Advanced Internet Security

Lecture on

Advanced Internet Security

System Security, Process and Awareness

Prof. Walter Kriha

HdM Stuttgart

Lecture Overview

- Security Principles and Trends (today)
- The construction of a security solution (conceptual framework, architecture, infrastructure, APIs)
- Explain how security technology is embedded into the business and social context of a company
- Perform a security analysis
- Infrastructure security (protocols, services, Single-Sign-On, firewall architectures, object-based security)
- Software Security (attacks, buffer overflows, web-application security, frameworks)

Creating awareness for security problems in complex situations is our main goal!

Security Awareness: De-Mail

Heike Stach, project head portals at the secretary of internal affairs: "De-Mail is subject to legal regulations covering electronic communication. This means that tapping of contents is only possible after authorization through a judge, just like regular mail. Otherwise the complete communication and storage will be encrypted by the provider. (Detlef Borchers) / (vbr/c't) 2009

How do digital signatures, encryption and tapping go together in this case? Can you separate the technical protection of privacy from the legal protection? (take a look at the job-card discussion as well)

Strategic Goals

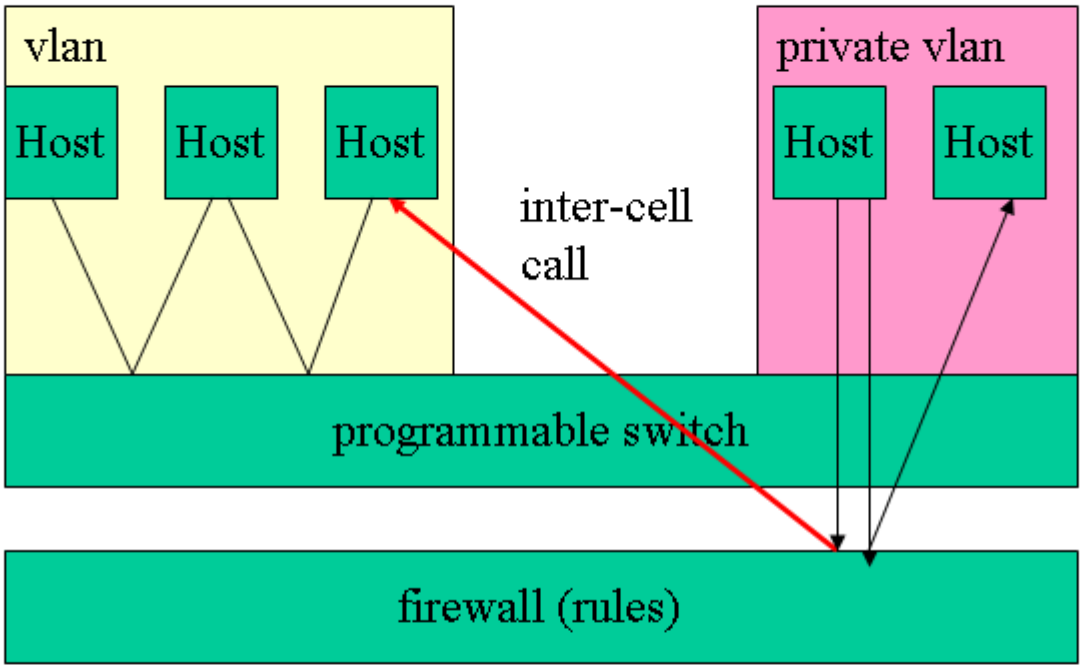
- Show technical and social reasons why security is such a problem for companies
- Show which security components exist and how they form a system
- Develop a conceptual security framework (policies, processes etc.)
- Explain software development within a security framework
- Create an awareness for security and privacy issues in systems, software and real live.
- Put the Principle-Of-Least-Authority (POLA) and authority reducution at the core of security architectures

Tactical Goals

- Experience the difference between channel based and object based security
- Understand mechanisms and consequences of Single-Sign-On
- Learn mechanisms and problems of authentication and authorization
- Learn some core APIs e.g. to authenticate users in Java
- Understand the problems of an end-to-end security approach within intranets
- Learn to identify security problems and to chose the proper mix of security technology to fight them.

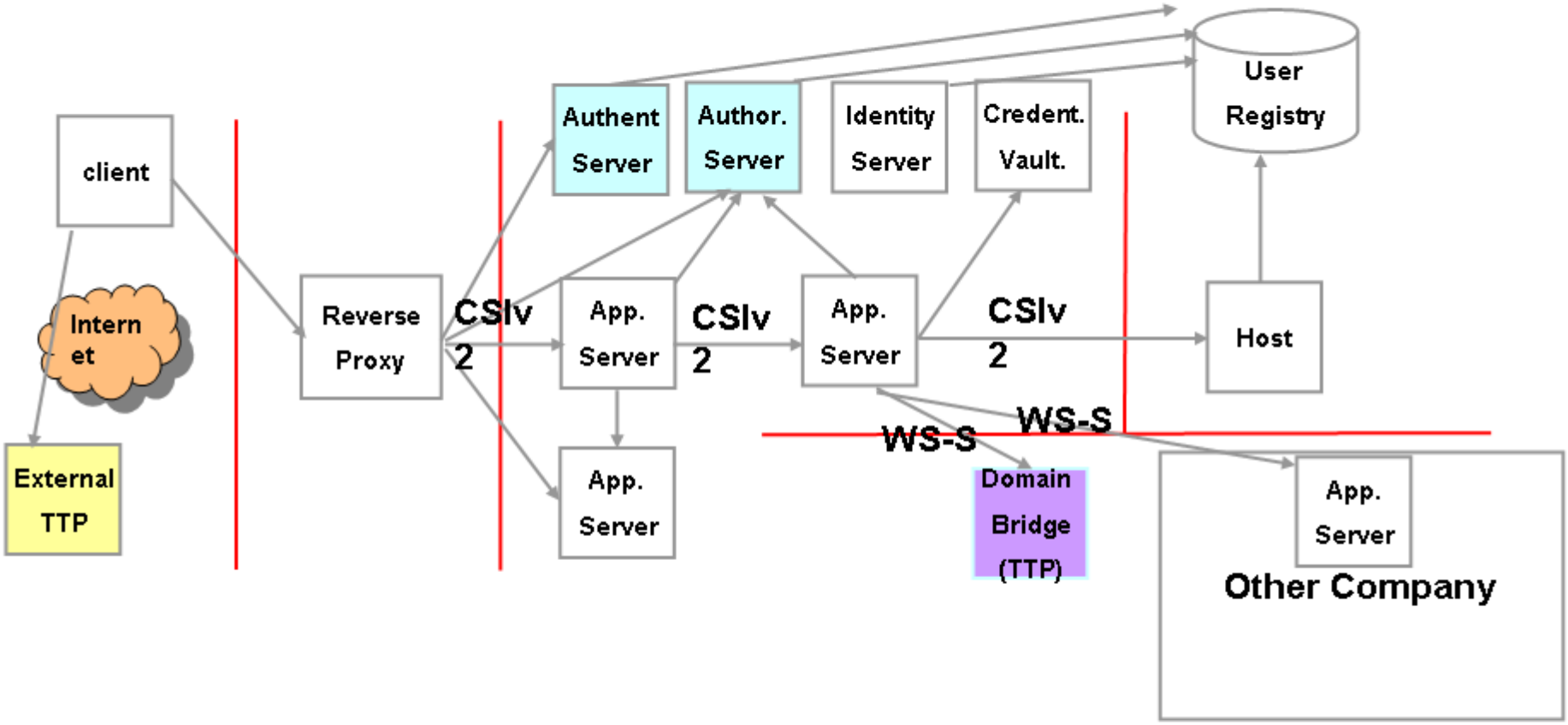
The approach used here is clearly based on a domain concept of security. Specific technologies are introduced when needed and explained in the context of a larger security problem.

Showcase: large scale firewall design



Based on private virtual lans, intelligent switches and network security cells a large scale firewall is designed to fit an international company. The vlan becomes „private“ by routing all requests through the firewall – cell internal ones as well as cross cell requests. Several firewalls have been collapsed into one to ensure rule consistency.

Showcase: End-to-End Security



Trusted Third Parties generate signed statements (tokens, certificates) which allow things, proof things etc. TTPs are useful to create federated domains as well. Theoretically the only place where client would produce her login credentials would be the first external TTP.

Security as a System

„If you think technology can solve your security problems, then you don‘t understand the problems and you don‘t understand the technology“

„To my initial surprise I found that the weak points had nothing to do with the mathematics. They were in the hardware, the software, the networks, and the people“

„.. in order to understand the security of a system, you need to look at the entire system and not at any particular technologies“

Bruce Schneier, Secrets and Lies, pg. xii ff. See also www.counterpane.com

Application Security as a System

Protocol/ Infrastruture	Software/ Implementation	Procedure	Context
Wrong authentication protocol for admins	Weak password handling	No security Sign-Off	Usability problems
Lack of access control	No filters	No security standards and procedures	Unexpected user/admin behavior
	No clean software architecture	No incident response in place	

Security is the result of a multi-dimensional effort. Some applications failed in all dimensions (see OBSOC at the Chaos Computer Club homepage).

System Threat Models

User Threat Model

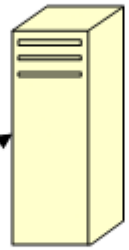
phising (credential attacks through social engineering)

certificate confusion

User Conceptual Model

Peer Threat Model:

sesson takeover, web trojans, XSS, SQL injection

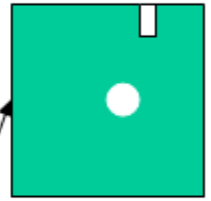


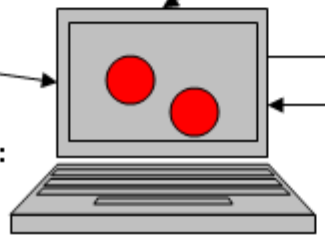
Web2.0 Threat Model:
Collaborating Users, malware, semantic Attacks

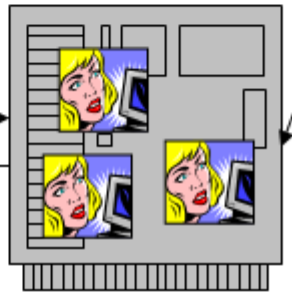
Developer Threat Model:

authorization errows

input /output validation errors







Server Threat Model:
SSL Cipher Specs

buffer overruns

authentication problems

maintenance problems (e-mail to customers etc.)

Plattform Threats

Plattform Threat Model:

Browser bugs

credential attacks (cookie/sessionID stealiing)

virus/trojans

Ambient authority

Trusted path

Situational Threat Model:

- home, kiosk, Internet Cafe

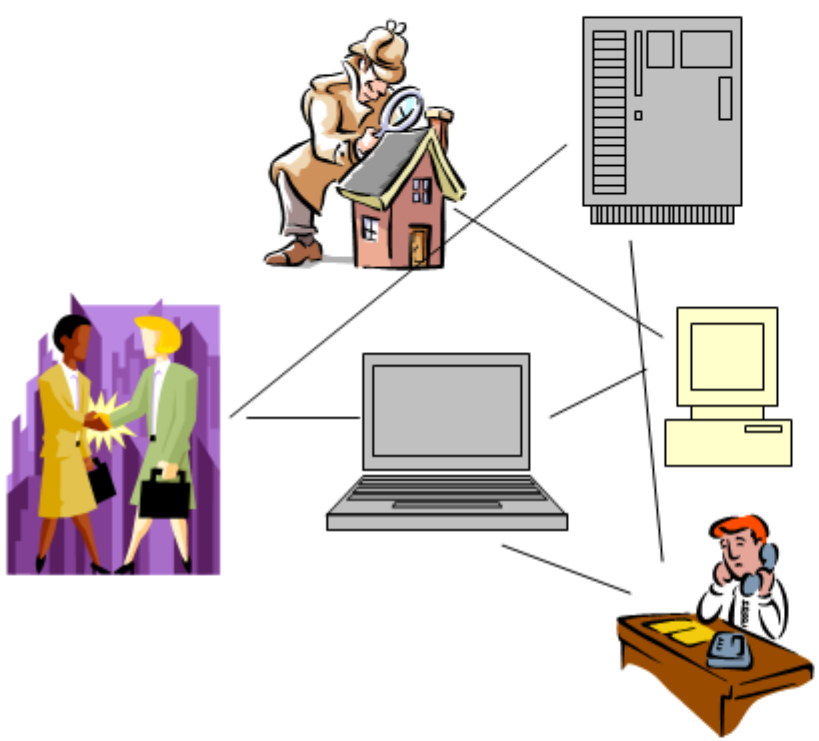
Internet Threat Model:

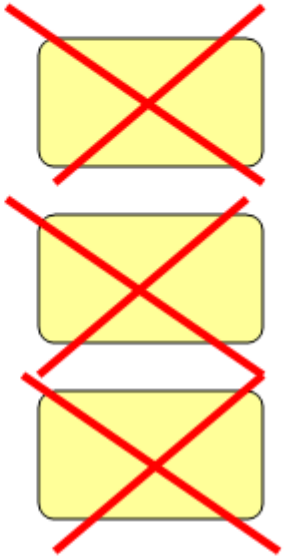
Integrity, confidentiality, partner ident.

Intranet Threat Model:
RBAC, SSO

A good threat model is the basis for security related design patterns which can be pre-implemented in the architecture of web-development frameworks. A threat model requires understanding of the technologies used: http, html, SSL, SQL etc. and of the partners involved and their conceptual models. Don‘t forget the developers and admins.

Security as an „ility“





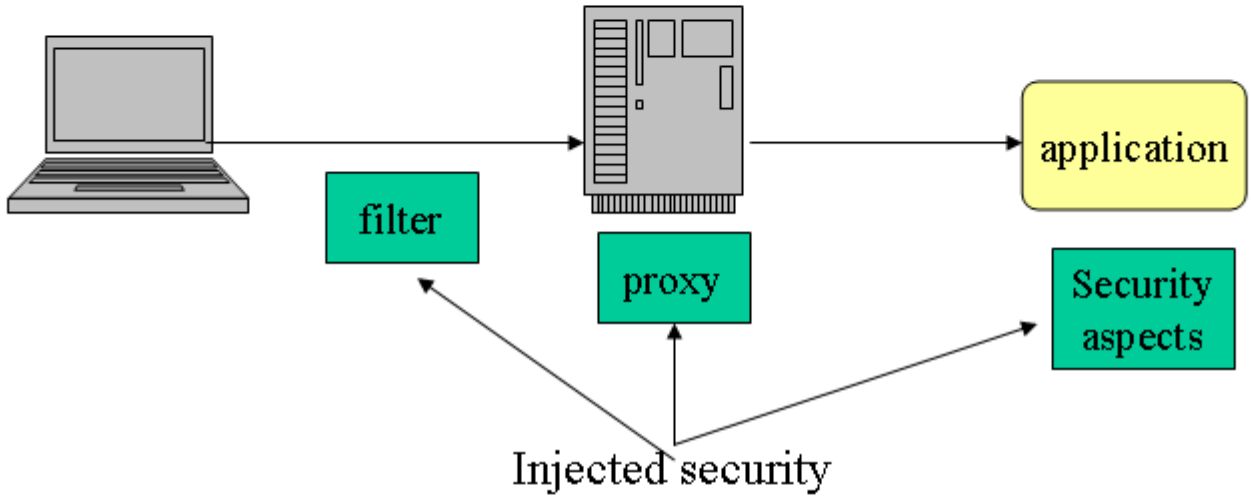
Security „Module“

Quality „Module“

Reliability „Module“

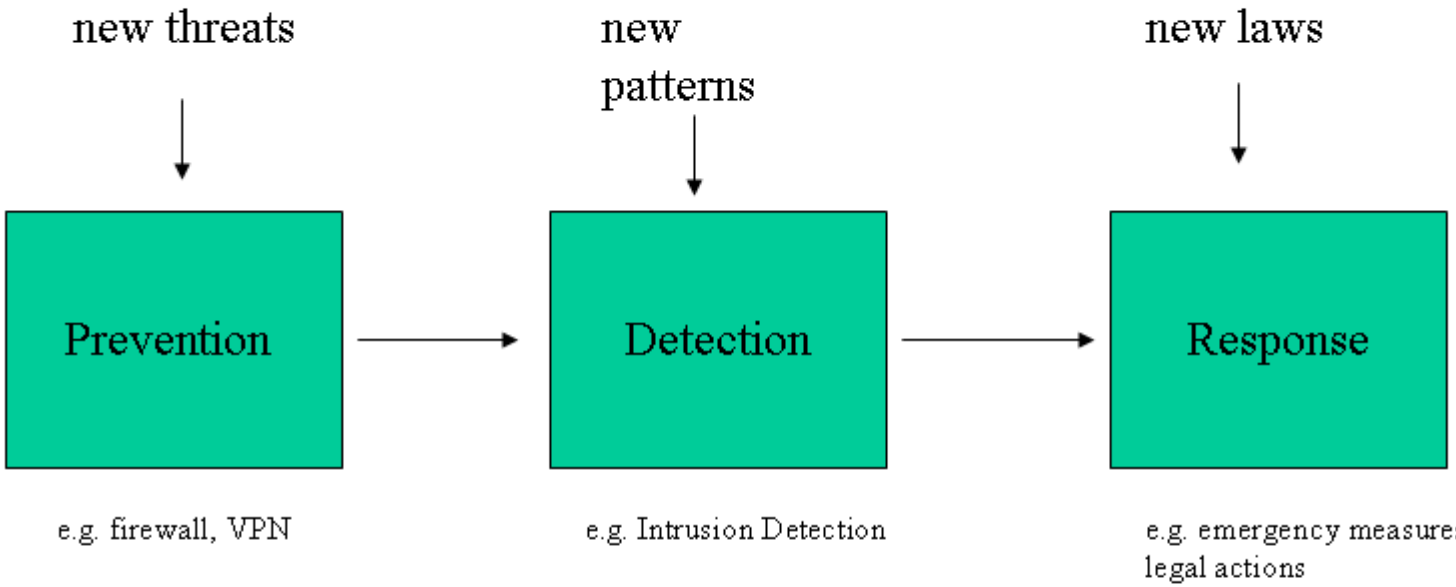
Many system features are so called „ilities“. The have in common that they do not contribute to functionality directly (they are „orthogonal“ to it) and that they cannot be located in one or at least a few spots. Instead, they are distributed all over the whole system. Therefore they are hard do concentrate in modules or components. Security is one of them: any spot in source code, network design, host setup or user behavior can wreck it.

Security as an „aspect“



Software security nowadays relies a lot on external components (infrastructure). But the architecture of software itself has certain security qualities (or is lacking there). Permission checks can be addes through external configuration – authority as a fundamental ability to cause causal effects is an architectural quality. Be wary of AOP with respect to security. Security rooted in infrastructure is both a benefit and a danger.

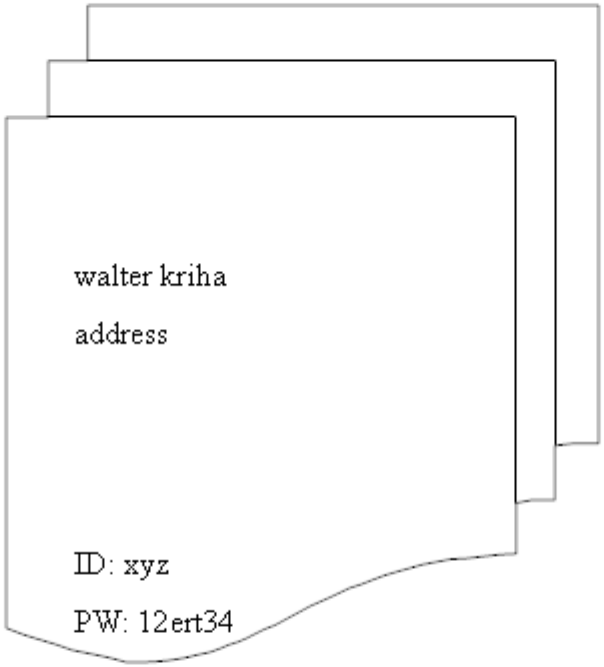
Security as a process



The definition and installation of a security infrastructure is only the beginning (this includes cost as well). If the infrastructure is not monitored, maintained and improved permanently it will be outdated very soon. Also, if nobody checks the audit logs, attacks go unnoticed and may finally succeed.

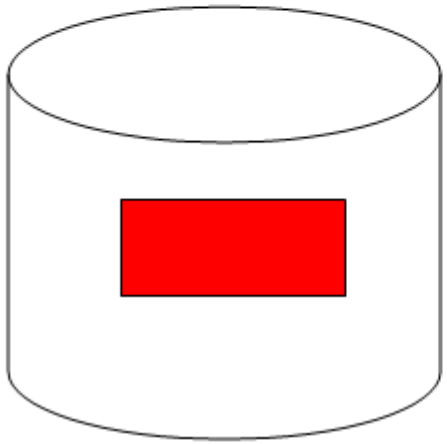
Make a test: assume a Cross-Site-Scripting attack was found on one of your pages in a web application. Do you have a strategy to deal with it? How long will it take to implement it? If you can‘t react within minutes you have a problem.

Security as an awareness problem (1)



Example: AOL mass mailing offers free internet hours. It combines personal address and login information. An interceptor can use the information to register a user and abuse the account. The users reputation and finances can be damaged. Of course, for AOL it is easier to combine personal data with login data right away so they know immediately who registers.

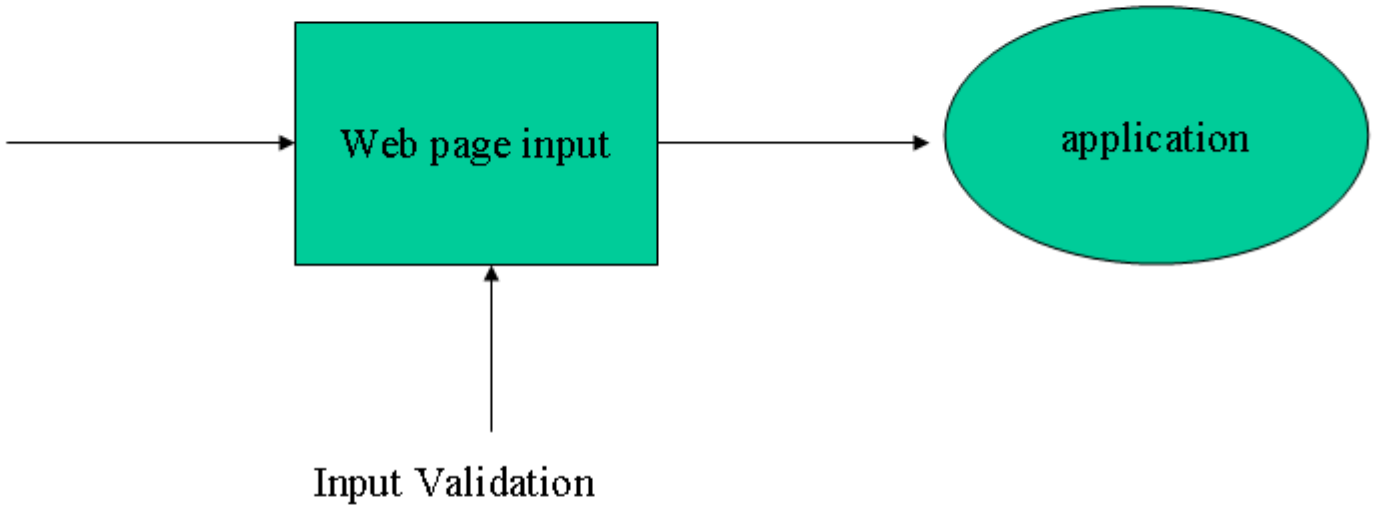
Security as an awareness problem (2)



A couple of lines in a software package may allow a fallback to weaker security modes (e.g. SSL)

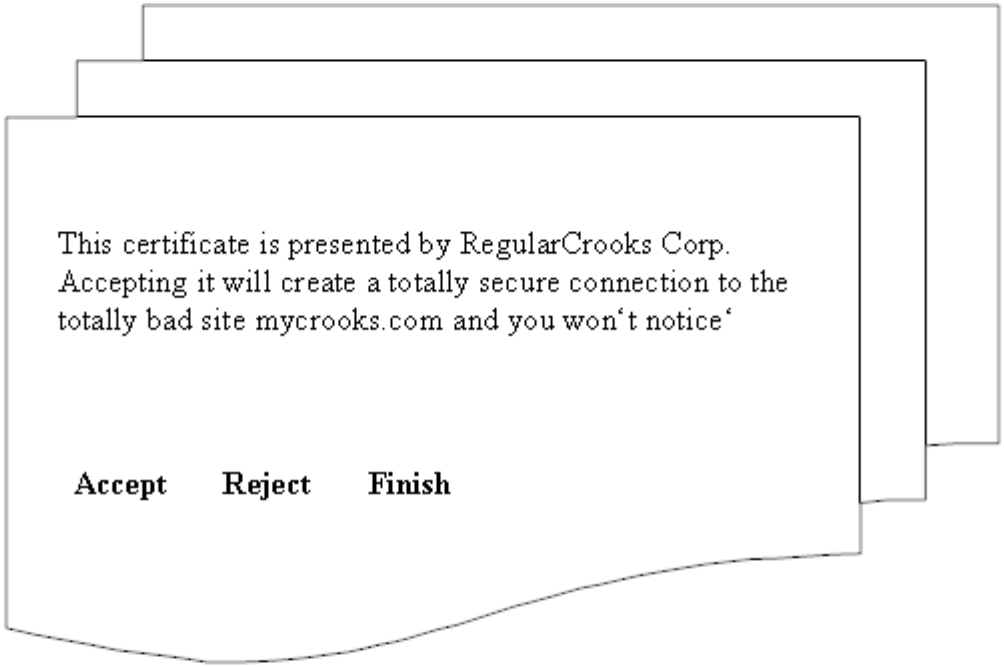
Even security „specialists“ create software that contains extremely dangerous features. Force vendors to explain their implementations – see security policy and guidelines. Force vendors to use secure defaults.

Security as an awareness problem (3)



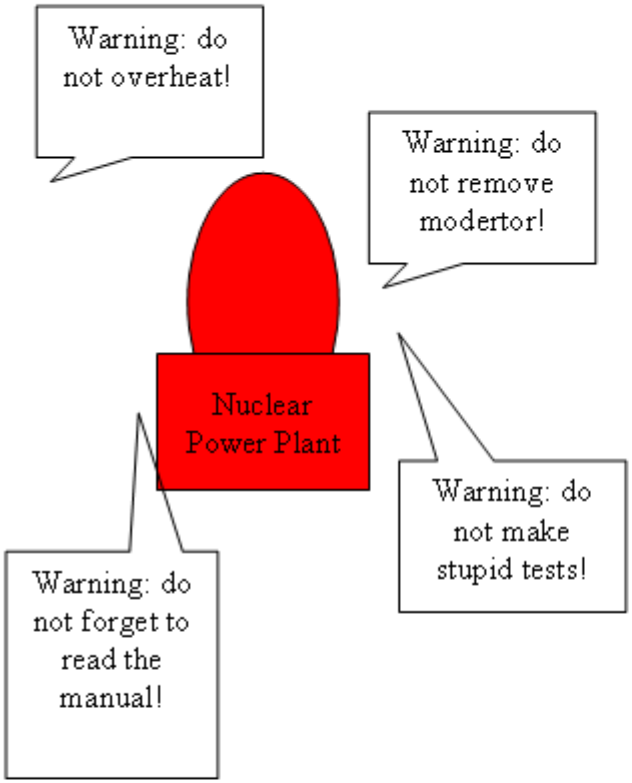
Can you tell for your application what kind of input is expected? Do you have a description of the grammer of your input language? And finally: Are you REALLY doing input validation or did you only go to a conference on the topic? Remember what Erich Kästner said: Es gibt nichts Gutes – ausser man tut es…

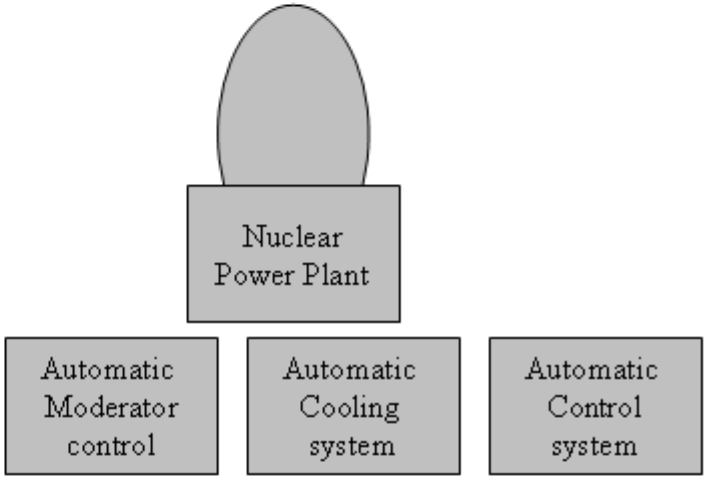
Security as an usability problem (1)



Do you really look at the many dialogs presented during an SSL session setup? Do you really know the server you are connecting? Many companies bounce you to some (to you unknown) download server for software upgrades. Or could it happen that you just hit the return button a couple of times to be done with it? Take a look at how Firefox 3 deals with self-signed certificates.

Security as an usability problem (2)





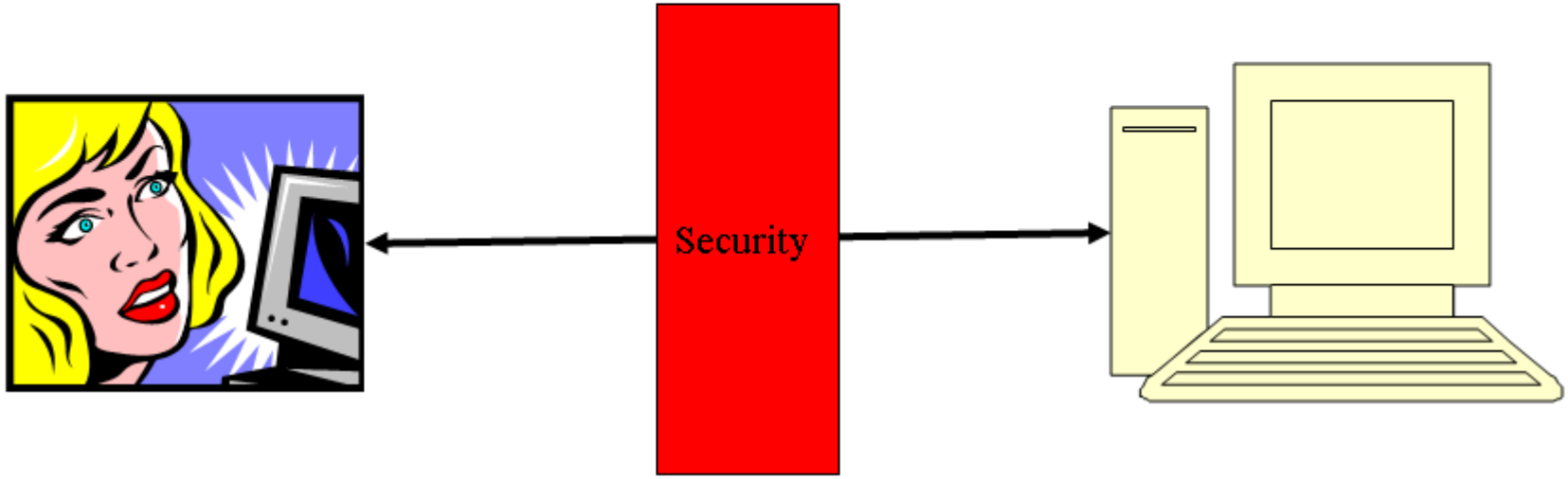
Do not overestimate usability! A fundamentally insecure system will not become safe through lots of warning dialogs. Only authority reduction will help. The well known Windows warning dialogs „do you really want to open….“ are an example of security by admonition (Ka Ping Yee)

Security as an usability problem (3)

Developers are human beings and have a right to usable security mechanisms. If developers have to use badly designed security mechanisms in software of infrastructure it will lead to insecure systems.

Make a case for usable security both for end-users and developers/administrators.

Security as a business problem (1)



Users want to get their job done. In many cases security is perceived as an obstacle for the user. Security mechanisms need to balance security requirements with usability and acceptance. Otherwise users will find ways to work around security. This is true for business users as well as software developers. This has consequences for security policies and processes. If your firewall team always says NO to requests they should not be surprised if firewall piercing happens frequently.

Security as a business problem (2)

In February 2002 Bill Gates announced that Microsoft would now put the focus on security in their products. The .Net server project declared a half year delay after that. February 2005 Gates introduces the distributed model for security and February 2006 for the first time Gates talked about authority reduction for Internet Explorer.

There is a clear dependency between business goals and product security. As Bruce Schneider says, there is little incentive for companies to make their products secure. Time to market or ease of use are valued higher than security issues. (see resources for wired article on vendors selling broken systems and the Gates statement)

Security as a legal problem

End User License Agreement and Guarantees:

This car is sold without any guarantee of fitness for any kind of purpose. It requires an ideal environment without rain, hail, etc. Do not use it for critical activities like shopping, vacations etc.

At any time this vehicle may lose parts, tires or general functionality. If trying to restart it does not help you may inquire about our repair rates.

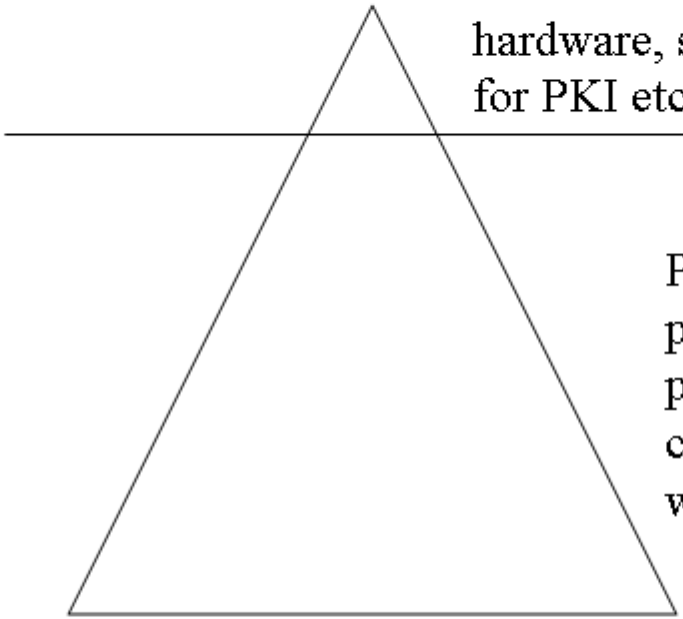
If we find a problem with the vehicles technology it is your responsibility to learn about this and have it repaired at your costs.

If the vehicle causes bodily or financial problems to you or anybody else due to construction or other failures in the manufacturing process: your problem

To make it short: we guarantee for nothing and you are carrying the risk and the costs associated with use.

It looks like computer products, especially software have been operating outside the law for many years now. This time is coming to an end (see the Gates statement). Expect a major impact on the software production process due to increased security requirements. The other big problem: A better customer protection will NOT solve the system problems behind company security!

Security as a cost problem

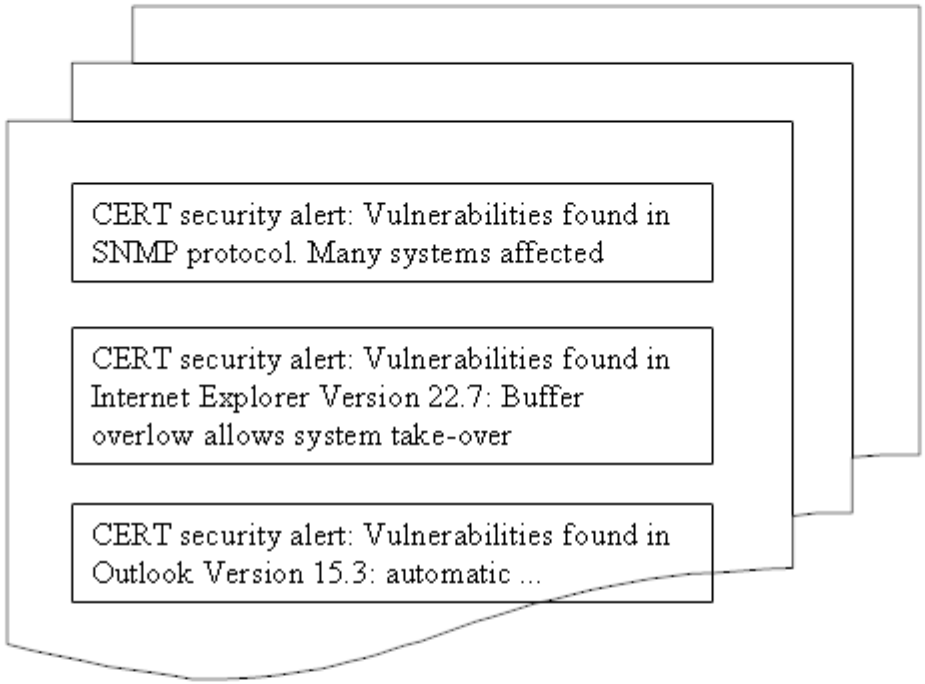


firewall hardware, VPN hardware, software packages for PKI etc.

Planning, eductation, help desk, processes etc. If you have a strict password policy with frequent changes and quality checks you will need a good help desk ...

Hardware and other infrastructure costs are impressive (e.g. 125K for an intelligent switch) but they are only the tip of the iceberg. Designing a security policy for a bank requires much more: education, process definitions, sign-off processes, endless meetings, software architecture definitions, help desk, legal work etc.

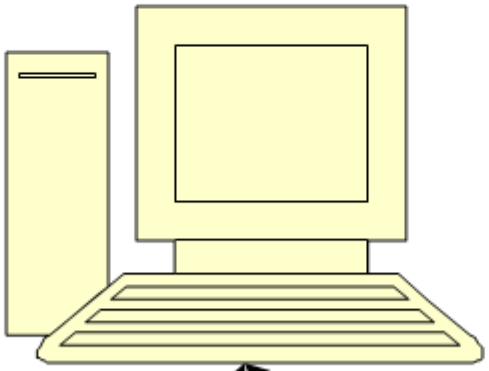
Security as a maintenance problem



Just keeping up-to-date on vulnerabilities and updates is almost a full time job. Do you know where to find this information for the products you use? Where to register for notifications? If you are a hacker, the best and safest way to find an exploit is to try well-known ones: there are countless systems out there running old versions of software. See www.cert.org , www.redhat.com and http://www.secadministrator.com (windows systems) to register. What does this mean for small and medium sized businesses? Is there a business opportunity behind it?

Security as a complexity problem

security technology is cutting across all domains and components (orthogonally to function and performance)



Users focus on their job and want to get it done. They are trusting and non-technical. Under pressure they will forget all security rules

Specialists are under business pressure to cut costs. They a weak in human factors too.

There is plenty of potential for misunderstandings in this complex relationship.

Security as a privacy problem

The USA Patriot Act changes some of Americans' fundamental legal rights in the name of the war on terror, including:

- Freedom of association: The government may monitor religious and political groups without evidence of criminal activity.
- Right to liberty: Americans may be jailed without being charged or being able to confront witnesses against them.
- Freedom from unreasonable searches: The government may search and seize Americans' papers and effects without probable cause to aid terrorism investigation.
- Freedom of speech: The government may prosecute librarians, telecommunication company officials and anyone else who reveals they have received a subpoena for records related to the terrorism investigation.
- Right to legal representation: The government may monitor penal communications between attorneys and clients, and deny lawyers to Americans accused of crimes.
- Right to a speedy and public trial: The government may jail Americans indefinitely without a trial.
- Freedom of information: The government has closed once-public immigration hearings, secretly detained hundreds of people without charges, and has encouraged bureaucrats to resist requests for public records under the Freedom of Information Act.

source: www.wired.com. How anonymous are users of the web?

Security Theater

Ask every proposed security measure whether it:
- Really solves the problem it is intended to solve
- does not create undue costs way beyond the risks covered (mass investigation, mass data collection)
- does damage security instead of increase it (e.g. looking for a rare event in huge amounts of data creates many false positives)
- is a reasonable trade-off between risk and danger (remember: security is a trade-off)
- might not increase real security but might align risk perception and real risk better

Remember that the monkey in us

-Will overestimate risks which are especially gruesome but not very likely
-Will underestimate risks which have some positive side-effect (smoking)
-Will judge risks according to availability
- Will generally be unable to do a reasonable risk judgement based on statistics
- Will judge risks emotionally

source: www.schneier.com.

Does this mean technology is unimportant?

- Threats (Network sniffing, attacks, trojan horses, viruses, worms, IP spoofing)
- Virtual Private Network: FreeSwan - IPSEC with Linux
- Secure e-mail: Pretty Good Privacy and GNU Privacy Guard
- Firewalls: Packetfilter, Stateful Filtering, Stateful Inspection,
- Circuit Level and Application Level Firewalls
- Webserver with SSL Support (Linux/Apache)
- Virus protection: Antivirus MailExchanger
- Software Bugs: Buffer Overrun Bugs
- Network Intrusion Detection (Snort under Linux)

These topics will be covered in the exercises. In the lecture we will concentrate on what else is necessary to make a system „risk manageable“. A special focus will be on software design and processes as well as frameworks (Security Frameworks (JAAS, J2EE, Sandboxing, SE-Linux, RBAC). Some of the above will also be introduced in the lecture (needs to be synchronized with the exercises).

General Security Principles

- **Least Privilege (Need-to-know, need-to-do). Do not grant more rights than needed to fulfill a certain task. (Unix root/windows admin violates this principle)**
- **Avoid ambient authority: do not leave authority (the ability to cause effects) lying around.**
- **Default is „deny“: Never allow everything and then start taking rights away. Do it the other way round.**
- **Defense in depth: Do not rely on one line of defense only**
- **Concentrate defensive measures: Do not distribute defensive measures too far, you will only get synchronization problems. This rule contradicts the „defense in depth“ principle.**
- **Protection, detection and response: Do not just try to prevent security incidents: Go and expect them, track them and be prepared for emergency measures.**
- **Permanent vigiliance: The true costs of this principle are staggering according to Gartner Group. Not least because broken systems are sold**
- **Fail save stance: An error leaves the system in a state where no access is possible – not even legal access.**
- **No security by obscurity: But don‘t tell about infrastructure**
- **Simplicity is so important (example: step-up authentication)**

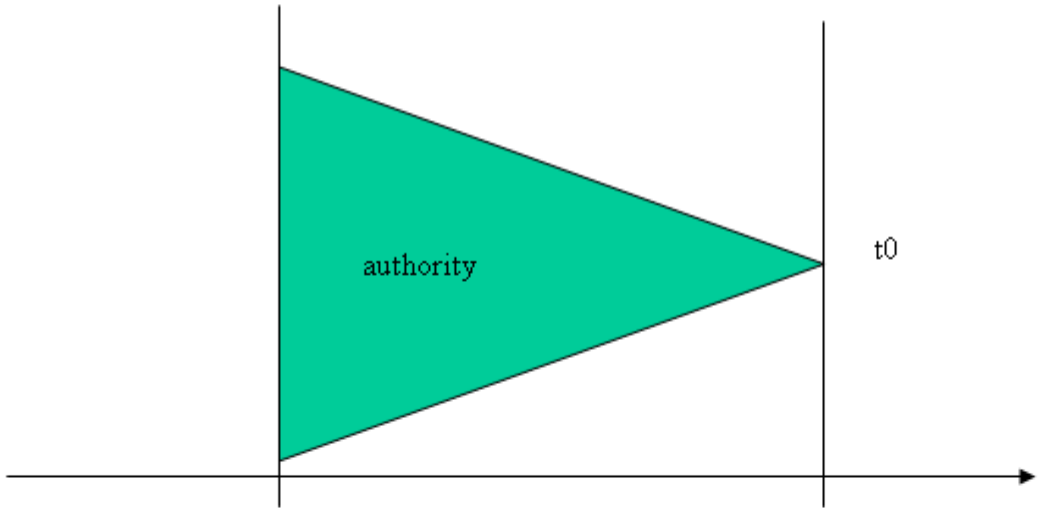
These principles make more and more sense over time and serve - like the names in design patterns - as stand-ins for complex problems.

Permission, Authority, Causality

Permission: I can potentially do something – but am I allowed? Software Architecture vs. Protection System

Causality: Influence and propagated authority that finally led to an effect

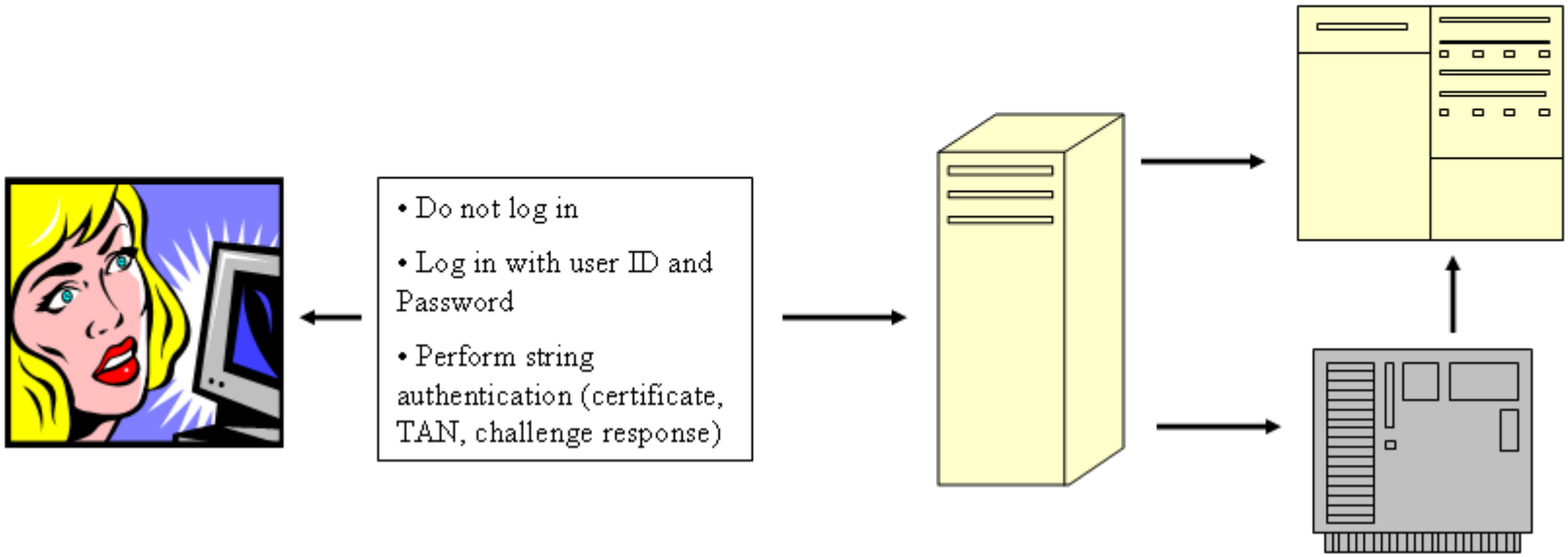
Authority: Ability to cause an effect



See John Sowa, Process, Time and Causality. This lecture has its focus on the permission aspect (security as correctness of a solution with respect to its business goals). The master lecture deals with the authority aspect in a much wider sense: security as a sub-aspect of general system safety.

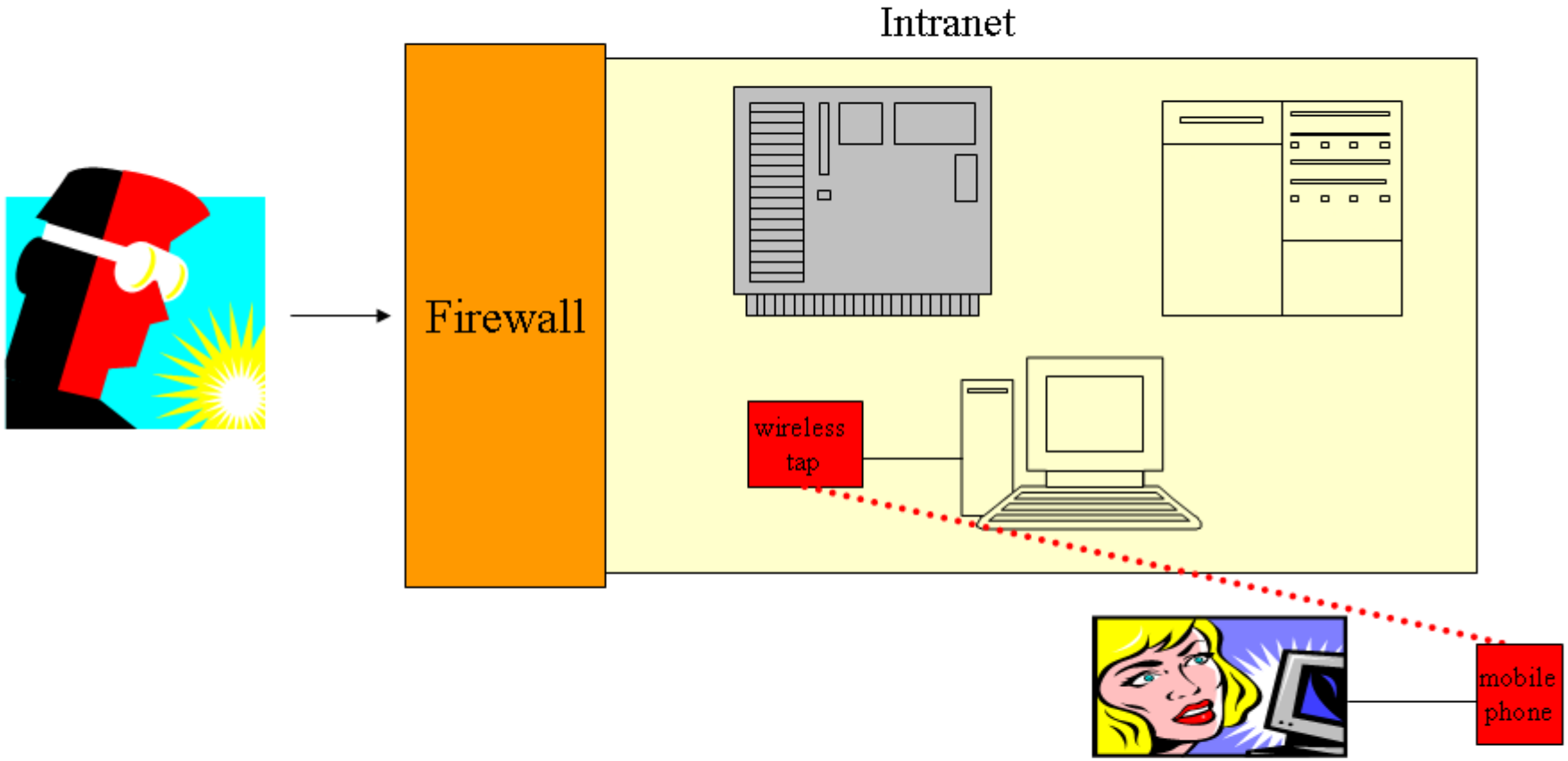
Security Trends and Analysis

Example: Step-up authentication: a good idea?



Providing different authentication level looks convenient but requires that ALL processing steps involved with user requests check the current level and possibly deny access. The customer facing modules catch the error and create a new login dialog. If only one backend system or component does not check you have created a big security hole. There is also a usability problem behind: it is quite confusing for the user to be suddenly prompted for additional credentials.

Example: Wireless through the wall



You‘ve just finished a multi-million dollar demilitarized zone with the latest in firewall technology just to find out that some users connected a wireless tap to their desktop machines so they could access their machines (and the whole intranet) via their mobile phones attached to laptops etc. Now ain‘t this convenient? How do you prevent and detect such things? What does this mean for the future of firewalls and intranet security? Is there still an intranet?

Security Analysis

We will use showcases like the previous one about wireless taps to demonstrate various use-cases for security analysis and how it could be performed. We will look especially at three cases:

1. Preparation for new technology offering new threats but also new possibilities to improve services for the business users (the wireless tap example in detail)
2. How to react if a new vulnerability is detected (showcase: recent SNMP warning by CERT)
3. Emergency response for incidents. How to react on security incidents: Being under attack
4. How to analyse standard software for use within the company: Questions to ask about encryption, user handling, protocols and interfaces, legal stuff.
5. How to analyse internal software: Specifications, risk-analysis and Sign-off process

Please see Jürgen Butz, Mobile Security (resources) for a complete analysis and mitigating measures.

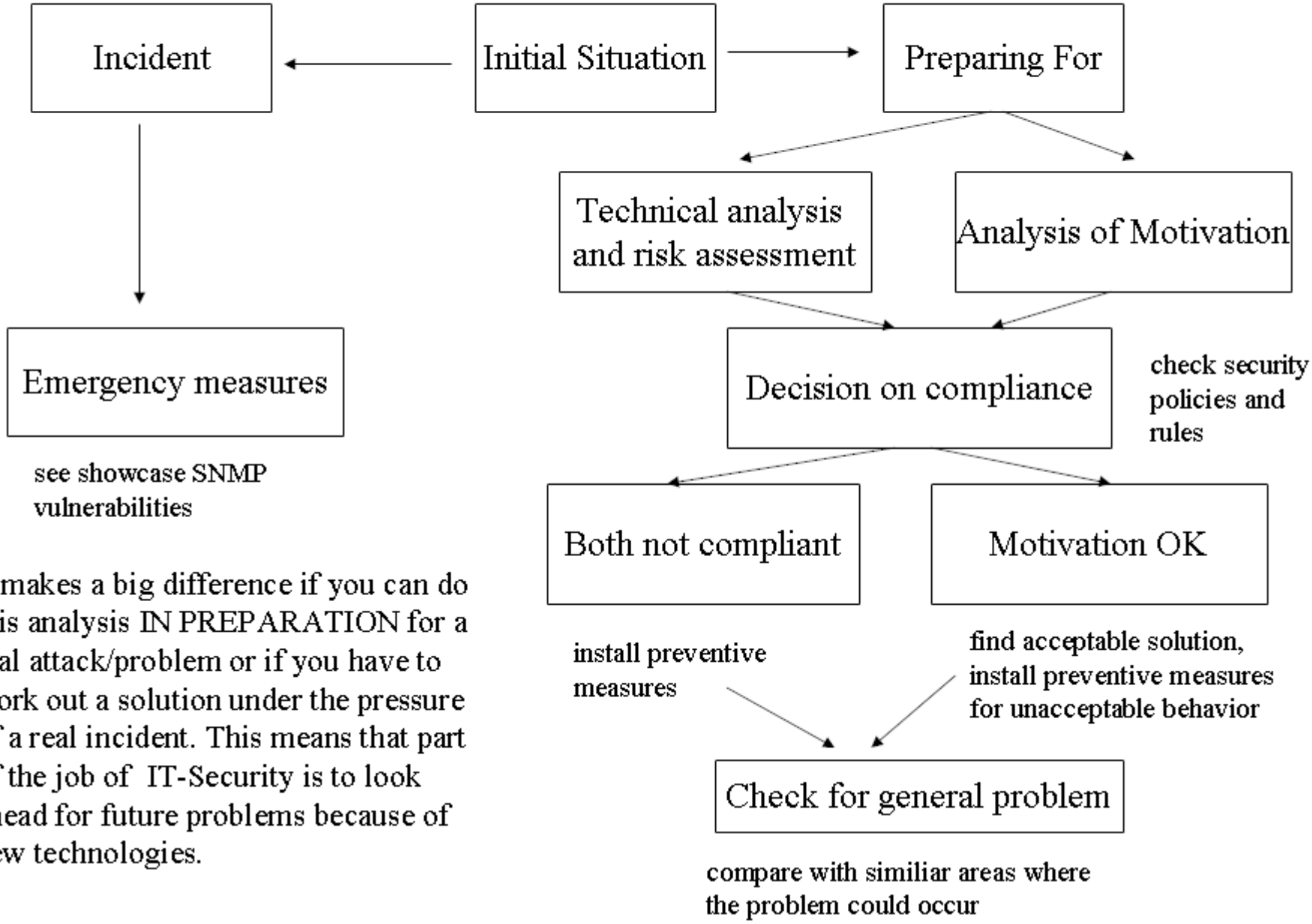
Steps of a Security Analysis (1)

- The following will assume that you have READ about the problems with wireless taps but that this is NOT a real INCIDENT yet. You want to be prepared!
- How critical is the situation with the wireless taps? Clearly work out the technology, possible threats and consequences for the whole company (business, reputation, processes etc.)
- Work out the MOTIVATION behind the behavior (adding wireless taps). Is it criminal or more a question of convenience or the will to do a good job?
- Come to a decision if the Motivation is in compliance with the security policies. Does the company acknowledge that there is a real need for the behavior?
- Come to a decision if the behavior/technology used is in compliance with security policies.

Steps of a Security Analysis (2)

- Define your response depending on compliance of motivation and behavior:
- If neither motivation nor behavior are in compliance with security policies: Take measures to prohibit/avoid the problematic behavior (legal, technical)
- If behavior is problematic but the motivation is justified and in compliance: Take measures to transform the problematic behavior into an acceptable one.
- Do a further analysis of the situation: Do you see signs of new technology generating problems? Do you see other areas that might show the same vulnerabilities (private modems, wireless devices like keyboards etc.)
- You might have discovered a general problem which requires further analysis and possible changes to your security infrastructures and policies. In this case: do you need more internal encryption on the intranet?

Steps of a Security Analysis (3)



It makes a big difference if you can do this analysis IN PREPARATION for a real attack/problem or if you have to work out a solution under the pressure of a real incident. This means that part of the job of IT-Security is to look ahead for future problems because of new technologies.

Results of a Security Analysis (1)

The need for wireless mobile connections to the intranet for business users was DENIED. IT-Security takes measures to prevent the use of wireless taps:

1. Prevent installation of non-standard software on workstations and PCs. This could be a major system engineering effort. One solution is to scan all stations every night and delete all software not registered with system management. Access to drives and partitions can be denied as well (user rights).
2. Communicate decision to ALL employees via intranet and direct mail. Possibly have everybody sign a declaration of compliance. Update your policies and rules if necessary. This step protects you from legal problems and sends a clear message to everybody.
3. Have corporate and IT-Security check regularly for wireless taps.
4. Increase efforts to further secure the intranet via strong authentication and encryption.

Results of a Security Analysis (2)

The need for wireless mobile connections to the intranet for business users was ACCEPTED. The technlogy itself (wireless taps) was declared illegal. IT-Security takes measures to prevent the use of wireless taps (see previous page)

1. In addition to preventive measures and detection of violations, a process is started to provide a pool of wireless modems, protected by the central firewall and company standard encryption. The results of this process (software, systems, rules etc.) WILL GO THROUGH A SEPARATE SECURITY ANALYSIS.
2. Even if the motivation is OK, the result of the analysis could be that current encryption on wireless communication devices is not good enough to implement such a wireless pool.

The lessons to be learned here are that just saying NO is not a good strategy in many cases. (Actually, this COULD be a case for a clear no).

And that technological change will permanently be a threat to your infrastructure. The next big thing besides wireless communication could be the large scale use of SOAP based WebServices which pass firewalls easily because the use port 80 (http). In effect creating a remote procedure call „hole“ into the company. Vigiliant security people are already looking at filter/gateway technologies to deal with the situation.

Security trends of the next decade

- Further De-Perimeterization (no more borders)
- Consumerization (office computers replaced by cheap, consumer level goods (xxxpads, xxxphones, etc.) and still used within companies
- Decentralization (offloading of data to clouds. Multi-location problem)
- Deconcentration (embedded special purpose computers and sensors, from mainframe to PCs to special hardware)
- Decustomerization (Hardware vendors, social networks, service companies. Who‘s customer are you really? Google users are googles product!)
- De-Personization (agents acting for us with other agents and machines. Machines as social gatekeepers)

From: Bruce Schneiers forward to Security 2020

The enemy is you!

There's really no such thing as security in the abstract. Security can only be defined in relation to something else. You're secure from something or against something. In the next 10 years, the traditional definition of IT security—that it protects you from hackers, criminals, and other bad guys—will undergo a radical shift. Instead of protecting you from the bad guys, it will increasingly protect businesses and their business models from you.

Welcome to the future. Companies will use technical security measures, backed up by legal security measures, to protect their business models. And unless you're a model user, the parasite will be you.

— Bruce Schneier

Dispossession and Loss of Control

- Who owns your game console?
- How many miles will your car be allowed to run? How many pages is your printer allowed to print?
- What will your smart meter tell the energy providers? Military and business wisdom says you should not tell enemies about your needs and regular behaviors.
- Will software „age“ your devices and consumer goods?
- How many times will you be allowed to read your e-books? Watch your movies?
- Malware will be remotely deleted – what else will disappear?

Software will allow fine grained usage control of your applications and devices, based on the data the same applications and devices have already collected and sent to your providers. Energy providers create databases with „fingerprints“ of your electronic devices at home, collected by smart-meters.

Resources (1)

- Doug Howard, Kevin Prince, Security 2020: Reduce Security Risks This Decade
- Bruce Schneier, Secrets and Lies, Digital Security in a Networked World. In this book Schneier turns from his previous believe in cryptography to a system-oriented approach. Shows how the best cryptography can be made useless easily.
- www.counterpane.com, Schneiers homepage with articles on all security aspects e.g. the effect of WebServices on firewalls etc.
- www.cert.org/encyc_article/tocencyc.html Security of the Internet. A short primer, good to read.
- www.cert.org/tech_tips/home_networks.html A good introduction to securing your home systems
- Diffie/Landau, Privacy on the line. How the right to privacy is threatened by governments.
- Juergen Butz, Mobile Security, **http://www.linecity.de/ A complete analysis and coverage of mobile security issues.**

Resources (2)

- Dan Geer, Risk Management is Where the Money is. Looks at how insurances and banks handle risk by quantifying it and then turning it into a business.
- The Strange Tale of the Denial Of Service Attacks against grc.com, by Steve Gibson. http://grc.com/dos/grcdos.htm . Shows how script kiddies can shut down internet sites by using tweaked IRC clients and remote control agents. Quite interesting.
- Deanonymizing users of the SafeWeb Anonymizing service. Explains why the SafeWeb service does not work because it still allows (requires) script code. What else do you expect from a company where the CIA is a founding member?

Resources (3)

- Linux Sicherheit, Tobias Klein. Shows how to work with open source software on Linux. Tackles almost all operational technologies.
- Building Internet Firewalls, Zwicky & Cooper. Covers firewalls in depth. First part explains DMZ architectures. Second part goes into protocol details and can be used like a dictionary.
- „Gates finally discovers security“, wired magazine, http://www.wired.com/news/infostructure/0,1377,49823,00.html. What could you do to make a decade old patchwork of software secure – and still keep a customer base that is used to do things quickly and easily (e.g. outlooks content handling)?
- „Do OS vendors sell lemons?“, wired magazine, http://www.wired.com/news/politics/0,1283,50931,00.html?tw=ascii. Has numbers on broken government systems and concludes that vendors ship their systems already broken.
- Walter Kriha, Security and Software-Quality, talk at BWCon/eXept. http://www.kriha.de/krihaorg/dload/security/quality/securityandquality.ppt