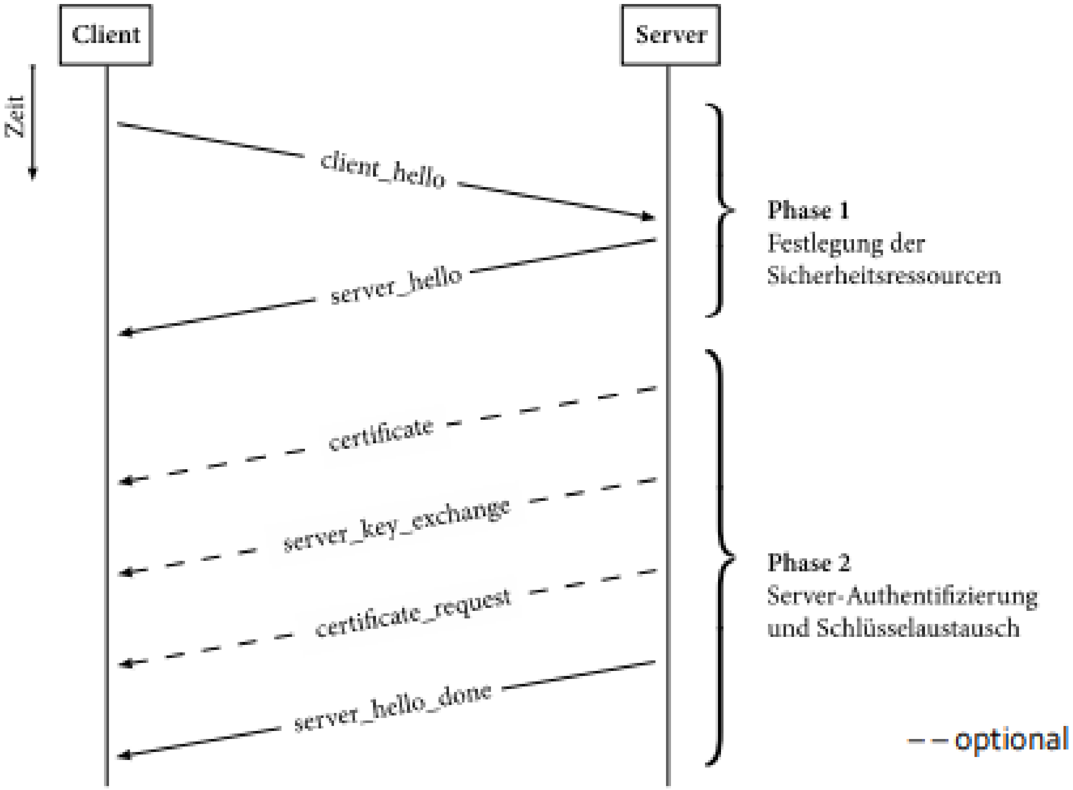
8.1 Record Protocol (Layer 1)

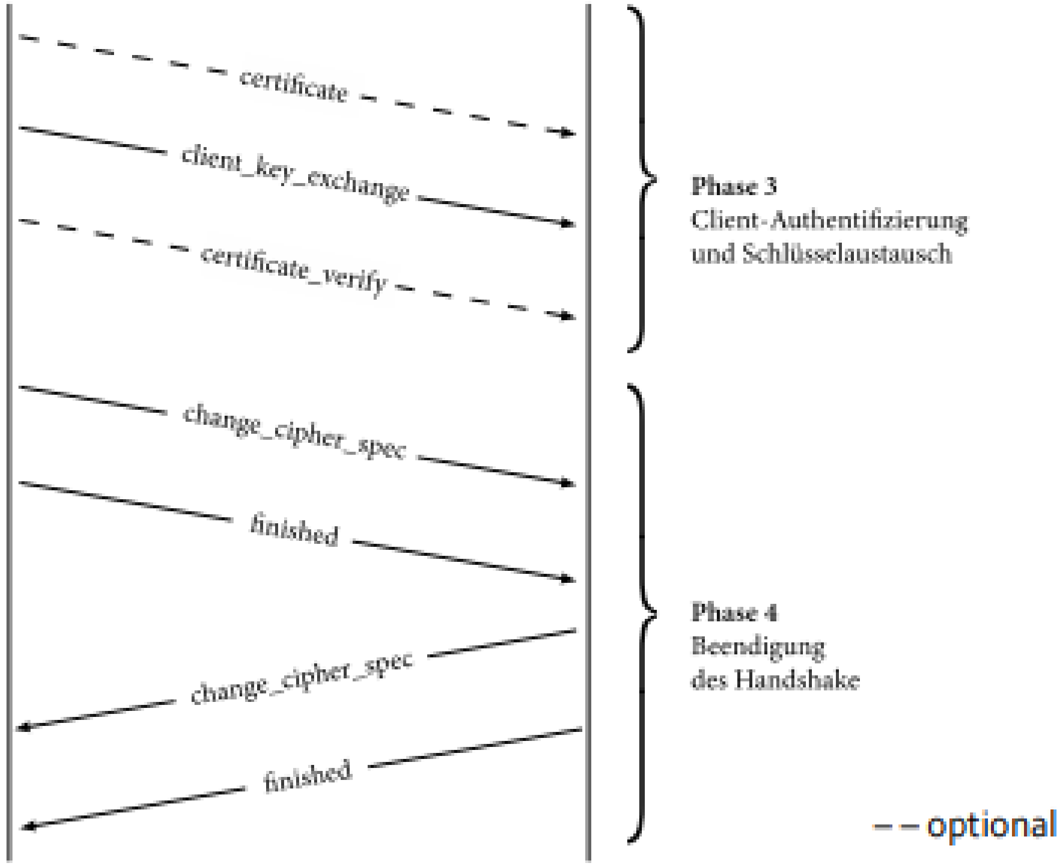
- represents the lower level of the TLS protocol
- encapsulation of exchanged messages
- decomposition into blocks for transmission
- end-to-end encryption
  - symmetric algorithms
  - see the following handshake protocol
- integrity and authenticity are ensured by cryptographic checksums



8.2 Handshake Protocol (Layer 2)

- server and client decide on:
  - mode of encryption
  - type of message authentication
  - secret key
- authentication via certificates is possible





8.3 Change Cipher Spec Protocol (Layer 2)

- change to the negotiated cipher suite
- cipher suite identifies a combination of four algorithms
  - key exchange
  - authentication
  - hash function
  - encryption

8.4 Alert Protocol (Layer 2)

- signaling on error states
- protocol defines two fields:
  - level of error alert
    - warning
    - fatal → connection is immediately interrupted
  - type of error alert
    - detailed error description

8.5 Application Data Protocol (Layer 2)

- pass application data transparently
  - without consideration of its content
- based on security parameters data is…
  - fragmented
  - compressed
  - protected
  - encrypted

8.5.1 Further Reads

- `https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/`
- `https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/`

9 Authentication

9.1 Introduction

Authentication is the process of verficating if someone is the one who he claims to be. There are different kinds of authenticators:

- knowledge-based
  - PIN, passwords
  - Challenge-Response
- biometrics
  - fingerprint, iris, voice, signature, keystroke behavior
- ownership-based
  - something that you do not notice, but what is stored on a medium
  - IDs, magnetic cards, certificates, smart cards
- multi-factor authentication
  - combination of different types of authentication
  - 2 Factors: eg deposit card + PIN, credit card + signature, password + PIN sent by SMS
  - 3 Factors: eg password + smart card + fingerprint

9.1.1 Knowledge-based Authentication

Knowledge-based Authentication using passwords

- Alice agrees with Bob on a secret password p for authentication of Alice to Bob
- Bob applies a one-way or cryptographic hash function H on the password, and stores the image value H(p)
- when authenticating the password is send and then gets hashed again and compared to the stored hash

9.2 Kerberos

Wiki: Kerberos is a computer-network authentication protocol that works on the basis of tickets to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

- works according to the KDC principle
- *User* wants to use a certain service
- *Client* is the local Kerberos application
- *Server* provides the desired service
- *Authentication Server (AS)* is used for primary user authentication
- Ticket Granting Server (TGS) issues tickets for certain services
- KDC includes AS and TGS

9.2.1 Accreditation

- User and his password are provided to the AS
- TGS and its secret key are also accredited by the AS
- Server and its secret key are made known to the TGS

See pages 5 - 12 for more on Kerberos

10 Identity Information in Directory Services

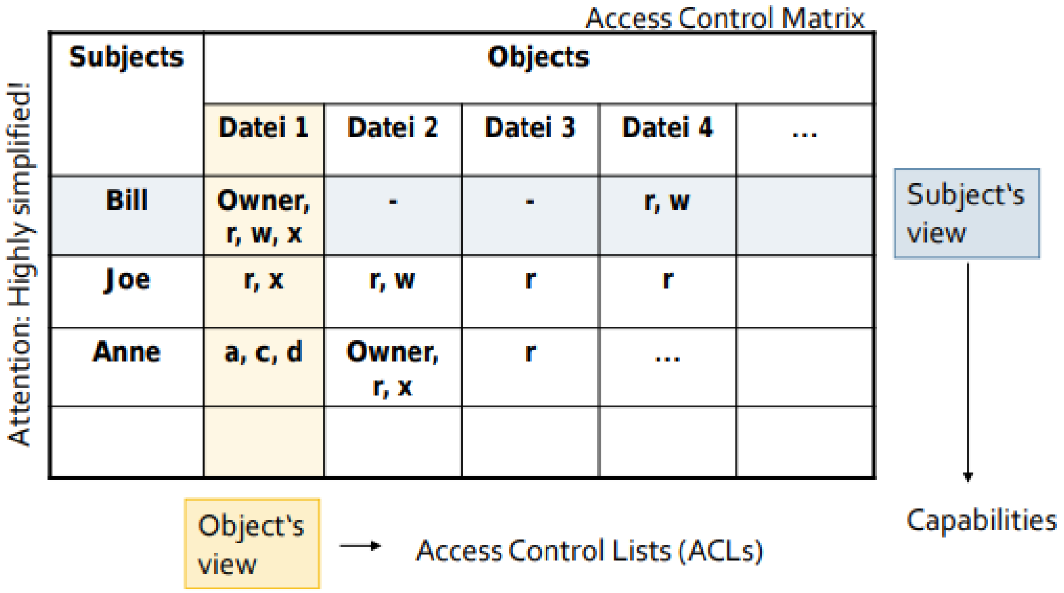
- directory service is a special 'name service'
- property-based requests
  - comparison: full-name DNS request
  - similar to 'Yellow Pages'
- OSI X.500 is the 'classical' Directory Service
  - however the complex 'Directory Access Protocol' (DAP) prevented it from becoming more widespread
- LDAP = Lightweight Directory Access Protocol
  - standardized by IETF

Important: See pages 14 - 23

11 Management of Access Rights

Authorization is the process of verification and access right assignment for a resource/service to a subject and is not to be confused with *Authentication* which is the process of verificating claimed properties. Access Control is a process of access rights management and control.

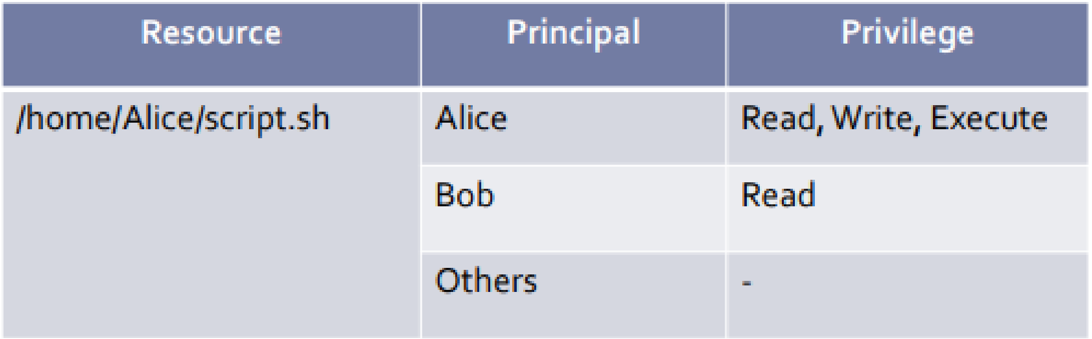
Access Matrix



Subjects	Objects				
	Datei 1	Datei 2	Datei 3	Datei 4	...
Bill	Owner, r, w, x	-	-	r, w	
Joe	r, x	r, w	r	r	
Anne	a, c, d	Owner, r, x	r	...	

- group- and role-based access rights management:
  - complexity reduction by clustering users into ’role groups’
  - inheritance relationships in rights management
  - permissions based on roles

Access Control Lists



Resource	Principal	Privilege
/home/Alice/script.sh	Alice	Read, Write, Execute
	Bob	Read
	Others	-

- *principal* is a user, group or process that can be authenticated
- simply put: ACL is a set/list of resources, principals and corresponding access rights

Access Control Models

- Discretionary Access Control (DAC)
  - access rights are assigned per user
  - owner of a resource can pass his own rights
- Mandatory Access Control (MAC)
  - rights passing is not allowed
  - the system alone decides on which user has access to which resources
- Role-Based Access Control (RBAC)
  - user could potentially be assigned multiple roles
  - access rights are role-based

11.1 Realization in Operating Systems

Unix/Linux

- data/directories are associated to an inode descriptor (contains ID of the owner, ID of the group, ACL etc)
- assignment of rights to the file owner, group, everyone else

Windows 2000/XP/Vista/7

- permissions/restrictions can be assigned to individual users and groups
- security descriptors contain owner-ID, group-ID, Access Control Elements with Allow/-Deny entries, logging operations

12 Internet Firewalls (Chapter 6)

Definition: Firewalls are hard- or software components, which control the interconnection point between two network areas and implement security strategies by restricting packet forwarding.

Fundamentals:

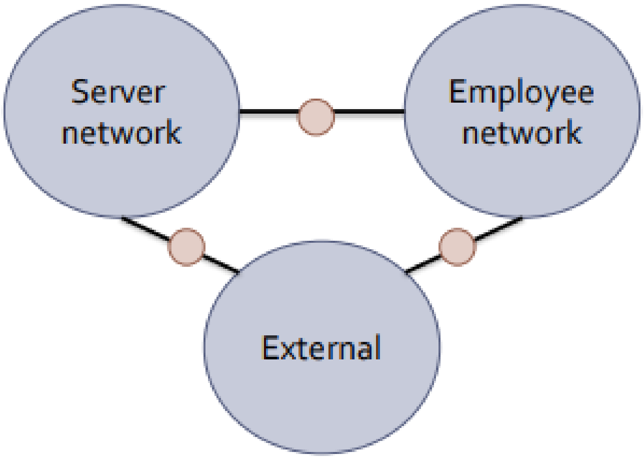
- Packet filter
  - entity, which selectively processes flowing packets according to predefined rules, in particular, preventing packet forwarding
- Proxy approaches
  - representative of a client process
- Network Address Translation (NAT)
  - address translation, public and private addresses are distinguished
- Bastion Host

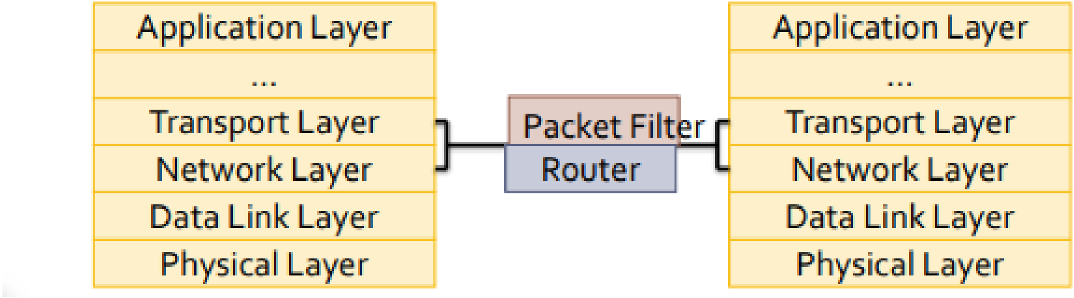
  - computer with particularly high protection requirements; vulnerability mainly results from the computer's exposed location

- Dual-Homed Host
  - computer with at least two network interfaces for two different subnets

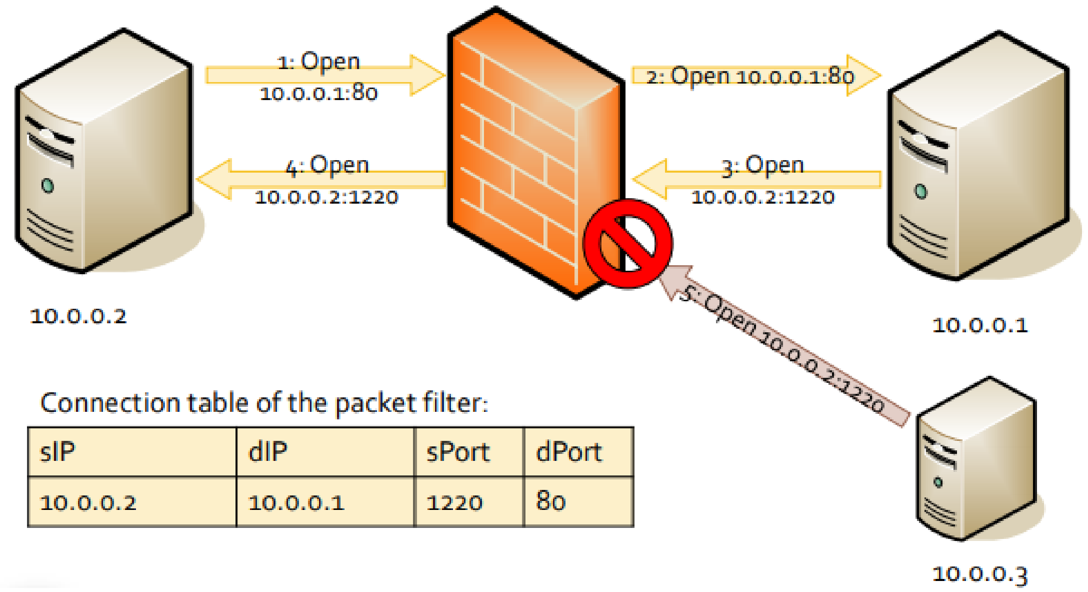
These approaches are now covered in more detail.

12.1 Packet Filter





Then there are also router filter rules for example `deny icmp 129.12.0.0 0.0.255.255 any` in Linux environments this can be done via iptables, ipchains, ipfilter, . . .
There are also dynamic packet filters:



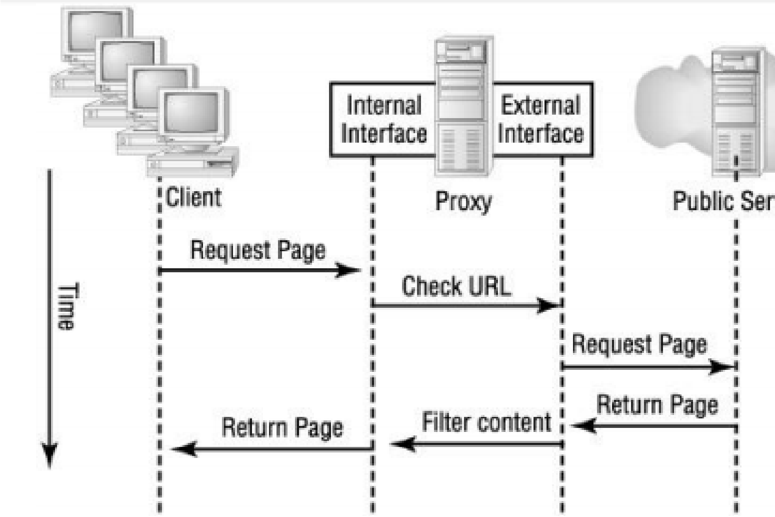
Connection table of the packet filter:

sIP	dIP	sPort	dPort
10.0.0.2	10.0.0.1	1220	80

Filter Table Guidelines

- "default deny" → prohibit everything, which is not explicitly allowed
- order → filter table is usually processed sequentially, analysis is terminated after all the rules have been applied
  - correct order should be maintained
- prevent spoofing attacks
  - packets coming from ’outside’ with ’inside’ addresses are rejected; the same holds true in the other direction if the source address is not an ’inside’ address
- static filters: UDP blocking
- controlled handling of ICMP
- prevent source-routing
- efficiency: unneccessary filtering rules have to be removed

12.2 Proxy Firewall



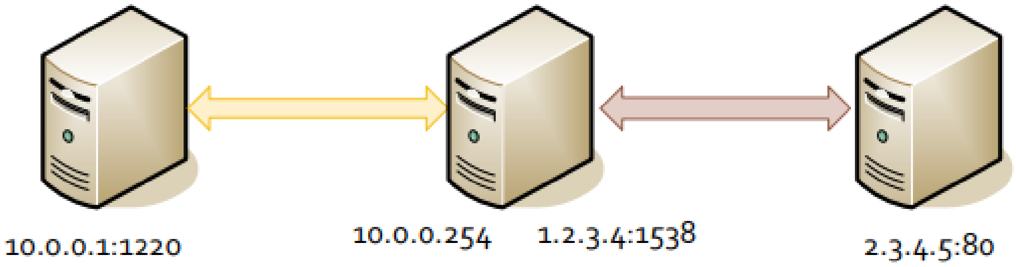
Firewalls 24Seven,
Second Edition by Matthew
Strebe and Charles
Perkins Sybex © 2002

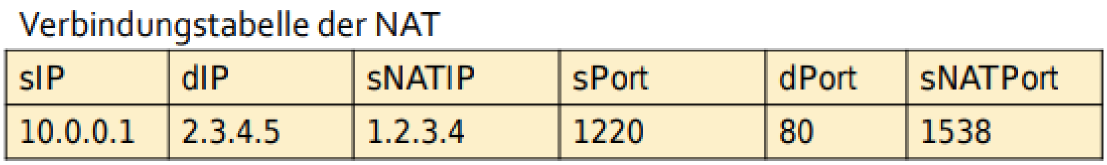
- typically at transport layer or as an application proxy
- transport layer: requires client code modification
- application proxy: can perform service-specific controls

12.3 Network Address Translation

NAT is a proxy concept at the network layer. Initially it was intended as a measure to preserve the IPv4 address space while today it is used to conceal internal network structures.

- doing NAT in practice gives up the end-to-end principle as it leads to numerous diffuclties (eg ftp)



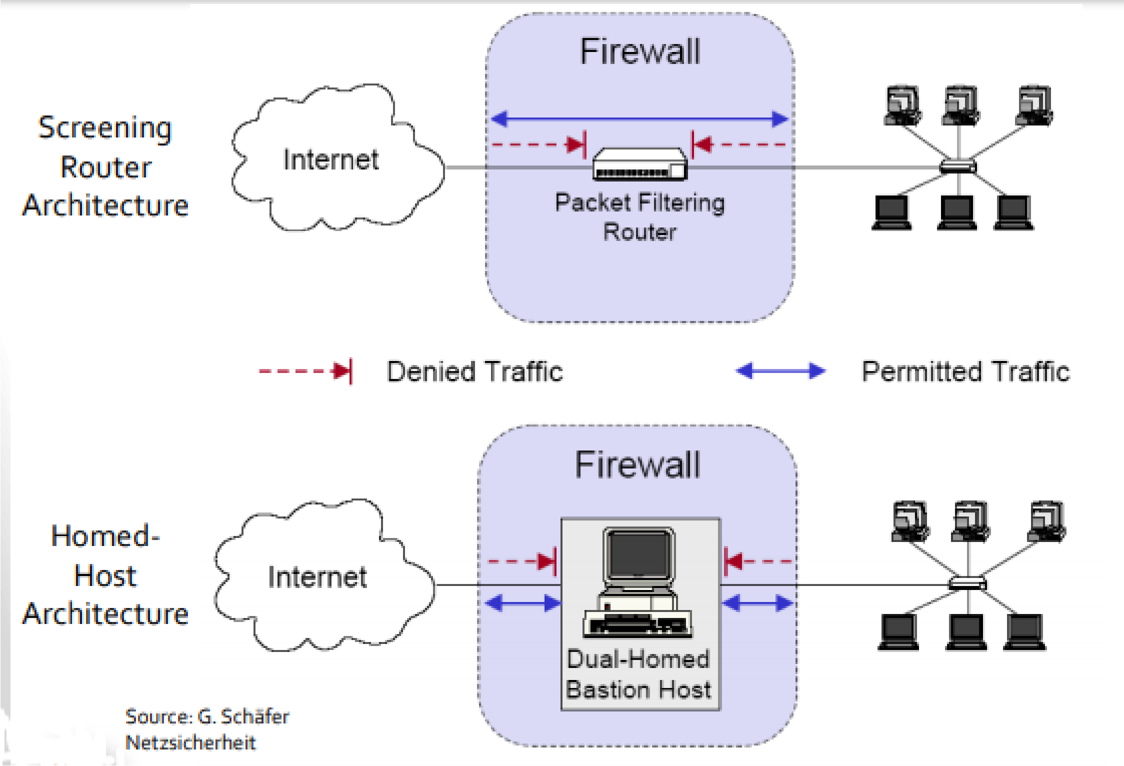


Verbindungstabelle der NAT

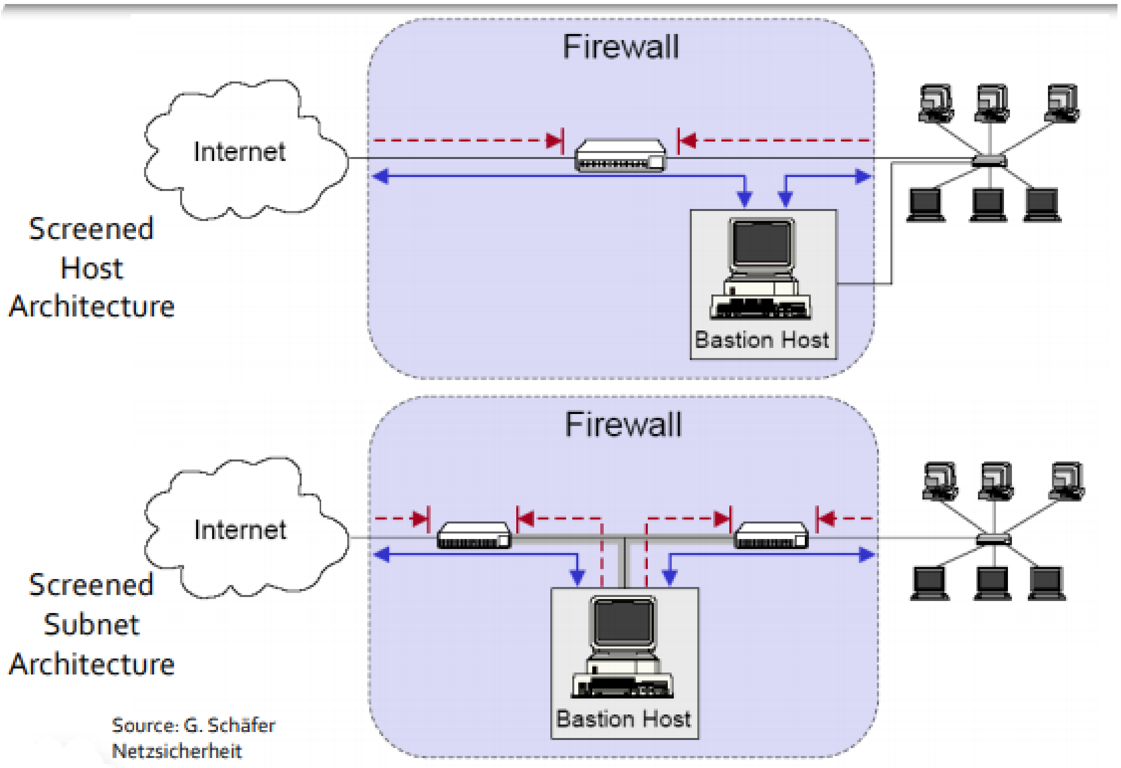
sIP	dIP	sNATIP	sPort	dPort	sNATPort
10.0.0.1	2.3.4.5	1.2.3.4	1220	80	1538

12.4 Architectures

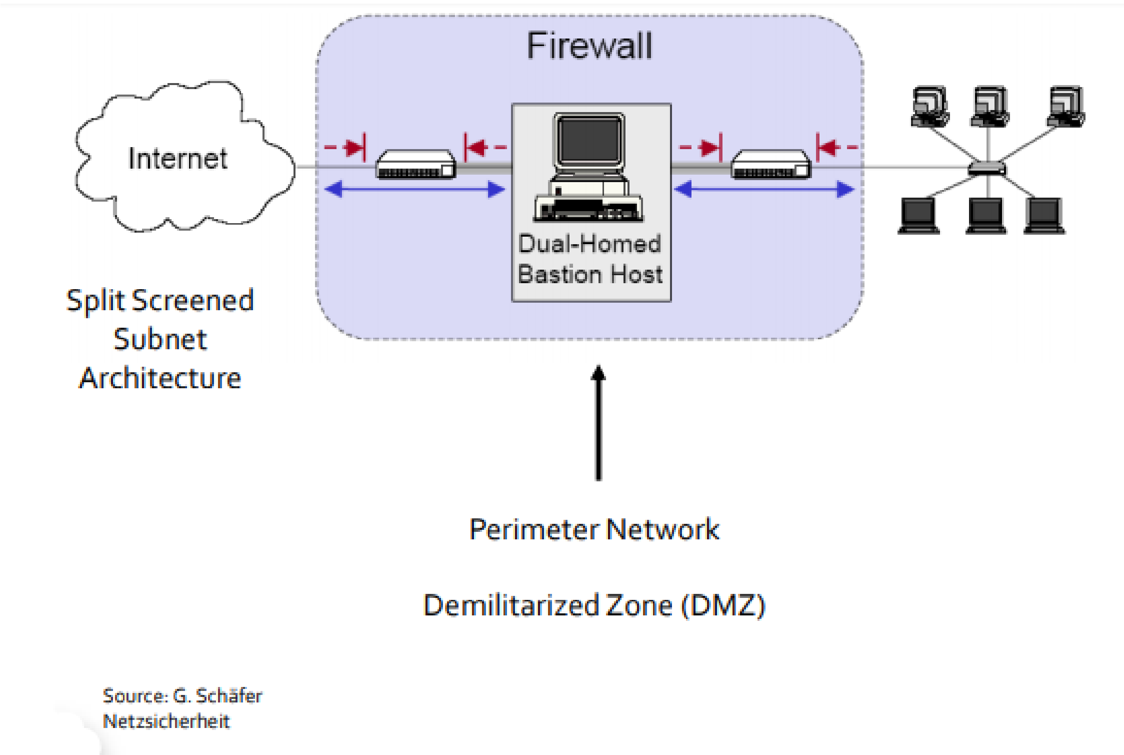
Let's look at different architectures utilizing different kinds of firewalls:



Source: G. Schäfer
Netzsicherheit



Source: G. Schäfer
Netzsicherheit



Split Screened Subnet Architecture

Perimeter Network

Demilitarized Zone (DMZ)

Source: G. Schäfer Netzsicherheit

One traditional conception in network design has been that of the "perimeter" which means that there’s an "inside" and "outside" to our network. However this is not applicable to the modern situation because of eg

- mobile devices
- peer-to-peer systems
- ubiquitous computing
- ad-hoc networks
- sensor networks
- . . .

So there is no clearly defined "perimeter network" available anymore.

12.5 IT-Security Management Aspects

- determination of the required security level
- firewall placement and coordination
  - clear transition between ’internal’ and ’external’?
  - select entrance architecture (dual homed, screened, subnet,..)
  - should the subnets be protected from one another?
  - do devices require ’personal firewalls’?
  - how can the three stages (entrace, subnet, end-system) be kept consistent and checked for errors?

- Analysis of open communication channels
  - dependencies on the first point
  - administration concept: who gets to issue which rules?
- firewall management requires security policy support

13 Intrusion Detection Systems (Chapter 7)

Motivation:

- computer has been compromized and is used for (illegal) data distribution
- network operator performs IP accounting and finds out that a computer, which has previously generated next to no load, is suddenly generating a high amount of it
- Goal: attack detection and intrusion detection alarm

Intrusion Detection Systems

- find and report suspicious activity in systems and networks
- intrusion prevention: initiation of control measures
  - intrusion response

13.1 Classification of Intrusion Detection Systems

Location:

- Host-based
  - system breach and misuse detection
  - examination of log files
  - integrity checks by checksums
  - inspection of "privilege escalation"
- Network-based
  - monitoring and verification of network traffic, which can take place at various network locations
- Hybrid

Detection:

- signature-based
- anomaly-based

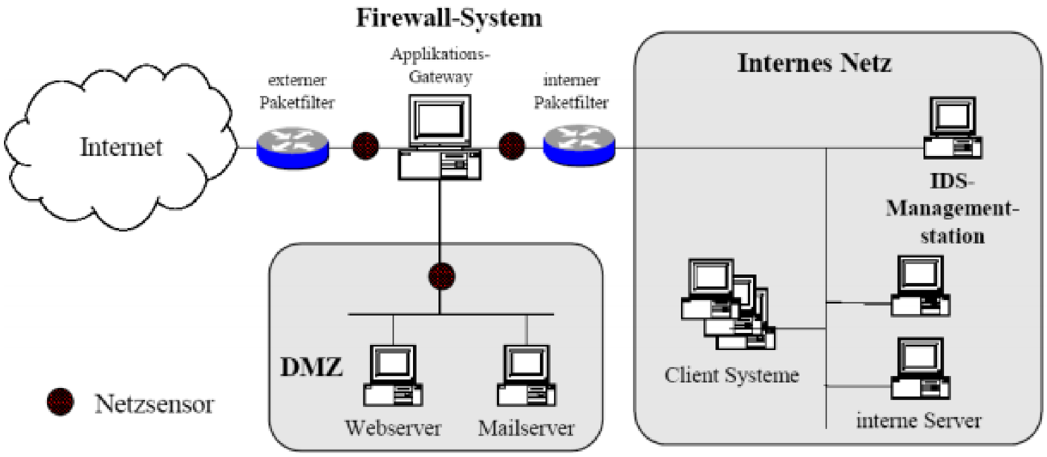
13.1.1 Signature-based Detection

- break-in (attempt) detection based on known procedures
  - eg buffer overflow attack
  - eg implies *default.ida* within a URL in an HTTP packet together with a certain pattern in the URL Argument Name Field is a Code Red attack
- signatures must (same holds true for virus scanners) be kept up-to-date
- challenges:
  - register the attacks
  - describe the attacks
  - errors of type 1 and 2 (classification problem)

13.1.2 Anomaly-based Detection

- detection of "normal" user behaviour deviations
- normal behaviour has to be statically describable
- classification problem
- normal behaviour should be determined through learning
- very effective attacks which are not deviating much from normal user behaviour might remain undetected

1. Example: Securing Gateways



13.2 Intrusion Detection System - Honeypots

Approach:

- place unsecured server/service ("honeypot") in the network
- monitor honeypots
- analyse attacks and compromises
  - identify tools, tactics and intruder motives

Typical objectives:

- detect botnet attacks
  - botnet = network of compromized computers that can be remotely orchestrated by the attacker
- detect phishing attacks

13.3 IDS in IT-Security Management

- Intrusion Detection is a reactive IT-security approach
  - complements preventive measures, such as firewalls
- data protection legal requirements must be met
- intrusion prevention (response): given automatic reactions, one has to make sure they cannot be used as an attack themselves (such as Denial of Service)
- integration with network management is appropriate and necessary

14 Incident Management (Chapter 8)

14.1 History of CERTs / CSIRTs

- CERT = Computer Emergency Response Team
- CIRT = Computer Security Indicent Response Team
- trigger: internet worm 1988
- need of an IT-security 'fire brigage' became evident
- CERT/CC (Coordination Center) was founded by DARPA located at CMU

Today:

- not just 'response', but generally incident handling
- many CERTs and CSIRTs in the world eg DFN-CERT, CERT-Bund
- in Germany: CERT-network
- international network: FIRST (Forum of Incident Response and Security Teams)

14.2 CSIRTs Tasks

Reactive Services

- alerts and warnings
- incident handling
  - incident analysis
  - incident response on site
  - incident response support
  - incident response coordination
- vulnerability handling
  - vulnerability analysis
  - vulnerability response
  - vulnerability response coordination
- artifact handling
  - artifact analysis
  - artifact response
  - artifact response coordination

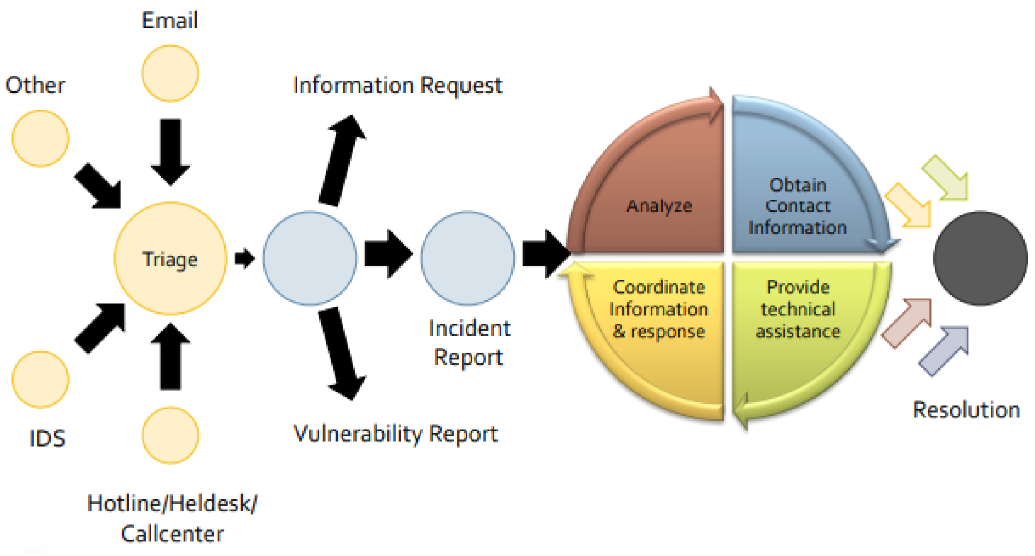
Proactive Services

- announcements
- technology watch
- security audit or assessments
- configuration & maintenance of security tools, applications & infrastructures
- development of security tools
- intrusion detection services
- security related information dissemination

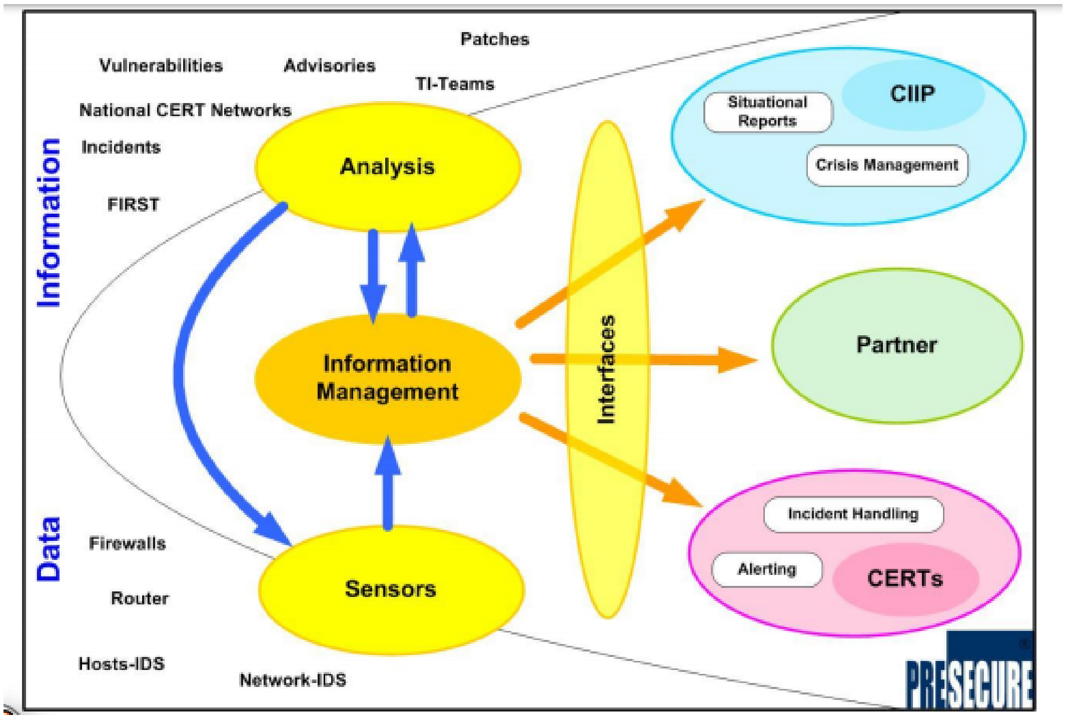
Security Quality Management Services

- risk analysis
- business continuity & disaster recovery planning
- security consulting
- awareness building
- education/training
- product evaluation or certification

1. Incident Handling


Email
Other
Information Request
Triage
Analyze
Obtain Contact Information
Coordinate Information & response
Provide technical assistance
Incident Report
IDS
Vulnerability Report
Resolution
Hotline/Heldesk/ Callcenter

2. Coordination: Early Warning System



14.2.1 Naming of Vulnerabilities

Naming requires standardization

- otherwise cooperation & coordination become complex

Standard: common vulnerabilities and exposures

- managed by The Mitre Corporation

Example:
Name: CVE-2004-0309
Description: Stack-based buffer overflow in the SMTP service support in vsmon.exe in Zone Labs ZoneAlarm before 4.5.538.001, ZoneLabs Integrity client 4.0 before 4.0.146.046, and 4.5 before 4.5.085, allows remote attackers to execute arbitrary code via a long RCPT TO argument.
Status: Entry
Reference: BUGTRAQ:20040219 EEYE: ZoneLabs SMTP Processing Buffer Overflow
Reference: CERT-VN:VU#619982

Part III: Trustworthy Software Engineering

Trustworthy Software

- in Cordis.Europa.Eu security document defined as: Trustworthiness can be seen as software and infrastructure that is secure, reliable and resilient to attacks and operational failures; guaranteeing quality of service; protecting user data; ensuring privacy and providing usable and trusted tools to support the user in his/her security management.
- trustworthiness needs to be considered from the outset rather than being addressed as add-on feature

So we focus on: Identity & Security By Design (SBD)

- who is it for?
- why does it matter?
- what is it all about?
- where does it apply?
- when to apply?
- how to apply?

15 Identity (Chapter 1)

Internet as a danger zone in terms of identity

- what exactly needs to be protected?
- what should one orient towards?
- which data is exceptionally worthy of protection?

Security vs Identity

- for starters: keynote by Dick Hardt at WWW 2007 on "Identity 2.0"
- speech on identity by Kim Cameron

15.1 Identity - Problem

- Kim Cameron: "The internet was built without a way to know who and what you are connecting to."
- initial situation:
  - internet services are left on their own
    - must provide security → isolated identity solutions
  - criminalization of internet
    - leads to loss of internet's credibility, for example drawback for e-businesses
  - identity layers are complex
    - successful attemps, such as SSL and Kerberos - however overall too many different scenarios are required, so agreement is difficult

Possible solution: Identity Metasystem

- such a system provides confidential support to ensure who is connecting to whom/what on the internet
- many questions:
  - who holds the data?
  - who trusts whom?
  - what scales?
  - how does one realize openness to new developments that do not yet exist?

15.2 Identity - Identity in Detail

- there are numerous definitions of identity, the lecture is based on Kim Cameron's definition: "*Digital identity is a set of **claims**, which are made by a **digital subject** about self or other subjects.*"
  - digital subject = person or thing (referred or real) in a digital realm that is described or with which one is dealing
    - "with which one is dealing" = often in the context with request/response model
    - example digital subject: real persons, devices, resources, rules/policies and relationships between digitial subjects
    - discussions of the 'subject' term extend into philosophy
  - claim = claim suggests that something is true, typically something that seems to be controversial or questionable
    - remark: claim is a relationship between a certain instance, a digital subject and an identity attribute

We must be able to **structure our understanding** of digital identity:

- we need a way to avoid returning to the **Empty Page** every time we talk about digital identity
- we need to inform peoples' thinking by teasing apart the factors and dynamics explaining the successes and failures of identity systems since the 1970s

- we need to develop hypotheses - resulting from observation - that are testable and can be disproved
- our goals must be pragmatic, bounding our inquiry, with the aim of defining the characteristics of an unifying identity metasystem
- the "**Laws of Identity**" offer a good way to express this thought
- beyond mere conversation, the Blogosphere offers us **a crucible**
  - the concept has been to employ this crucible to *harden and deepen the laws*

These definitions embrace Kerberos, X.509, SAML. They take this problem of the evaluation of the usefulness of a digital identity up to a higher level in the systems sense of multiple layers. These definitions separate the layer of where stuff is communicated from the layer where evaluations are done – a very important step forward.

15.3 Identity - Laws of Identity

1. **User Control and Consent**
   - *digital identity systems must only reveal information identifying a user with the user's consent*
     - systems need to appeal in their convenience & simplicity
     - constantly care about users' confidence
       - requires holistic commitment
       - user must be cetral to control with respect to which identities are used and which data is made public
       - system must protect from deception (eg website location and missue)
       - system must inform the user of possible consequences upon certain action (data sharing, login etc)
       - the holistic approach must be used as a paradigm in all contexts (eg when logging into a company or a private blog it should always be clear that the user consents to the release of certain)
2. **Minimal Disclosure for a Constrained Use**
   - *the solution that discloses the least identifying information and best limits its use is the most stable long term solution*
     - one should assume that data/information violations are unavoidable
     - to reduce risks, information use should be checked with respect to 2 strategies: "must be obtained" or "must be saved"
     - less information implies less value implies less risk
     - "as little as possible identification information" means:
       - reduction of linkable information
       - use of claim transformations
     - avoid unnecessary information storage for "possible future" use (why should a credit card be stored by the shop?)
     - the law is closely related to information disasters
3. **Justifiable Parties**

- *Digital identity systems must limit disclosure of identifying information to parties having a necessary and justifiable place in a given identity relationship*
  - user has to have a clear understanding of whom the information is/will be exchanged with
  - system itself may not draw conclusions about relationships between subject and parties (eg Microsoft Passport is useful for logging into MSN but why should it know if I login to Google or eBay?)
  - in which situations are regulatory identities required?
  - same holds for intermediaries (what should they know to achieve their goal)
  - all participants must submit statements of how the information will be used

4. **Directed Identity**

   - /a unifying identity metasystem must support both "omni-directional" identifiers for public entities and "unidirectional" identifiers for private entities
     - digital identity should always be viewed in the context of another identity or a set of identities
     - omni-directional = public entities require "beacons" (publicly known identifier or URI) → eg websites (URLs) or public devices
     - uni-directional = private entities (people) require an ability not to be turned into a beacon
       - they require a unidirectional identifier, which can be used in combination with a trusted beacon (no correlation, eg user-bank interaction)
       - negative examples: bluetooth and RFID, partially WLAN

5. **Pluralism of Operators and Technologies**

   - *a unifying identity metasystem must channel and enable the inter-working of multiple identity technologies run by multiple identity providers*
     - system may be ideal with respect to one characteristic, but not with respect to another
     - example: Authority vs Employer vs Individual
     - old and new technologies must be used and co-exist; identity system must not be in competition with technology, but must use it
     - technologies may have more growth than others (identity ecology)

6. **Human Integration**

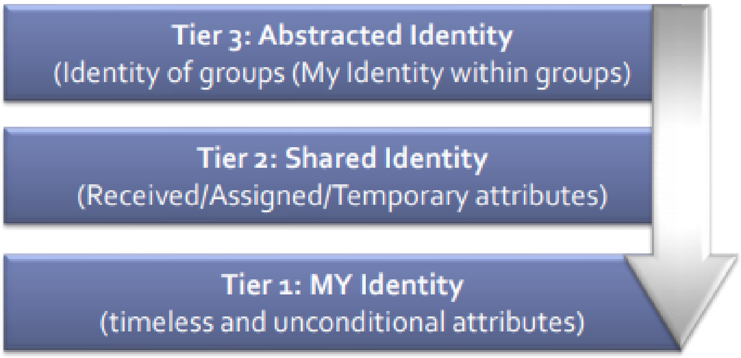
   - *a unifying identity metasystem must define the human user as a component integrated through protected and unambiguous human-machine communications*
     - communication can be completely secure but what about the last two meters (off the screen and into the eyes of the viewer); does the user really know who it is he's communicating with?
       - phishing attacks are a good example of this
     - protocol for use of safety issues has to become a ceremony, absolutely predictable and controlled
       - example: communication in the cockpit (channel 9 on United Airlines)

7. **Consistent Experience Across Contexts**

    - *a unifying identity metasystem must provide a simple consistent experience while enabling separation of contexts through multiple operators and technologies*
        - simplicity and clarity are the main goal - identities have to be used in a similar fashion to all other things on the desktop
            * user must be able to see, verify, add and remove identities
        - which type of identity is acceptable in which context?
            * properties of such candidates are defined by the using parties
            * users must be able to recover the identity in the given context and understand which information is associated with it
            * person (human/legal) could possibly accept different types of identities
            * user must be able to choose the best identity in his opinion

Part III: Trustworthy Software Engineering

16 Identity in the Light of Privacy, Security and Trust (Chapter 2)

- 7 Laws of Identity define requirements of dealing with identities
    - first focus on conceptual/basic understanding
- identity in global context has to comply with different levels
    - layered approach of identity management:
    - 
Tier 3: Abstracted Identity
(Identity of groups (My Identity within groups)
Tier 2: Shared Identity
(Received/Assigned/Temporary attributes)
Tier 1: MY Identity
(timeless and unconditional attributes)

Based on
"Digital Identity",
Phillip J. Windley
and Ping Identity Corp.

16.1 Identity - Security - Privacy

- identity (in a digital setting) is often "only" closely linked to security, identity is more!
    - security = protect data from unauthorized access, removal, tampering
    - privacy = protect attributes, preferences, etc which are associated with identity, against unnecessary use by subject
    - identity is in relation to others → attributes realize trust relation ships

16.2 Identity & Trust

- Trust (wiki) = In a social context, trust has several connotations. Definitions of trust typically refer to a situation characterised by the following aspects: One party (trustor) is willing to rely on the actions of another party (trustee); the situation is directed to the future. In addition, the trustor (voluntarily or forcedly) abandons control over the actions performed by the trustee. As a consequence, the trustor is uncertain about the outcome of the other's actions; he can only develop and evaluate expectations. The uncertainty involves the risk of failure or harm to the trustor if the trustee will not behave as desired
- **Trust = Conviction and belief in the sincerity, honesty and good intentions of another party with respect to a risk-prone action.**

16.2.1 Trust examples

- shopping with credit card, which trust relationships & risks exist?
  - identity and Credit Card company
  - identity and service
  - identity/service and card register
  - identity/service and money
- → trust is always associated with risk
- trust is something one connects to a person
  - one cannot enforce trust by another person

16.2.2 Trust properties

- trust is rarely transitive
  - example: I trust Sara's taste in music, she, in turn, trusts Peter's - therefore I would possibly trust Peter in selecting music for my Birthday Party
- trust cannot be shared
  - example: A trusts B and C, which does not imply that B and C trust each other
- trust is not symmetric
  - example: if I trust you, you don't necessarily trust me in return
- trustworthiness cannot be self-declared ("trust me!")
- trust is a value closely related to evidence
  - buying a brand computer which is more expensive because I trust the brand
  - Computer allows access upon login, since the provided evidence (login/password) serve as proof
- trust is hard to quantify
  - I trust Sara more than Peter - what does that mean?

  - in business context trust can be evaluated against risks (given obvious risk levels)
  - otherwise a contract is used as a basis: analysis is required, risks are evaluated and thereby cotractual relationships are defined; leads to Service Level Agreements (SLA) providers and users

- Trust by reputation
  - trust in a person can develop from other people's statements about him/her (Communities of Trust)
  - examples:
    * all security experts advise caution when traveling in the following countries
    * ebay: one buys a product from a handler he doesnt know, but which has good reviews (high reputation)

16.3 Identity & Privacy

- privacy is an important and complicated topic (tightly coupled with data protection)
- identity and privacy are closely related
  - what does privacy mean for a person?
    * generally: private data shouldn't become public
    * however, often: private data disclosure is ok if it yields considerable benefits
  - privcacy must be observed in context
    * eg discount systems: provide us your address and date of birth and we'll give you a 15% discount
- privacy is partially legally regulated
  - eg Federal Data Protection Act, European Data Protection Directive, Patriot Act
- conclusion in legal context: own applications systems must take identity and privacy into account (see Laws of Identity)
  - embed the concepts of identity and privacy in design
  - use of identity and privacy-relevant information must be comprehensible, verifiable and reportable at any point in time
  - identity management system or an identity metasystem must be able to answer questions on identity privacy terms
  - legal requirements forcte system operators to testify on privacy policy
    * eg webshop sends cookies to customers, what should the privacy policy say? (eg We use cookies)
- privacy principle - respect privacy
  - accountability
  - identifying purposes
  - requirement of affected person's consent

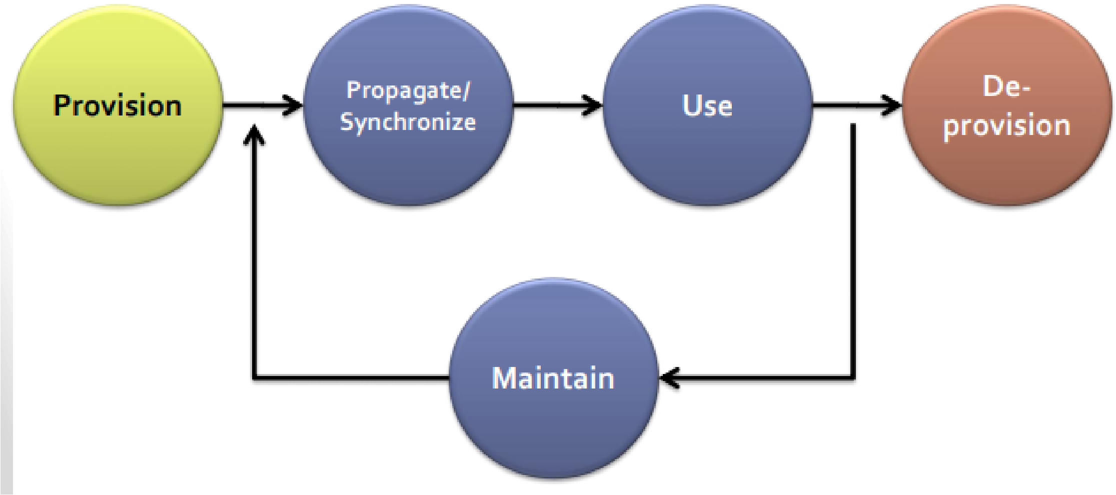
  - minimal privacy data collection (time limit)
  - limitations of use
  - data collection accuracy
  - protection
  - access to personal data (to the owner)
  - comprehensible regulations

17 Identity Management Systems (Chapter 3)

- what is needed for identity implemenation?
  - some kind of **identity metasystem** → contains 3 certain roles (can be more)
  - **identity provider**
    - person or an organization, which creates digital identities, either for themselves or on behalf, eg online shop could create identities for customers, authorities provide identities for their employees
  - **relying party** (human/legal person)
    - person or organization, which requires digital identity before allowing entry/acess
    - example: users willing to revoke a contract - the relying party defines which claims are required in order to execute cancellation, as well as which formats and credentials are accepted
  - **digital subject**
    - individual or entity for which claims are made

Definition **Identity Management**: "Identity management is the set of processes, tools and social contracts surrounding the creation, maintenance and termination of a digital Identity for people or, more generally, for systems and services to enable secure access to an expanding set of systems and applications."

Identity Management Lifecycle



There are three identity management levels:

- personal identity management
- organization-related identity management
- federated identity mangament

17.0.1 Personal Identity Management

Entity-perspective

- Management of different identities (different accounts for different systems)
- Management and control of which information is provided to a service (z.B. Email, phone number etc.)
- eg MS Passport, MS Cardspace

17.0.2 Organization-related Identity Management

Organizational perspective

- Management of identities of an organization
- Different services of an organization are provided with and updated by identity information.
- Traceability of data flows and data accesses
- Management of privileges and roles within the organization
- Definition of organization‘s policies as to the entities i.e. which data can be accessed
- eg SUN Identity Management Suite (SUN Identity Manager), Microsoft Identity Integration Server, IBM Tivoli Identity Manager

17.0.3 Federated identity Management

What is meant by “Federation“?

- “Federation is an association of independent organizational units, which have

a trust relationship.“

- Among the latest developements in the field of IdM.
- Is driven both by the state and industry
  - Common and simplified resource access
  - Complex problems/business processes and a high level of specialization require cooperation.
  - Harmonization of business pocesses
  - Cost savings with respect to administration and resource use
- Frequently used technologies
  - SAML (Security Assertion Markup Language)

    - XML (Schema, Encryption, Signature etc.)
    - Web Service interfaces

- Federation perspective
    - association of organizational units, organizations or even nations
    - Shared use of resources and services of Federation partners
    - Cross-organizational business processes within the Federation
    - Modeling and definition of trust relationships
    - Federative services are then made available according to the defined trust relationships providing ease of access to resources/data (i.e. Single Sign-on)

- Example projects/products/approaches: Liberty Alliance Projekt (SAML 2.0) , WS-Federation specification , SUN Identity Management Suite (SUN Federation/Access Manager) , Ping-ID, PingFederate , Shibboleth , FOAF+SSL

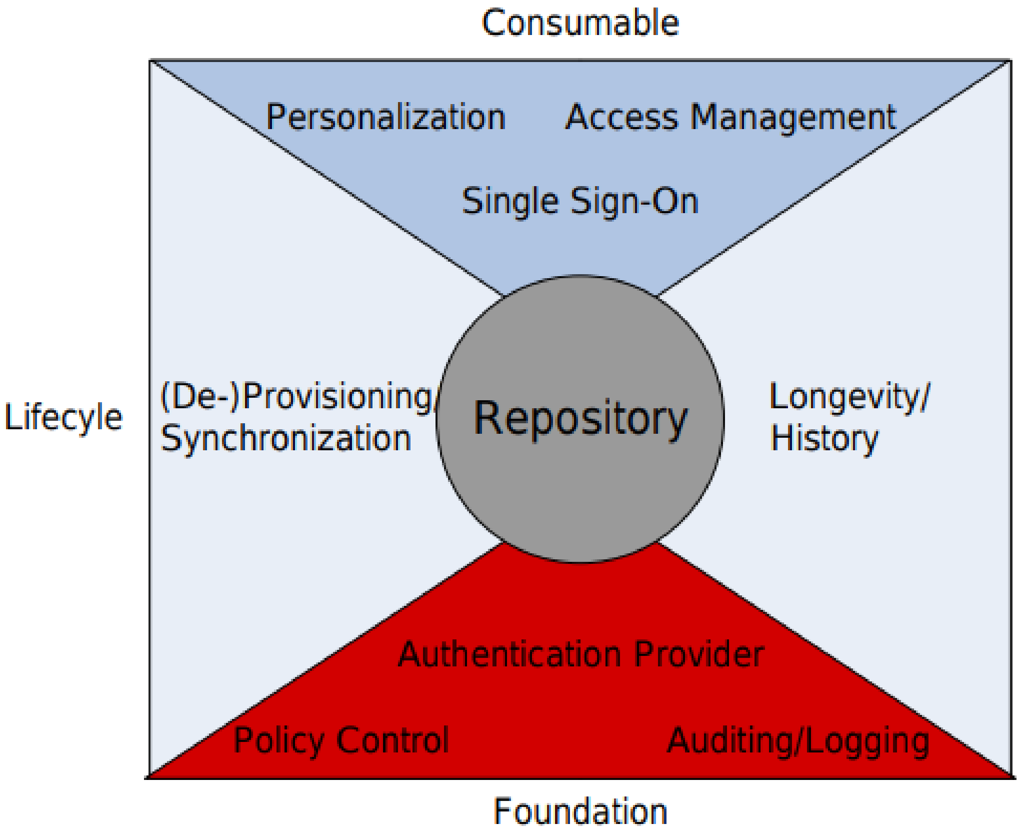
17.1 Anticipitated Benefits of IDMS

Reduced management overhead

- Better optimization/automatization of business processes
- Reduced time required for providing a new employee with access rights to resources
- Reduced risk of a former employee accessing resources
- Policy and legal requirement compliance support (privacy)
- Data consistency (data matching, modification checks, . . .)
- Standard interfaces (APIs, standards . . .) to data/services/resources

17.2 Components of IDM systems

"he focus of identity management is on user provisioning — the creation, maintenance, and termination of user accounts and management of credentials in support of authentication and access control." (HurwitzGroup, 2001)



17.2.1 Basic Components

- **Repository**
  - Repository represents the core component for many identity management systems
  - It is a **logical data storage** (i.e. database, directory service), in which identity information, guidelines and other organization information can be stored
- **Propagation**
  - Depending on the system in use, an identity entry could need to be transferred from the current reposiroty to another one
- **Authentication Provider/Identity Provider**
  - is responsible for primary identity authentication
  - often issues a credential, which can be used for further authentication and authorization (z.B. SAML Token)
  - provides multiple interfaces (z.B. LDAP, Kerberos), by means of which service can perform authentication
- **Policy Control**
  - policy control governs rules of information usage, disclosure and logging
  - authorization policies determine which identity can access and manipualte which information

    - policy control monitors the defined guidelines, creates events to be audited and signalled of according to certain rules (for example, security warnings)

- **Auditing, Monitoring**
    - auditing provides necessary mechanisms for information detection and storage
    - that information normally contains access protocols and data operations (specifically in the repository)
    - if form a basis for tracking whether the policies are being adhered to and is used for subsequent security checks

17.2.2 Lifecycle Components

- **(De-)Provisioning and propagation/synchronization**
    - applies automation of all the procedures and tools to manage the identity lifecycle
    - this Lifecycle is split into initial provosioning, synchronization and de-provisioning phases
    - in the initial provisioning phase the according service is supplied with the necessary identity information such that the new identity can use the service (provisioning process)
    - in the synchronzsation phase identity information is updated and compared between services (synchronization and propagation process)
    - in the de-provisioning phase all the identity information is removed (de-provisioning process)
- **History, Longevity**
    - History and longevity tools create historical records, by means of which one can examine evolution of an identity overtime (i.e. creation, activation, locking, new status, removal)
    - these components provide means for such activites as investigating whether or not a certain identity exists in the system and which changes it underwent

17.2.3 Usage Components

- **Single Sign-on**
    - Single Sign-on enables an identity to perform its initial authentication and access numerous services and data without further re-authentication.
    - Initial authentication is typically performed by an associated Identity Provider, which issues a credential.
    - That Credential is then used to authenticate to other systems.
- **Personalization**
    - Personalization and preference management tools provide the identity an ability to set up individual settings for applications/services bound to that identity.
- **Access Management**
    - Similar to policy control
    - Identity can define policies as to which identity can access/modify which information.

18 WAM (WebComposotion Architecture Model)

WAM is a modeling language for designing distributed, organization-spanning web applications and (who would've thought) was developed by the lecturer/prof and is absolutely industry irrelevant for that matter. But hey that's just how things are at Chemnitz University. By the way are you looking for irrelevant + outdated material? You'd love to study some unstructured slides enhanced by Internet Explorer 6 screenshots? You hate effective use of slides and think that using results of pedagogical research to help students learn better is for nonbelievers? MAN I HAVE SOMETHING FOR YOU! Go to `https://www.tu-chemnitz.de` and get yourself the new and updated "Bullshit University Education"-StarterPack right now!