Assessing and Mitigating Risks in Computer Systems

Lars-Helge Netland

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen, Norway

June 2008

ISBN 978-82-308-0631-9
Bergen, Norway 2008

Printed by Allkopi Tel: +47 55 54 49 40

Table of Contents

Executive Summary

When it comes to non-trivial networked computer systems, bulletproof security is very hard to achieve. Over a system's lifetime new security risks are likely to emerge from e.g. newly discovered classes of vulnerabilities or the arrival of new threat agents. Given the dynamic environment in which computer systems are deployed, continuous evaluations and adjustments are wiser than one-shot efforts for perfection. Security risk management focuses on assessing and treating security risks against computer systems. In this thesis, elements from risk management are applied to two real-world systems to identify, evaluate, and mitigate risks. One of the pinpointed weaknesses is studied in-depth to produce an exploit against the affected system. In addition, approaches to handle common software security problems are described.

Acknowledgements

I sincerely thank my supervisor, guru, and friend Professor Kjell Jørgen Hole for truly remarkable guidance over the last five years. It has been a privilege to work with you and your excellent team of researchers in applied computer security. I extend my deepest gratitude for the all the hard work you have done over these years, and wish you the best for the future. Many thanks to André N. Klingsheim. I have cherished our joint ventures, and look forward to turn our great teamwork into a profitable business. Great things happen when we put our minds together. A big thank you to Yngve Espelid for close collaboration over the last five years. It has been an outstanding pleasure to work with you. You have special talent for making and breaking software. You are also very good at chugging Foggy Bottom Ales. I also thank Khalid Azim Mughal for encouraging and helpful advice during my PhD studies. Thank you to former and current members of the NoWires Research Group. I will definitely miss our monthly gatherings. I would also like to thank co-authors Thomas Tjøstheim and Vebjørn Moen. In addition, I thank mom for weekly meals and support; dad for both stressing the importance of a solid education and sweeping the way for a member of the "curling generation"; my brother for all the unconditional help; and Mike for adopting and housing a total stranger back in 1997. In closing, I would like to thank my common-law spouse Kristine for her love, patience, empathy, and support. You mean the world to me.

Assessing and Mitigating Risks in Computer Systems

Introduction

Over the last years, a criminal revolution has taken place in cyberspace. Online criminals now employ basic economic concepts to develop their fraudulent businesses. Malicious hackers rely on techniques such as *specialization* of goods and services, examples include phishermen who create fake websites and botnet herders who manage large collections of compromised computers; *outsourcing of production*, exemplified by the growth of automated crimeware tools; *multivariate pricing*, best illustrated by credit card fraud schemes, where factors such as issuing bank, geographic location, and the credit card's rareness determine its value; and *bulk pricing*, demonstrated through offerings of large collections of e-mail addresses and credit card numbers to discount prices [1, 2].

Secure Computer Systems

The production of secure networked computer systems is a difficult undertaking. A striking difference from traditional engineering disciplines is the presence of skilled and creative attackers, who relentlessly try to break systems. Other factors that add to the challenge of creating secure computer systems include the trinity of trouble [3]:

- The growing *connectivity* of computers and software. The ongoing push to publish systems on the Internet results in increased risks, as adversaries find it easier to launch attacks;
- the increasing degree of program *extensibility* results in flexible software that can swiftly accommodate new business needs, but also open up for unwanted malicious extensions; and
- the rising *complexity* of computer systems. More lines of code translates to a higher probability of introducing vulnerabilities.

Systems fail for a number of reasons. In [4], Anderson argues that cryptosystems fail more often because of implementation faults and managerial issues than because of insufficient cryptographic primitives. In the case of software security problems, recent numbers show a 50/50 split between implementation *bugs* and design *flaws* [5]. A bug is an implementation-level programming mistake that most often can be easily fixed. Buffer overflows are bugs that can usually be removed by introducing proper array bounds checking. A flaw is a weakness introduced at the design-level that cannot normally be fixed through simple adjustments to the program code. A system redesign is often needed to remove flaws. Whereas many bugs can be identified by automated tools, finding flaws typically require manual inspection by trained professionals.

It is widely recognized that the cost of fixing defects increases with each new stage in the Software Development Life Cycle (SDLC). Studies show that a flaw left unfixed in the design phase can become more than ten times as expensive to remove during system maintenance [5]. This suggests that companies are wise to have a strong focus on detecting and resolving security problems as early as possible in the development process. As a consequence, identifying design flaws is more attractive from a cost-saving perspective than discovering implementation bugs.

Security Risk Management

Risk management of system architecture and design has shown great promise in finding and removing flaws. At Microsoft, this technique has been identified as a critical success factor for the company's strides in software security [6]. Risk management involves assessment and treatment. In the risk assessment phase, the risk management team creates an overview of the system, identifies system threats and vulnerabilities, and evaluates and ranks threat/vulnerability pairs. The risk treatment phase outlines steps to mitigate unacceptable risks.

Overview

This thesis consists of a topic introduction and seven papers. Two papers use elements from risk management of real-world systems to identify and handle system design flaws. Two additional papers explore a Man-in-the-Middle (MitM)[1] vulnerability identified in the risk analysis of the Norwegian banking industry's newest Internet banking system, called BankID. One paper presents a pattern for input validation, which enable newcomers to computer security to quickly understand the basics of this prevalent form of defect. The two last papers give accounts of the implementation of two security software prototypes.

The rest of the introduction is organized as follows: first, key concepts used throughout the dissertation are developed; then an overview of risk management in computer security is given; next is a discussion on trust and its relationship to risk; followed by summaries of the seven papers in the dissertation; then some brief comments on the societal impact of the work in this thesis are made; in closing, some remarks on possible future research are given.

Computer Security

A *computer system* is the collection of hardware, software, information, and people that an organization make use of to carry out computing tasks [7]. *Computer security* concerns the building of computer systems that continue to function under malice, error, or mischance [8]. Combined, these definitions hint at the large challenges involved in making secure systems. Firstly, the broad scope significantly adds to the problem of creating secure computer systems. Making secure software is a hard problem in itself, but does not solve the computer security problem alone. Phishing—a scam that tricks end users into revealing their secret credentials— is a form of attack that deceives the human component of a system, and cannot be dealt with exclusively in software. Secondly, a thorough treatment of computer security requires expertise from many disciplines. Examples include lawyers who can help to ensure that legal requirements are understood and met; economists who can aid in analyzing and shifting the financial incentives for potential attackers; and cryptographers who can assist in safe use of cryptographic primitives.

[1]'Middleperson' and 'interposition' are politically correct terms that refer to the same class of vulnerabilities. Published papers in this dissertation use the MitM phrase, and it will therefore be used throughout the text for consistency.

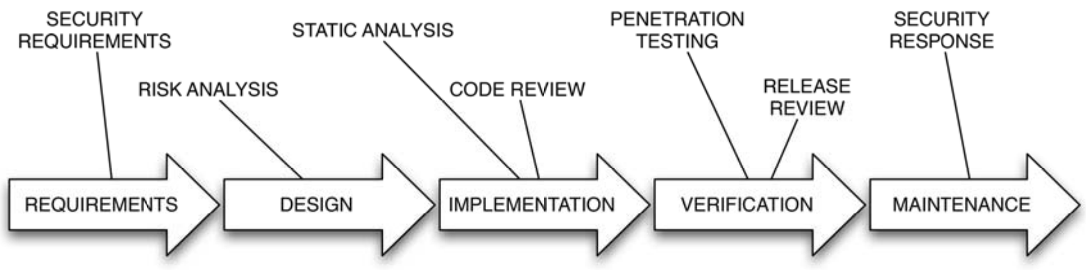


Figure 1: Security best practices in the SDLC

In “The Protection of Information in Computer Systems” [9], Salzer and Schroeder describe security as techniques that control who can use or modify information in a computer system. In particular, categories for security violations include unauthorized:

- information release,
- information modification, and
- denial of use.

The authors go on to argue that protecting a computer system against security violations is a difficult undertaking, and that no complete method exists that allows the construction of secure large general-purpose systems. Instead, they point to eight design principles to guide developers in creating more secure systems. Examples include keeping the design as simple as possible, a warning against believing that holding the design secret will make a system secure, and a recommendation in favor of intuitive and easy to use interfaces.

Today, over 30 years later, we still rely on best practices when developing all but the simplest systems. Introductory textbooks in computer security all describe the three cornerstones of security: *confidentiality*, *integrity*, and *availability*, that cover the same ground as the three previously described security violations. These are not independent concepts, as they sometimes exclude one another, and therefore must be carefully balanced. As an example, encrypted information will only be available to those who possess the secret(s) needed to decrypt that data. A number of security services have been developed to realize and extend the three fundamental goals of security. In particular, these concepts are central in this thesis: *Authentication* is the process of establishing an understood level of confidence in the truth of some claim; and *non-repudiation* provides protection against someone that falsely denies that a communication took place [10, 11].

Software Security

Software security is the idea of engineering software that remains dependable under malicious attacks [5]. Current methodologies for creating secure software include Microsoft’s Trustworthy Computing Security Development Lifecycle (SDL) [12], the Open Web Application Security Project’s (OWASP) Comprehensive, Lightweight Application Security Process (CLASP) [13], and McGraw’s software security touchpoints [5]. Fig. 1 summarizes common security best practices shared by the three approaches. The labeled arrows denote the typical stages of software development, while the best practices are described in text. Fig. 1 does not favor any particular development methodology, it simply captures a relationship, depicted with lines, between best practices and the stage of development where they typically occur.

The *security requirements* best practice from Fig. 1 involves identifying and planning security features demanded by the customer, as well as incorporating requirements mandated by industry standards and regulations. *Risk analysis* concerns pinpointing and assessing weaknesses that could potentially be exploited. The next section discusses this activity in detail. *Static analysis* involves using tools to locate software bugs. 'Findbugs' is one such tool developed to inspect Java source code [14]. *Code review* entails manual and dynamic code inspection. The first involves using programmers to review the code by hand, while the latter relies on automated tools to check the behavior of executing code. The runtime functioning of applications that use the HTTP(S) protocols can be examined in the 'Webscarab' framework [15]. *Penetration testing* seeks to break software by simulating attacks on the system. Different approaches exist, ranging from black box testing, in which testers are told to evaluate the system from an external point of view; to white box testing, in which the hired guns get detailed system information. The *release review* seeks to determine if the software is ready to be delivered to customers. *Security response* is about planning for failure by deciding how to gather and handle reported software weaknesses in the future. The constant evolution of threats and vulnerabilities will most likely require actions to improve the security of the system after release.

In applying the security best practices, *external review* can be a useful principle. This practice entails using people outside the development team to assess deliverables from the SDLC. Microsoft relies on a central security team that conducts release reviews and continuously improves the individual security best practices.

Security challenges

A report addressed to the US President [16] in 2005, pointed out a shortage of security researchers and practicians in the US, and called for urgent measures to remedy the situation. Elements in the call to arms were increased spending on cyber security education and larger emphasis on technology transfer to the private sector. Inspired by this challenge and having experienced the difficulties involved in engineering secure server-client systems in Java, we created a prototype aiming to reduce the making of secure networking code in Java to a matter of configuration. The thought behind the project was to make it easier for Java developers to create more secure applications. Our effort is described in Paper VI in this thesis [17].

According to OWASP, the most prevalent web application vulnerability in 2004 was unvalidated input [18]. Frequent weaknesses at the time included SQL injection and Cross-site Scripting (XSS) attacks. Successful exploits alter application behavior by forcing a parser context switch, after which metacharacters can be supplied to make a program behave in unintended ways [19]. These attacks can be defended against through inspection and filtering of input and output passed to subsystems. Paper V and VII in this dissertation address the validation problem [20, 21].

Security Patterns

A pattern offers proven solutions to recurring problems. Patterns follow a template description that allow readers to get quickly familiarized with the topic at hand. Security patterns describe reappearing security problems and their well-tested solutions. Patterns show great promise in capturing and conveying security knowledge, as they go beyond describing problem solutions [22]. The context of the problem, forces likely to affect a possible solution, and common pitfalls and consequences are important parts of a pattern. Paper V in this thesis presents a security pattern for validation of input [20].

Risk Management

Risk can be viewed simply as the possibility of suffering harm or loss. This type of risk is often referred to as a *pure* or *non-speculative risk*. Examples include uncontrollable events such as natural catastrophes. A *speculative risk* on the other hand involves a conscious choice that could lead either to a gain or a loss, e.g. investing in stocks [23]. This thesis deals solely with risks from which only a loss can occur. Proper management of non-speculative risks can control and minimize the negative effects of unwanted events.

According to NIST [24], risk management is the process of identifying risks, assessing risks, and taking steps to reduce risks to an acceptable level. At a bird's-eye view, risk management can be divided into risk assessment and risk treatment. Fig. 2 presents an overview of the risk management process.

Risk Assessment

The risk assessment phase involves

System description: This first step sets the scope for the risk management process. System information and assets— such as hardware, software, network topology, and users of the system—are identified.

Threat identification: Aims to single out entities who may (accidentally or intentionally) exploit a vulnerability in the system. Examples include organized crime, script kiddies, and insiders.

Vulnerability identification: The task of creating a list of system flaws and weaknesses that could allow threats to break the system's security.

Risk evaluation Based on the likelihood of occurrence and the resulting impact, each threat/vulnerability pair is assigned a risk level.

Risk Treatment

Risk treatment concerns finding ways to deal with the identified threat/vulnerability pairs and their associated risk levels. A risk acceptance criterion signifies how much risk someone is willing to tolerate. If a given threat/vulnerability pair carries a risk below this threshold, it is left untreated and accepted as is. Risks that do not fall into the *accept* category can be handled in three ways:

- *control* the risks by introducing countermeasures to reduce the risks down to a tolerable level;
- *reject* the risks and use workarounds to avoid the identified problems; or
- *transfer* the risks to other parties through for example insurance.

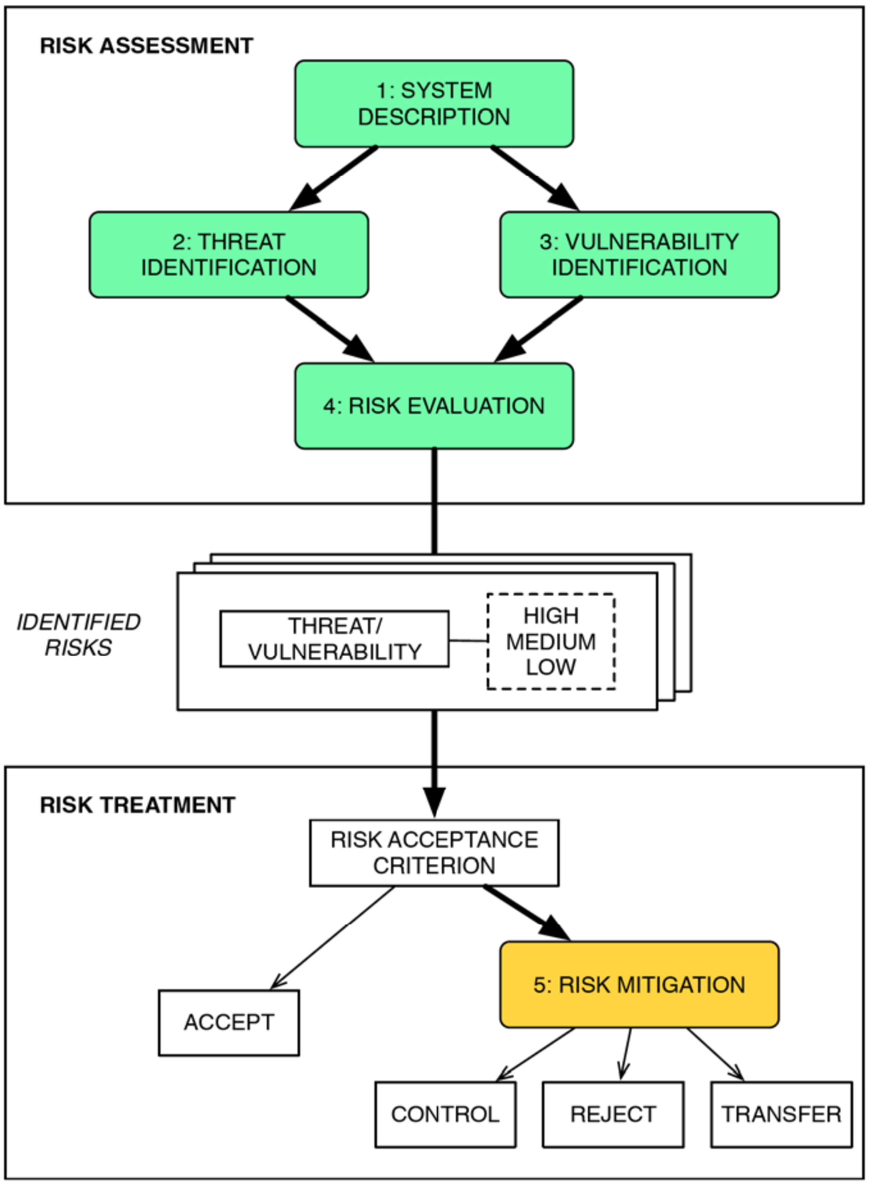


Figure 2: The Risk Management Process

The Risk Management Process

Fig. 2 shows the steps and activities involved in the risk management process. Note that steps 2 and 3 can be executed in parallel. However, they are closely related and must be paired in the risk evaluation phase. A vulnerability for which there does not exist threats that can exploit the given weakness, does not constitute a risk against a system, and should therefore be excluded. An identified threat with no matching vulnerability should be treated in the same manner.

Risk evaluations are based on the likelihood of occurrence and the negative impact resulting from a threat exercising a given vulnerability. Risks can be assessed *quantitatively* or *qualitatively*. A quantitative approach relies on mathematics to exactly express risks. Threat/vulnerability pairs are assigned probabilities of occurrence and impact values. An example is annual loss expectancies that can be derived by multiplying a potential monetary loss with the probability of occurrence during a year. It can be very difficult to quantify risks. An example is the challenge of accurately estimating loss of reputation.

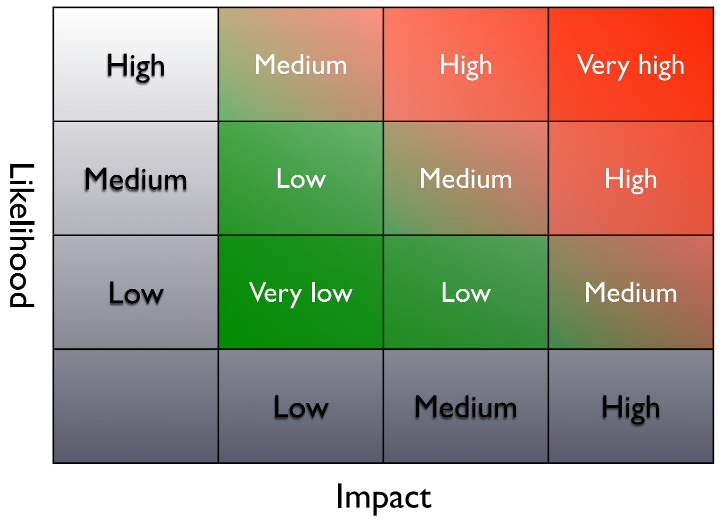


Figure 3: Five-level Risk Matrix

A qualitative approach describes risks using a hierarchical scale. As an example, likelihood of occurrence can be approximately described through three categories: low, medium, and high. A similar approach can be applied to impact estimation. Fig. 3 shows a risk matrix based on the example three categories for likelihood and impact. The illustration demonstrates a possible five-level risk matrix derived from the various combinations of likelihood and impact. In order for the risk evaluation results to be reproducible, the different risk levels must be defined. The assessments in this thesis follow the qualitative approach. In light of the choice of qualitative risk analysis, an appropriate definition is to define risk as the negative impact of the exercise of a vulnerability, considering both the likelihood and the impact of occurrence [24].

As illustrated in Fig. 2, the results of the risk evaluation phase is a set of threat/vulnerability pairs with an associated risk level. The identified risks can then be ranked and compared to the risk acceptance criterion. Different methods exist for selecting this criterion. One approach is to set a limit against which all risks are contrasted. Those ranked above the limit are subject to risk mitigation techniques. The As Low As Reasonably Practicable (ALARP) principle requires all risks to be mitigated to the point where the costs of applying further risk treatment grossly outweighs the expected benefits. This latter form of criterion has been successfully applied in the oil industry [25]. Systems in that industry involve risks with low likelihood and very high impact that should be avoided.

Residual Risks

It is important to note that all risks cannot be removed in a non-trivial system. Plans should be devised to handle these *residual risks*. An important task in this regard is to assign ownership of the remaining risks. Some system owners assume this responsibility themselves. An example is credit card companies that cover customers' losses from card misuse, as long as the bureaus themselves cannot trace the reported misuse back to the affected client. Another approach is to split the cost between the system owner and users, as widely adopted by the insurance industry in the form of individual shares. Yet another alternative is to communicate the remaining risks to customers, and let them bear the costs when losses occur. This approach is frequently used in amusement parks, where customers get risk warnings prior to entering rides.

The numbers on costs of fixing software defects, mentioned in the introduction of this dissertation, indicate that risk management is most effective early in the SDLC. However, it should be revisited and updated at later stages of development. Failure to do so may lead to uninformed decision making, as new threats and/or vulnerabilities surface, rendering previous risk analyses inaccurate or obsolete. A good risk management regime therefore plans for changes and continuously updates its evaluation of risks as threats and vulnerabilities evolve.

Risk Case Studies

Two of the papers in this thesis, "Open Wireless Networks on University Campuses" [26] and "Risk Assessment of Services in a National Security Infrastructure" [27], use elements of risk management on deployed systems, which translates to the maintenance phase in Fig. 1.

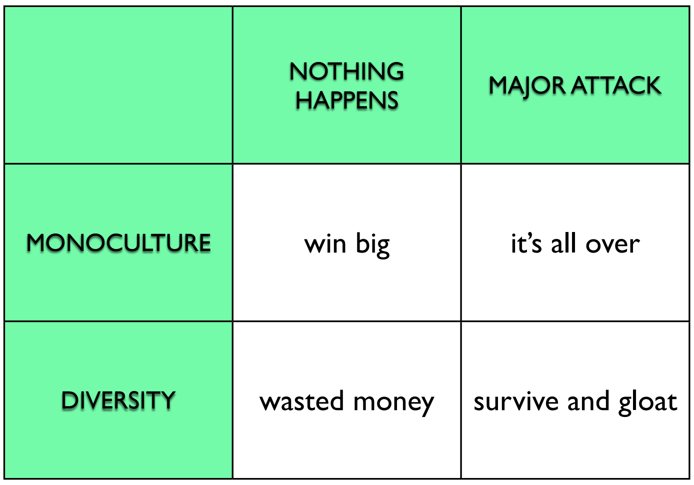
Open wireless networks

The human mind processes risk in two ways: either based on intuition, emotions, and experience; or analytically, using probability and statistics to make informed decisions [28]. In particular, fear is a powerful emotion that cause people to overestimate risks. The billions of dollars spent fighting terrorism after 9/11 is an often used example of bad decision-making based on dread. Perceived versus actual risk is a recurring theme in Bruce Schneier's monthly Crypto-Gram newsletter [29].

Upon floating the idea of an open wireless network on the University of Bergen campus, the IT department's knee-jerk reaction was fear of misuse. One of the risks highlighted was the horrifying scenario of pedophiles downloading child pornography. Another concern was related to illegal downloads of music and movies. Paper I [26] examines these and other risks introduced by an open wireless network, and suggests steps to curb unwanted behavior. The article argues that universities can miss out on good opportunities by distrusting employees, students, and citizens. A particularly interesting benefit is the increased availability of open networks versus networks that require authentication. Among the three fundamental goals of security, availability is the least investigated. For those who consider deploying an open wireless network, an important consideration is whether the increased availability and improved usability outweigh the added risks.

Internet banking

The results from the online banking risk management process make a case for the importance of risk management throughout the SDLC. The Internet threat and vulnerability landscape has changed drastically from BankID's start around 1999. While it may have been reasonable to conclude that our phishing/MitM vulnerability along with an organized crime threat did not constitute a significant risk against BankID in the beginning of the noughties, it is now safe to conclude that the identified risk was unnecessarily high during most of 2007. Attacks against similar banking systems in 2006 and 2007 support this claim [30]. The BankID community acknowledged this and took steps to remove the phishing/MitM vulnerability in November 2007 and January 2008.



	NOTHING HAPPENS	MAJOR ATTACK
MONOCULTURE	win big	it's all over
DIVERSITY	wasted money	survive and gloat

Figure 4: Monoculture versus Diversity

A comparison of Internet banking in Norway before and after BankID leads to an interesting observation. Prior to BankID, many different banks provided a variety of Internet banking solutions. Customers could do their banking with a financial institution that provided what they thought to be the best trade-off between security and usability. BankID is currently turning this diversity into a monoculture for Internet banking in Norway, as most Norwegians are expected to become BankID customers shortly. Fig. 4 shows the relationship between a diverse system and a monoculture, as described by security guru Dan Geer [31]. Following this line of thought, one way to improve the security of Internet banking in Norway would be to shut down BankID and revive the old online banking systems. Since BankID is not going away any time soon, other options for improvement must be considered. Fig. 4 illustrates that the security of Internet banking in Norway as a whole depends on the presence of several banking solutions in the market. If BankID later becomes the only Internet banking option available, an increase in security can be realized by attracting new businesses into the online banking market.

Trust

Trust is a positive expectation regarding the behavior of someone or something in a situation that entails risk to the trusting party [32]. Trust is not a binary concept, but involves many levels ranging from complete distrust to complete trust [33]. *Distrust* can be qualified with the same wording as the definition of trust, except that it is a negative expectation. The halfway point between these two concepts is *untrust*, a term that describes situations without sufficient information to either be trusting or distrusting. Another related notion is *mistrust*, which refers to circumstances where someone decides to trust a party that later turns out to be untrustworthy, or vice versa: a position of distrust is favored in a relationship with someone that is trustworthy. The latter version of mistrust translates to missed opportunities, whereas the first variant exposes the trusting party to an outright loss.

An understanding of the risks associated with a computer system is a pre-requisite for good trust management. In a risk-free environment there is no room for trust, as the actors have complete information and do not rely on others' behavior to achieve their goals. As mentioned earlier, there will always be remaining risks after a risk management process involving a non-trivial system.

Misplaced trust plays a central role in several papers in this dissertation. The online banking papers in this thesis [27, 34, 35] describe a situation where the Norwegian banking industry trusted people not to modify their client software. First in the form of two unprotected HTML parameters, later an unguarded protocol version number. It would have been wiser to distrust the users and mitigate the risks. In the end, the banks rejected the risks posed by our exploits through modifying BankID.

A position of distrust against trustworthy users of wireless networks can lead to foregone opportunities. An open network can introduce a higher degree of usability, as students, faculty, and visiting guests do not have to deal with authentication mechanisms.

Failure to properly handle and validate input and output passed to subsystems is another example of mistrust. Developers who do so rely on a position of trust in dealing with potentially untrustworthy clients. Attackers who take advantage of this can become very costly for system owners. The recommended course of action is to apply risk mitigation, and introduce validation mechanisms such as the content validation framework described in Paper VII [21].

Thesis Summary

Paper I: Open Wireless Networks on University Campuses [26]

Paper I argues that it is reasonable for universities to consider giving access to wireless networks without authentication of end users. A risk assessment and risk mitigation of a generic network model supports the claim. An open wireless network comes with the benefit of increased usability, availability, and possibly improved privacy, as network operators will find it more difficult to track users' movements and activities on campus. Our work points out several potential areas for increased risk as a result of introducing an open network: illegal downloads, attacks on the local network, anonymous attacks on remote networks, bad press, and problems related to legal requirements. Steps can be taken to mitigate the identified risks, such as filtering and monitoring of network traffic, and to establish good relationships to the media. Experience with an open wireless network at our own department shows that it is indeed possible to mitigate the risks related to misuse. It should be noted that the future legal status of open wireless networks in the EU and US is unclear. Authentication of users may become mandatory for all public networks in these regions.

Paper II: Risk Assessment of Services in a National Security Infrastructure [27]

Paper II uses elements from risk management to discuss the Norwegian financial industry's Public Key Infrastructure (PKI) for Internet banking, called BankID, from an external point of view. The authors provide a PKI primer and give an overview of the BankID architecture and design. The paper then looks at risks associated with customer authentication and discuss non-repudiation in BankID. In short, the paper finds that BankID is vulnerable to particularly efficient Distributed Denial-of-Service (DDoS) attacks on the application layer; bank customers are exposed to significant risks related to combined phishing/MitM attacks; and the absence of an independent third party gives the banking community an advantage in future non-repudiation conflicts. A number of steps suggest how to mitigate the identified risks: (i) PINs and passwords should be used only locally, (ii) vulnerabilities to well-known attacks such as phishing and MitM should be addressed immediately, and (iii) the true level of non-repudiation should be determined by independent lawyers and security experts prior to adoption on a national level.

Paper III: A Proof of Concept Attack against Norwegian Internet Banking Systems [34]

Paper III explores the combined phishing/MitM attack mentioned in Paper II. By inserting a proxy in the BankID log-in procedure, an attacker could let a bank customer complete the authentication protocol and subsequently steal the Internet banking session. The main weaknesses that allowed the attack were two unprotected Java applet parameters and an unprotected URL authorization token. Proof of concept code was developed and demonstrated for the Financial Supervisory Authority of Norway to show the feasibility of the approach. The exploit gave access to a customer account in two randomly chosen Internet banks based on BankID. The paper suggests that BankID's security could be much improved by moving to a traditional PKI solution, where private-public key pairs are stored and managed solely by customers. The authors acknowledge that such a transition may not be economically viable for the Norwegian financial industry.

Paper IV: Robbing Banks with Their Own Software—an Exploit against Norwegian Online Banks [35]

Paper IV revisits and improves the attack described in Paper III. The starting point was to reverse engineer BankID's log-in procedure, both in terms of protocol design and Java source code. The reverse code engineering exercise revealed a possible vulnerability in the generation of random numbers, affecting customers relying on Java prior to version 1.4. The combined phishing/MitM attack was updated to circumvent fixes introduced by the BankID community in November '07. A version rollback vulnerability allowed the creation of a new exploit. The paper goes on to argue that the outlined attack is particularly dangerous, as BankID's own Java Applet is used unmodified, thereby allowing an attacker to capitalize on one of the trust points in the Internet banking system. A description of our disclosure process demonstrates the Norwegian banking industry's inability to quickly address security problems pointed out by independent researchers. In closing, the paper recommends a through analysis of BankID. More weaknesses in BankID has recently been pointed out in a paper by Gjøsteen [36].

Paper V: Security Pattern for Input Validation [20]

The input validation pattern addresses the problem of attacks hidden in message content. The pattern uses the template format established in [22]. The offered solution involves performing syntactical and semantical checks of all sources of input. Known uses include Stinger [37], an input validation engine for HTTP requests. The pattern features a pencil drawing of an airport security checkpoint, which is intended to serve as a Gestalt for input validation. The goal is to use simple building blocks to convey something that is more than the sum of its parts. An excellent example and inspiration for this attempt to create wholeness is the medieval village in the 'Single Access Point' pattern [22].

Paper VI: Simplifying Client-Server Application Development with Secure Reusable Components [17]

Paper VI describes a software prototype that was implemented to reduce the complexities involved in developing secure client-server applications. The initial observations that spurred the project was the intricacies involved in creating secure networked applications in Java Standard Edition, version 1.4.2. Common network programming issues, such as choosing to act as client or server, choosing a thread model, and transport mechanism selection, were implemented and offered to other developers as a matter of configuration. The idea was to encourage reuse and offer secure networking in Java to programmers without a security background. An example on how to develop an HTTP server using the communication component is included in the paper for illustrational purposes. Paper VI also comments on the importance of openness and simplicity in order for developers of security software to be trusted by other developers and end users.

Paper VII: A Reflection-Based Framework for Content Validation [21]

In 2004, OWASP ranked unvalidated input as the most common web application vulnerability. Paper VII addresses this problem with a software prototype for validation of input and output in object-oriented systems. XML and the Java reflection API provide a flexible solution that can easily be updated to accommodate changes in the underlying risks to a particular system. The framework comes with five pre-defined categories of content validation rules, and also allows developers to specify their own rules. The paper includes an example of how to use the framework for setting up validation of a bill payment web form. The walk-through shows how to set up validation rules using ranges, regular expressions, and custom logic.

Societal Impact

Our work on BankID has been referenced and commented by many media outlets (see http://www.nowires.org/Press/Press.html). We have received both roses and thorns for our discoveries. Supporters include The Data Inspectorate and the Minister of Government Administration and Reform. They have publicly stressed the importance of independent research on national security infrastructures. The Norwegian Post and Telecommunications Authority and the Norwegian Financial Services Association have not been equally amused. Unfortunately, much of the public debate has centered around the legality of our exploits, and not on how to produce more secure computer systems. The patching of the version rollback vulnerability shows that the Norwegian financial industry can benefit from working closer with security researchers to improve security.

What Lies Ahead?

Given the broad scope of computer security problems, interesting results surface in the cross-section between research disciplines. A recent example is security economics, where results from microeconomics and game theory have been successfully applied to computer security problems. Successes include the use of *information asymmetry* to explain the prevalence of poor security in the software market, as consumers find it hard to evaluate the level of security in different products; insights from *conflict theory* suggest that companies should hire fewer programmers and more software testers; economic research on public goods suggests that government interaction is the most effective way of stopping viruses and spam [8]. These success stories indicate that new exciting breakthroughs in computer security can be found by working closely with other disciplines. Promising candidates include economy, law, and psychology.

Given the Norwegian courts historical tendency to rule in favor of the Norwegian banking community—a practice described in [38]—Norwegian bank customers are wise to pay attention if BankID is to become a national security infrastructure. Current weaknesses in BankID hint at the challenges involved in designing such infrastructures. In particular, adequate privacy seems to be to achieve, a theme that was briefly explored in [39]. A study from the US National Research Council strongly recommends strict scrutiny and in-depth deliberation long before deploying or designing a nationwide identity system [40]. The report also stresses the importance of seeking input from all stakeholders. BankID is a proprietary system owned by the Norwegian banking community that was developed for Internet banking purposes. Given the challenges involved in designing a nationwide identity system from scratch, it will be extremely difficult to retrofit BankID into a sound national security infrastructure, if not impossible. Studies of BankID's future successes and failures can provide useful insights for national security infrastructure researchers.

In terms of nationwide identity systems, it could be interesting to look into the applicability and effects of different risk acceptance criteria. Assume that BankID is chosen to become Norway's new national security infrastructure for services across a wide range of industries. Should individual banks be allowed to choose a criterion aligned with their own risk appetite, or are customers better off if the Norwegian Government mandate treatment of all risks above a certain threshold? It could also be interesting to look into the applicability of the ALARP principle in conjunction with national security infrastructures.

References

[1] R. Anderson, R. Boehme, R. Clayton, and T. Moore, "Security Economics and The Internal Market," http://www.enisa.europa.eu/doc/pdf/report_sec_econ_&_int_mark_20080131.pdf, February 2008, study commissioned by the European Network and Information Security Agency (ENISA).

[2] D. Turner, T. Mack, M. K. Low, M. Fossi, J. Blackbird, D. McKinney, E. Johnson, S. Entwisle, and C. Wueest, "Symantec Internet Security Threat Report—Trends for July–December 07," http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf, April 2008.

[3] G. Hoglund and G. McGraw, *Exploiting Software—How to Break Code*, Addison-Wesley, 2004.

[4] R. Anderson, "Why Cryptosystems Fail," in *ACM 1st Conference on Computer and Communication Security*, Fairfax, VA, USA, 3-5 November 1993.

[5] G. McGraw, *Software Security—Building Security In*, Addison-Wesley, 2006.

[6] S. Lipner, "The Trustworthy Computing Security Development Lifecycle," in *Annual Computer Security Applications Conference*, Tucson, USA, 6-10 December 2004.

[7] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, Prentice Hall, fourth edition, 2007.

[8] R. Anderson, *Security Engineering*, John Wiley & Sons, second edition, 2008.

[9] J. H. Salzer and M. D. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, 63(9):pp. 1278–1308, September 1975.

[10] S. T. Kent and L. I. Millett, editors, *Who Goes There? Authentication Through the Lens of Privacy*, The National Academies Press, 2003.

[11] J. Zhou, *Non-repudiation in Electronic Commerce*, Arctech House, 2001.

[12] M. Howard and S. Lipner, *The Security Development Lifecycle*, Microsoft Press, 2006.

[13] "OWASP CLASP Project," http://www.owasp.org/index.php/Category:OWASP_CLASP_Project, last checked May 2008.

[14] "FindBugs^{TM}—Find Bugs in Java Programs," http://findbugs.sourceforge.net/, last checked May 2008.

[15] "OWASP WebScarab Project," http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project#Download, last checked May 2008.

[16] President's Information Technology Advisory Committee, "Cyber Security: A Crisis of Prioritization," February 2005.

[17] Y. Espelid, L.-H. Netland, K. A. Mughal, and K. J. Hole, "Simplifying Client-Server Application Development with Secure Reusable Components," in *International Symposium on Secure Software Engineering (ISSSE 06)*, Washington DC, USA, 13-15 March 2006.

[18] "Top Ten 2004 — OWASP," `http://www.owasp.org/index.php/Top_10_2004`, last checked May 2008.

[19] S. H. Huseby, *Innocent Code*, Wiley, 2004.

[20] L.-H. Netland, Y. Espelid, and K. A. Mughal, "Security Pattern for Input Validation," in *VikingPLoP (Pattern Languages of Programs)*, 28 September - 1 October 2006.

[21] L.-H. Netland, Y. Espelid, and K. A. Mughal, "A Reflection-Based Framework for Content Validation," in *International Symposium on Frontiers in Availability, Reliability, and Security (FARES 07)*, Wien, Austria, 10-13 April 2007.

[22] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns—Integrating Security and Systems Engineering*, John Wiley & Sons, 2006.

[23] A. Calder and S. G. Watkins, *Information Security Risk Management for ISO27001/ISO17799*, IT Governance Publishing, 2007.

[24] G. Stoneburner, A. Goguen, and A. Feringa, "Risk Management Guide for Information Technology Systems—NIST Special Publication 800-30," Technical report, National Institute of Standards and Technology, July 2002.

[25] T. Aven and J. E. Vinnem, *Risk Management*, Springer, 2007.

[26] K. J. Hole, L.-H. Netland, Y. Espelid, A. N. Klingsheim, H. Helleseth, and J. B. Henriksen, "Open Wireless Networks on University Campuses," *IEEE Security & Privacy*, 6(4):pp. 14–20, July/August 2008.

[27] K. J. Hole, A. N. Klingsheim, L.-H. Netland, Y. Espelid, T. Tjøstheim, and V. Moen, "Risk Assessment of Services in a National Security Infrastructure," *IEEE Security & Privacy*, accepted for publication.

[28] S. Berinato, "CSO Disclosure Series — Reporter's Notebook: The United States of TMI," `http://www.csoonline.com/article/217064`, February 2008.

[29] B. Schneier, "Crypto-Gram Newsletter," `http://www.schneier.com/crypto-gram.html`.

[30] The Financial Supervisory Authority of Norway, "Risiko– og sårbarhetsanalyse (ROS) 2007," March 2008.

[31] D. E. Geer, "The Evolution of Security," *ACM Queue*, 5(3), April 2007.

[32] L. F. Cranor and S. Garfinkel, editors, *Security and Usability—Designing Secure Systems That People Can Use*, O'Reilly, 2005.

[33] S. Marsh and M. R. Dibben, "Trust, Untrust, Distrust and Mistrust—An Exploration of the Dark(er) Side," in *iTrust 2005*, volume 3477 of *LNCS*, pp. 17–33, Springer, 2005.

[34] Y. Espelid, L.-H. Netland, A. N. Klingsheim, and K. J. Hole, "A Proof of Concept Attack against Norwegian Internet Banking Systems," in *Proc. 12th International Conference on Financial Cryptography and Data Security (FC08)*, number 5143 in LNCS, pp. 197–201, Cozumel, Mexico, January 28–31 2008.

[35] Y. Espelid, L.-H. Netland, A. N. Klingsheim, and K. J. Hole, "Robbing Banks with Their Own Software—an Exploit against Norwegian Online Banks," in *Proceedings of The Ifip Tc 11 23rd International Information Security Conference*, pp. 63–77, Milan, Italy, September 8–10 2008.

[36] K. Gjøsteen, "Weaknesses in BankID, a PKI-substitute Deployed by Norwegian Banks," in *Proceedings of the 5th EuroPKI Workshop*, volume 5057 of *LNCS*, pp. 196–206, Trondheim, Norway, June 16–17 2008.

[37] "OWASP Stinger Project," `http://www.owasp.org/index.php/Category:OWASP_Stinger_Project`, last checked May 2008.

[38] K. J. Hole, V. Moen, A. N. Klingsheim, and K. M. Tande, "Lessons from the Norwegian ATM System," *IEEE Security & Privacy*, 5(6):pp. 25–31, Nov/Dec 2007.

[39] K. J. Hole, T. Tjøstheim, V. Moen, L.-H. Netland, Y. Espelid, and A. N. Klingsheim, "Next Generation Internet Banking in Norway," Technical Report 371, Institute of Informatics, University of Bergen, February 2008, `http://www.ii.uib.no/publikasjoner/texrap/pdf/2008-371.pdf`.

[40] S. T. Kent and L. I. Millett, editors, *IDs—Not That Easy: Questions About Nationwide Identity Systems*, The National Academies Press, 2002.