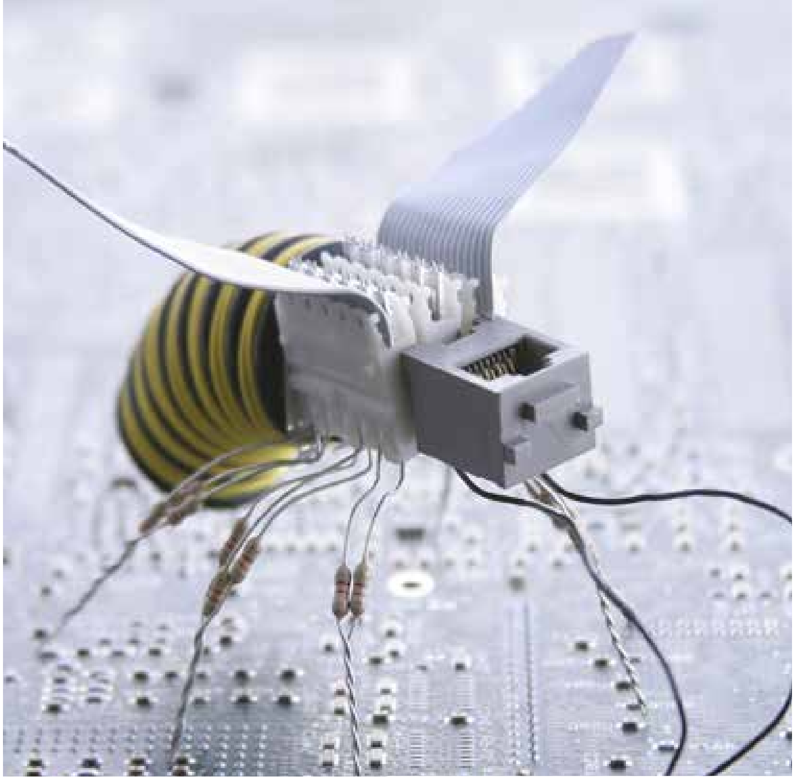


Application Security of Core Banking Systems
A first reality check

October 2012, Version 1.0.1 | A SEC Consult study

in cooperation with

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

www.sec-consult.com

SEC Consult
ADVISOR FOR YOUR INFORMATION SECURITY

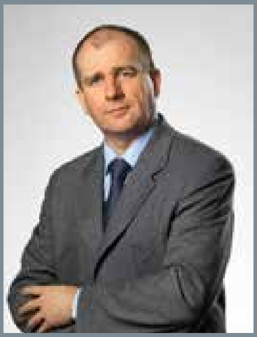
TABLE OF CONTENTS

LIST OF FIGURES

LEGAL NOTICE
Publisher and responsible for the content: SEC Consult Unternehmensberatung GmbH in cooperation with Capgemini Consulting Österreich AG
Editorial work: SEC Consult Unternehmensberatung GmbH in cooperation with Capgemini Consulting Österreich AG
Typesetting and layout: corporate identity prihoda gmbh
Photos: (unless otherwise indicated) iStockphoto.com
Print: Druckerei Janetschek GmbH

This publication contains information in summary form and is therefore intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. SEC Consult Unternehmensberatung GmbH and Capgemini Consulting Österreich AG cannot accept any responsibility for loss occasioned to any person acting or refraining from action as a result of any material in this publication. On any specific matter, reference should be made to the appropriate advisor.

1 | WELCOME



Markus ROBIN
General Manager, SEC Consult

I am very proud to present this SEC Consult study on application security in Core Banking Systems (hereinafter named CBS for short). This study offers the reader an initial appraisal of the status of application security in Core Banking Systems and the corresponding risks.

Over the last decade online security has become an increasingly important issue to the banking sector. On an almost daily basis the media confronts us with stories of hacking, data breaches and new, critical security vulnerabilities found in 'toxic' software products (e.g. operating systems, content management systems, ERP-systems, access control systems etc.). But what about the security of the most mission critical system a bank can buy from a vendor? What about the application security of the Core Banking Systems on the market? Our preparatory investigations showed that in terms of application security, there is a real shortage of quality information that might indicate the maturity of the Core Banking Systems market.

The aim of this study is to remediate this shortage of information by providing an insight into the maturity of application security in major core banking packages and the related risks for banks and software vendors. We offer an assessment of application security by comparing the commitments and promises of the vendors' with the results of hands-on security tests conducted by our computer security professionals.

This report is aimed at two key audiences. The first includes all banking professionals involved in the selection, procurement, implementation, testing and security management of core banking packages. The second includes professionals from core banking vendors involved in product management, software development, testing and security management.

I would like to thank all the participating vendors and all SEC Consult colleagues involved for their efforts and contributions. I would especially like to thank our partner Capgemini for supporting this study with their in-depth experience and knowledge of the core banking vendor market.

Last but by no means least I would like to thank the three participating banks for providing us with the opportunity to test their Core Banking Systems. This allowed our computer security experts to perform real life, hands-on testing, without which we would not have been able to achieve our goal of uncovering and highlighting critical issues in the application security of Core Banking Systems.

The process of carrying out this study has highlighted the value of transparency, with regards to application security, to both banking customers and core banking vendors alike. In this respect the study has already proved itself to have been a successful initiative and has become a building block for the continuous improvement of applications security of Core Banking Systems.

If you require any further information or assistance relating to this study, please do not hesitate to contact us directly or via our local offices. Any feedback, comments or advice on the improvement of this study is warmly welcomed.

Markus ROBIN
General Manager, SEC Consult

2 | STUDY APPROACH

The survey focused specifically on Core Banking Systems (hereinafter named CBS for short) for top-tier banks that hold leading, global positions in financial markets and vendors that offer these CBS across various countries in Europe. Although most of the selected vendors (Avaloq, FIS and FIS KORDOBA, Infosys, Misys, Oracle, SunGard, TCS Financial Solutions) participated in this study, there were unfortunately some which did not take part (Callataÿ & Wouters, Delta Informatique, SAP and Temenos).

In the first part of the study (Part 1) we asked the vendors about their promises, commitments and relevant activities relating to the application security of their product. A questionnaire consisting of 52 questions was sent to all vendors. We recommended that the person responsible for IT security should answer the questions or at least conduct a quality assurance exercise of the questions and answers. All vendors followed this recommendation. The methodology for this part of the survey was based upon commonly known security standards, best practices, guidelines and the experience of the study's authors. Seven vendors with eight products participated in the first part of the study.

To validate the results of the survey we originally intended to conduct the second part of the study (Part 2) by conducting security 'crash testing' on the products of the vendors we had surveyed. We offered the vendors a (free of charge) security 'crash test' to be conducted by our computer security professionals (using the types of techniques that potential 'hackers' would use) on a test system provided by the vendor. Certain vendors showed a serious interest in participating in this test but unfortunately, after certain deliberations, all vendors declined to participate.

While developing an alternative approach to the second part of the study we managed to gain support from the financial service industry. Helpfully, three banks allowed us to perform a security 'crash test' on their CBS, systems which were already implemented by the banks in question. As a result at least three of the products described in Part 1 have been tested in the (alternative) second part of the study. Figure 1 summarizes the vendor and product participation in this study.

In the interests of the security of both the banks which participated in the application security test and all other customers using similar product releases, we will not disclose the names of banks, vendors or products that have been tested for security in this study.

This study was written, conducted and compiled over nineteen month period between September 2010 and March 2012.

Vendor	Product	Part 1 Survey with questionnaire	Part 2 Security crash test system at vendor	Part 2 Security crash test system at bank
Avaloq	Avaloq	✓	×	Crash tests have been performed for three products participating in part one of the study. To protect the participating banks the vendor or product names that have been tested for application security will not be disclosed in this study.
Callataÿ & Wouters	Thaler	×	×	
Delta Informatique	Delta-Bank	×	×	
FIS KORDOBA	K-CORE24	✓	×	
FIS	Profile	✓	×	
Infosys	Finacle	✓	×	
Misys	BankFusion Midas	✓	×	
Oracle	FLEXCUBE	✓	×	
SAP	SAP TB	×	×	
SunGard	Ambit CBS	✓	×	
TCS Finiancial Solutions	TCS BaNCS	✓	×	
Temenos	T24	×	×	

Figure 1: Invited vendors and products

3 | ABOUT SEC CONSULT

SEC Consult is an international leader in application security services and consultancy in the financial services industry. SEC Consult's competence in improving the application security of enterprise applications supports major international banks and global software vendors. We provide consultancy and specific, high-end services such as security quality gates, Secure Software as a Services (SSaaS) or Managed Vulnerability Information Services (MVIS) which help to protect our clients from 'toxic' (i.e. heavily insecure) enterprise software. Our many years of experience with the remediation of application security problems in banking software and systems (online-banking to Core Banking Systems, ATM software, trading software, etc.) allow us to help software vendors and banks to reduce their risk of application security vulnerabilities.

As a leader in application security services and consultancy in the financial services industry SEC Consult supports banks and software vendors in the following areas of Core Banking Systems.

For banks SEC Consult offers support for:

- Systematic assessment of the vendor commitment on application security (proven assessment method) in CBS.
- Security crash-testing CBS to validate the claims of CBS providers.
- Establishment of security quality gates for each new CBS release.
- Support with guidance and application security requirements for the procurement of CBS.

For software vendors SEC Consult offers support for:

- Security crash-testing of CBS to identify points at which the software fails to achieve state-of-the-art application security.
- Systematic and comprehensive application security testing on the development of new functionality and modules.
- Establishment of security quality gates for each new CBS release.

If you would like to know more about our services we would be very pleased to speak to you through our local offices or via email at office@sec-consult.com.

Contacts:

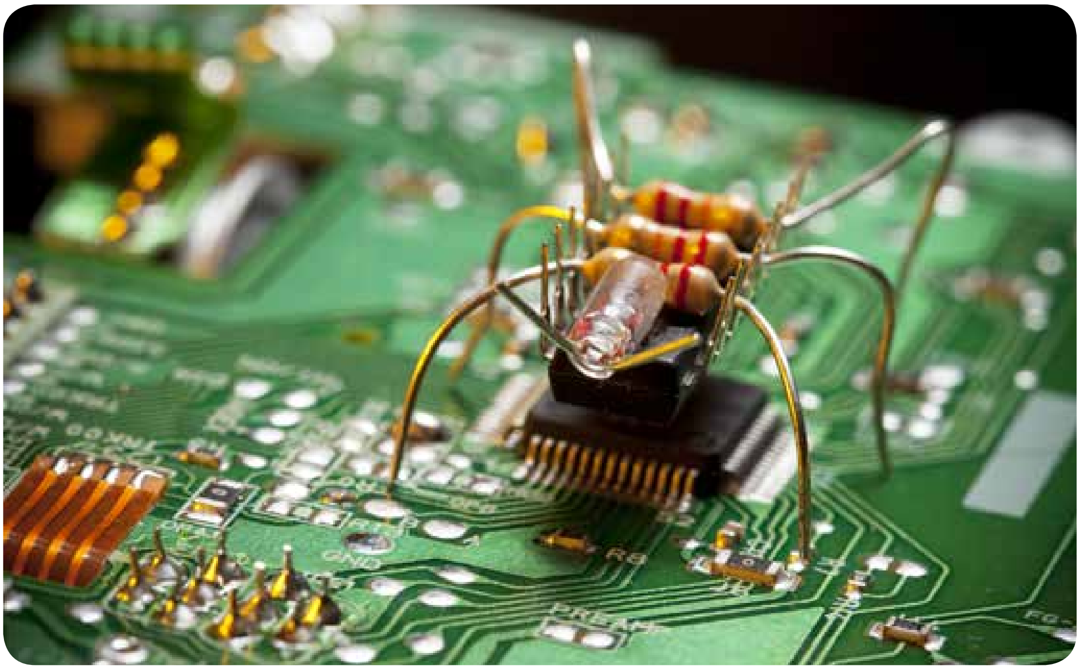
Website: www.sec-consult.com

Country/Region	Phone	Email
Austria	+43 1 890 30 43 0	office@sec-consult.com
Canada	+43 1 890 30 43 0	office-montreal@sec-consult.com
Germany	+49 69 175 37 34 3	office-frankfurt@sec-consult.com
Lithuania	+370 5 219 55 35	office-vilnius@sec-consult.com
Singapore, Asia/Pacific	+43 1 890 30 43 0	office-singapore@sec-consult.com
All other regions	+43 1 890 30 43 0	office@sec-consult.com

4 | THREAT LANDSCAPE

In the last few years the threat landscape concerning cyber-attacks on financial services organizations has changed dramatically and systematically.

Firstly, the profile of attackers is no longer confined to that of 'computer freaks' whose prime motivation is to demonstrate their intellectual superiority. Instead, today's threat landscape comprises of a broad spectrum of attackers.



Script kiddies use easily accessible attack software to perform visible and destructive attacks, without the need for expert security know-how. Disgruntled employees use their insider knowledge to blackmail their employers. Criminals use highly specialized value chains to prepare and conduct cyber-attacks and to transfer and manage cash flow relating to illegal activities. Industrial and governmental espionage groups use invisible, sustained attacks to harvest interesting data and know-how in order to use this information to their advantage. The term 'Advanced Persistent Threat (APT)' is increasingly used to describe attacks of this kind in which a specific entity is persistently and effectively targeted. Military cyber warriors screen foreign targets to prepare for future conflicts which no longer occur on the physical battlefield. Terrorists have steadily improved their ability to use cyber space for their operations.

Secondly, attackers are no longer approaching targets solely via the Internet. They attack from the internal network of branches or affiliates, contractors, partners, consultants or vendors. One single piece of intelligent, invisible malware installed in the computer of one client within a corporate network is sufficient to enable direct access to the internal network. Therefore, assuming that the internal network (intranet) is by default a secure space protected by firewalls is an outdated paradigm. Today internal information, applications and infrastructure of the intranet demand greater protection and security.

Most data breaches in banking are targeting data from back office and core systems including client statements, transactions, account balances etc. Therefore the overwhelming majority of successful attacks will not be reported due to the fact that they are either invisible or due to the fear of loss of reputation on the part of the victim.

> ‘Internal sabotage, clandestine espionage or furtive attacks by trusted employees, contractors and vendors are potentially among the most serious risks that a bank faces. […]’
>
> *Internet Banking and Technology Risk Management Guidelines, Monetary Authority of Singapore*

Thirdly, attackers have broadened the scope of their activities from simple fraud (phishing etc.), to the accessing and use of personal and behavioral data of banking customers to support multistage espionage activities. The financial crises has underlined the systematic importance of a stable banking sector and as such several countries have listed the banking industry as critical infrastructure that requires protection against cyber threats from foreign powers or terrorists. So called ‘Denial of Service’ attacks are a rising threat not only for banking websites but also for banking operations as a whole.

In 2010, the US think tank the Bipartisan Policy Center conducted a simulated cyber-attack in the USA called ‘Cyber ShockWave’. One of the attacks in the simulation was an ongoing, random change of the account balances of millions of citizens to spread fear, uncertainty and doubt and to jeopardize the trust in the banking sector. The fact that this study was carried out illustrates how seriously these issues are being taken.

The ‘asymmetry’ of the relationship between attackers and victimized organizations puts targets of cyber attacks at a systematic disadvantage. In order to defend itself against this type of attack an organization must seek out and repair the vast majority of its vulnerabilities while an attacker need only discover a small number of them in order to mount an effective assault. Thus a large attack surface (i.e. many applications, many systems and components, etc. in the intra- and extranet) provides an advantage to the attacker and a disadvantage to the targeted side, creating an ‘asymmetry’. Additionally, attackers are able to prioritize their search for vulnerabilities, hunting for the weakest points first. This means that continuously searching for potential or existing vulnerabilities will ultimately pay off for the cyber attacker.

Fourthly and probably most importantly: Security experts at a global level are in consensus that the overwhelming majority of security loopholes are found in software/applications themselve. This area of information security has been named application security.

Definition: Application security is a process performed to apply controls and measurements to an organization's applications in order to manage the risk of using them. Controls and measurements can be applied to the application itself (its processes, components, software and results), to its data (configuration data, user data, organization data), and to all technology, processes and actors involved in the application's life cycle.

ISO/IEC 27034-1:2011

Some cyber-attack experts are already issuing warnings about an 'epidemic' of these security vulnerabilities as their total and relative volume increases at an exponential rate. Figure 2 shows the lifecycle of a successful attack on the level of an application built using a given software.

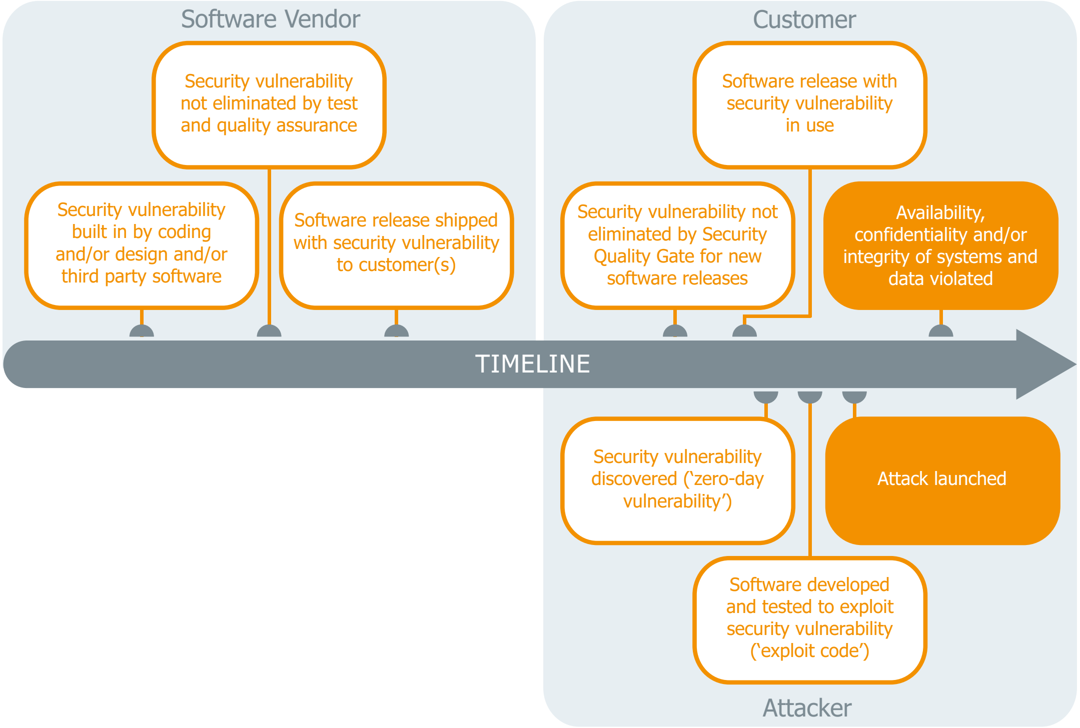

Software Vendor
Customer
Security vulnerability not eliminated by test and quality assurance
Security vulnerability built in by coding and/or design and/or third party software
Software release shipped with security vulnerability to customer(s)
Software release with security vulnerability in use
Security vulnerability not eliminated by Security Quality Gate for new software releases
Availability, confidentiality and/or integrity of systems and data violated
TIMELINE
Security vulnerability discovered ('zero-day vulnerability')
Attack launched
Software developed and tested to exploit security vulnerability ('exploit code')
Attacker

Figure 2: Lifecycle of a successful attack on application level with new security vulnerability ('zero-day')

The diagram emphasizes the point at which the attacker discovers the security loophole and the subsequent actions which follow. The security vulnerability in a software product is constructed/integrated and consecutively not eliminated by the vendor of the software product. The responsibility for security vulnerabilities in a shipped software product is therefore with the vendor.

'Security vulnerabilities in software are quality deficiencies of the respective software vendor.'

SEC Consult, 2009

On the customer side there is one final opportunity to identify vulnerabilities before implementation of the software by performing a security quality gate for new releases of the relevant software application. If the vulnerability fails to be identified at any of these stages and the software release containing the security vulnerability is implemented, it is up to the attacker when and how the attack is launched.

5 | TOXIC SOFTWARE

'Toxic Security' is a term that was coined in the aftermath of the subprime meltdown to describe financial instruments which cannot be readily identified as an asset or a liability. 'Toxic Software' is a term introduced by SEC Consult in March 2011 to describe similarly dangerous pieces of software or applications which contain serious vulnerabilities.

> 'Toxic Software which contains severe security vulnerabilities has a high probability to seriously harm the confidentiality, availability and integrity of its owner.'
>
> *SEC Consult, 2011*

In June 2011 the U.S. Department of Homeland Security published a set of new guidelines which incorporated a 'dictionary' of software weaknesses and associated mitigation practices developed by experts from government, industry and academia from across the software security community. This is just one indicator of the rising concerns regarding the impact of insecure, 'toxic' software.

Defining a clear strategy and enforcement of policies, processes and tools to avoid the procurement of toxic software by customers and to prohibit shipment of toxic software by software vendors is a major challenge and will remain so for at least the next two decades.

6 | MOTIVATING FACTORS FOR EMPLOYMENT OF EFFECTIVE APPLICATION SECURITY IN CORE BANKING SYSTEMS

This section details a number of important reasons why effective application security in CBS must be developed and maintained. These include the need to stay abreast of a changing threat landscape, the need for compliance with regulations and a developing understanding of application security issues.

The primary factor must be that all CBS are protected by state-of-the-art application security. The hackers of today penetrate applications using state-of-the-art skills, tools and knowledge. To protect themselves against these threats CBS must then ensure that their application security is equally near to the cutting edge.

There are a number of guidelines and standards already in place which define the requirements of state-of-the-art application security (e.g. Open Web Application Security Project (OWASP), WASC, BSI ISi-Reihe (Germany), ÖNORM A 7700, PCI-DSS etc.) and further examples are in the process of being written (e.g. ISO/IEC 27034).

Of particular importance is the ÖNORM A 7700 (Technical requirements concerning the security of web applications, hereinafter named A7700 for short) which provides standards which banks should use when considering secure web applications and web technology.

The A7700 is a set of requirements for the process of procuring and/or implementing applications with web technology in the front- and backoffice. The A7700 standard was derived from the global OWASP project’s vast knowledge pool on web application security and was refined with the cooperation of leading international banks to define a technology independent, concise and easy-to-use baseline for web applications security.

The second factor involves ensuring compliance with regulatory requirements relevant to CBS products. OCC 2008-16, Basel II, SAS70, ISO/IEC 27001, current national data privacy protection laws, national data breach notification laws, national banking specific laws etc. have mandatory or recommended requirements on the handling of information by CBS products. Regulations pertain to such issues as:

- Protection against unauthorized access
- Segregation of duties
- Auditing and logging
- Strict access control (no privilege escalation)

> ‘This bulletin reminds national banks and their technology service providers that application security is an important component of their information security program. […] Vulnerabilities in applications […] increase operational and reputation risk […].’
>
> *OCC 2008-16, Office of the Comptroller of the Currency, U.S. Department of the Treasury, 2008*

Security vulnerabilities caused by failure to adhere to these requirements put CBS at risk not only from hackers but from legal censure and are therefore especially deserving attention. It should however be stressed that while compliance with all relevant regulatory and legal requirements is mandatory, it is not per se sufficient to achieve state-of-the-art application security for a specific application.

Future regulations such as the EU General Data Protection Regulation proposed in January 2012 suggest fines of up to 2% of an organization's total global turnover if it has failed to implement appropriate security measures. The implementation of these regulations will make compliance an even more important issue than it currently is.

The third motivation for employing effective application security derives from a paradigmatic shift in our understanding of application security; the understanding that a secure application needs not only security features but also secure features.

> 'An automobile's crumple zone is not a counterbalance against defects in the vehicle's frame, nor is a seat belt substitution for the quality of the vehicle itself. These are protective features in addition to quality engineering not in spite of faulty manufacturing.'
>
> *David Rice, Geekonomics – The Real Cost of Insecure Software, 2008*

This means that e.g. an insecure feature of a CBS can give direct access to the database and therefore circumvents the security feature of role-based access rules. Consequently the attack surface of a CBS consists of all accessible layers and interfaces of the application. Figure 3 gives a simplified view on the attack surface of a CBS.

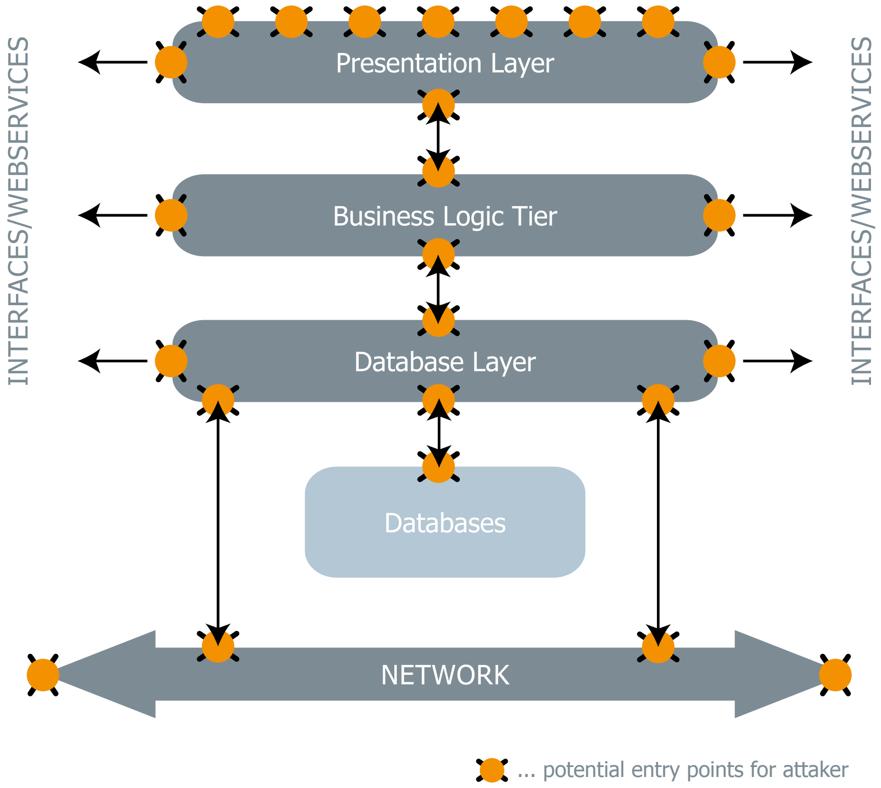


Figure 3: High level view of attack surface of a CBS

The integrated approach to web services and the transformation to service oriented architectures (SOA) are another motivating factor for the development of application security. Web applications and web services use the most advanced technologies. These technologies are potentially vulnerable to a very broad and well understood spectrum of attack vectors which must be protected against. Easing the integration effort with web services without simultaneously adapting their protection results is precisely the kind of weak links that attackers are searching to exploit.

Additionally the transformation of CBS products to integrated packages including online banking and branch functionality leads to the situation in which successful critical attacks on online or branch modules must be contained and prevented from reaching all back-office parts of the application.

The fifth factor we see is the continuously growing level of expectation and awareness regarding information security within the banking sector. Specialized external consultancies are working closely with banking departments, offering support with the latest methodologies and knowledge on information security in order to define clear security requirements and test them in the acceptance phase.

The sixth factor is that late 'bug fixing' of a security vulnerability in an already shipped software product puts all clients of this software release at risk. It increases the costs of remediation to both vendor and client and is therefore not only a risky but ineffective approach. Additionally, many security loopholes based on insecure design or inappropriate security requirements are more difficult to repair. Therefore the simple removal of a single bug is not sufficient. Significant redesign or refactoring of the application itself is necessary which exceeds a simple security hot fix for the application.

7 | MATURITY OF APPLICATION SECURITY OF CORE BANKING SYSTEMS

The following section summarizes the vendor responses of the of first part of the study about their promises, commitments and relevant activities relating to the application security of their product.

State-of-the-art application security

A secure CBS demands a vendor with an unquestionable commitment to state-of-the-art application security. All vendors define the maturity of application security for their CBS product as high, mature or highly sophisticated and therefore commit themselves to achieving state-of-the-art application security. This is a clear and consistent objective of the market.

Information security of vendor organizations

A secure CBS demands a vendor organization managing information security properly. Most CBS vendors have implemented an Information Security Management System (ISMS) which covers specific areas and processes within their organization. They follow the international standards ISO/IEC 27001 and some also have or will have these management systems certified by an independent third party. Many of them also contain the software development processes within the scope of their ISMS. Additionally, vendors seem well aware of the importance of Business Continuity Management and the integrity of employees. This catalogue of measures demonstrates the familiarity that vendor organizations have with the topic of information security and the necessity of the management of threats, risks and remediation actions.

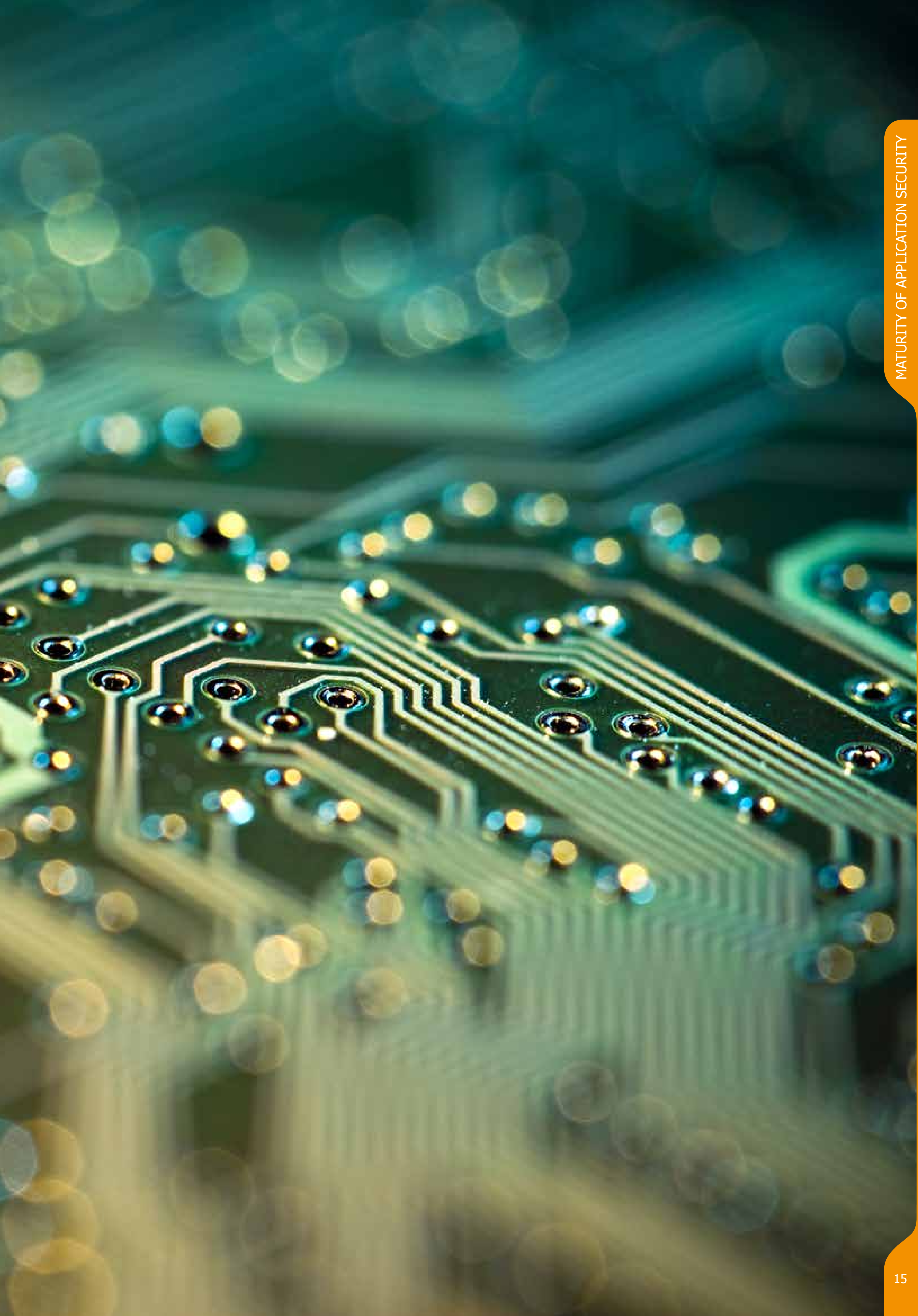
Software development organizations

The construction of a secure CBS demands software developers, who understand how to develop all aspects of the software with security in mind. In 2009 the overwhelming majority of vendors trained 90-100% of (core) development staff in application security. This statistic is far above software industry average. Additionally most of the vendors have stated that they have documented the roles and responsibilities in the development process in accordance with security policies.

Methods for secure software development

Developing a secure CBS demands not only the incorporation of security into all aspects of the software but also into the software development process itself. This requires a structured, integrated methodology for secure software development during routine development work.

The practice of secure software development (by the enforcement of systems such as Microsoft SDL, OpenSAMM, BSIMM, CMM-SSE etc.) is becoming increasingly well known throughout the software industry. However, only some of the vendors stated that they have enforced a selection or combination of these methods. Without comprehensive implementation of such methods, systematic and reproducible application security is difficult to achieve. Application security is key to all phases of the software development process from requirement to architecture, design, coding, test, deployment and maintenance. A number of vendors must improve their methodology in these areas in order to establish a solid base for application security.



Threat modeling and security requirements

Development of a secure CBS demands knowledge of existing internal and external threats and the means by which they must be mitigated through security requirements. The majority of the vendors stated that they have an up-to-date threat model for each CBS module available. This response was stronger than expected and shows an exceptional strength of understanding within the industry. Given this level of understanding, we see potential for some CBS vendors to improve their high level security requirements in the areas of integrity and confidentiality. Unambiguous security requirements in these areas are key to a trusting, transparent relationship between vendors and customers of CBS products.

Size and complexity of Core Banking Systems

CBS are large and complex enterprise applications in which the security of the whole requires state-of-the-art application security throughout every component. Figure 4 illustrates the relative sizes of various CBS in terms of the number of millions of lines of source codes out of which each is built.

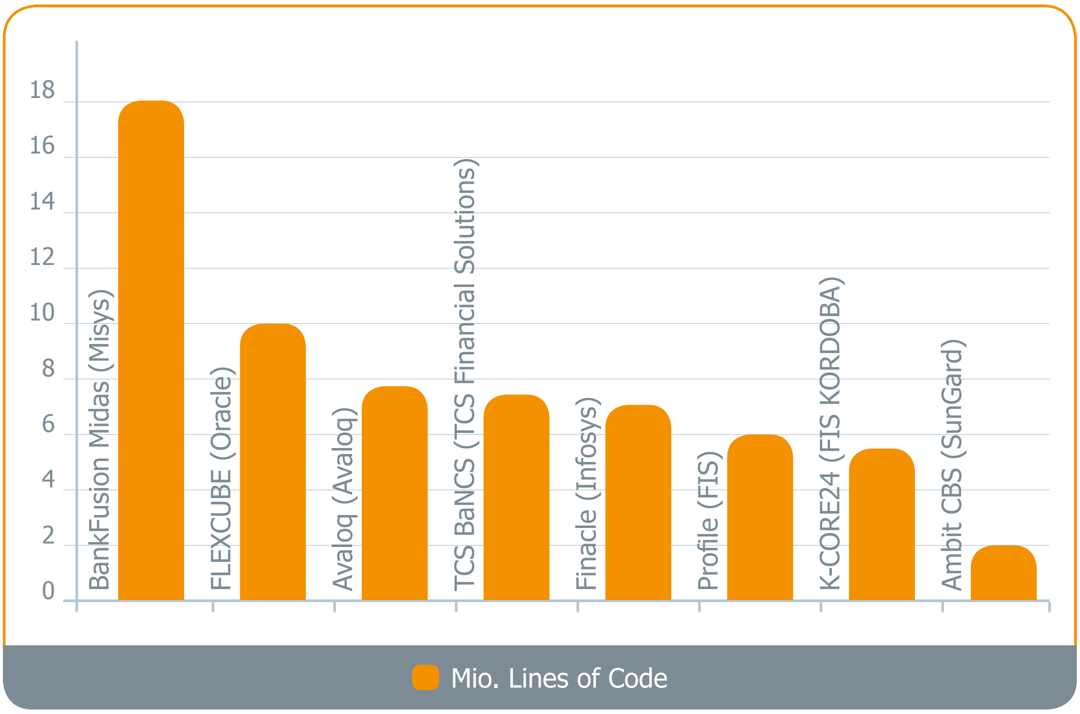


Figure 4: Approx. size of Core Banking Systems in million lines of source code of the software
Source: CBS vendor responses

In principle, the larger a system is, the larger its attack surface will be and the more difficult it will be to securely maintain. Due to this relationship between size and vulnerability managing the size and complexity of a CBS will remain a persistent challenge for vendors and customers.

Security Testing

A secure CBS demands security tests with high coverage of code base and a high level of assurance that security loopholes are eliminated. A comprehensive and detailed application security test is the last possibility to identify quality problems in defined requirements, design and coding of the CBS before shipping.

To test the necessary assurance level for application security, three levels of test depth can in principle be applied:

Security test approach	Description	Assurance level for application security
Black-box test	Test performed without access to source code in an automated and/or manual way by experts in application security.	Low
Automated, static security source code analysis ('Security Code Scan')	Scanning of source codes with tools to find indications for security vulnerabilities.	Low – Medium
Automated and manual security source code reviews and tests ('Security Code Review')	Automated and manual security review of source code and validation with dynamic tests by application security experts.	High

Figure 5: Test approaches for application security (generic overview)

The extent to which the different vendors subjected their software to these various test types is summarized in Figure 6. As the table shows, the extent of coverage varies significantly from vendor to vendor and from product to product. On the one side, certain vendors report 100% coverage of the full code base (i.e. the full functionality and all software modules of the CBS) using all three approaches. On the other, some of the vendors do not cover the majority of the code base or even do not perform security tests with medium to high assurance levels at all.

Vendor	System	Black-box Test	Security Code Scan	Security Code Review
Avaloq	Avaloq	○	●	◕
FIS KORDOBA	K-CORE24	○	○	◑
FIS	Profile	●	●	●
Infosys	Finacle	●	◕	●
Misys	BankFusion Midas	◔	○	○
Oracle	FLEXCUBE	○	◑	●
SunGard	Ambit CBS	●	◕	◑
TCS Financial Solutions	TCS BaNCS	◕	●	◔

● ... approx. 100 % coverage of CBS source code ○ ... approx. 0 % coverage of CBS source code

Figure 6: High-level comparison of test approaches and coverage of source base
Source: CBS vendor responses

The significant differences in the quality of the application security testing, as shown in the above table, imply that critical security loopholes may not be identified before certain products are shipped to market. Some vendors will have to improve in these areas to establish a solid base for application security for their customers. Additionally, all vendors must be prepared to prove their testing approaches as banks will increase the frequency of security crash testing of their CBS. In a bank's evaluation of a potential new CBS its vendor's approach to security testing should be one of the key criterion by which the bank verifies a vendor's application security capabilities.

Identified Security Vulnerabilities

A secure CBS demands structured identification and remediation of security vulnerabilities. Our survey showed a broad variation between vendors in terms of the number of security vulnerabilities that have been identified within their software packages.

Of the vendors that provided data on numbers of security vulnerabilities identified over a specific 30 month period, some reported having not encountered any security vulnerabilities, some only a handful and some more than 100. This confirms the significant differences in the quality of the application security testing and the coverage of the tests, implying yet further that inadequate testing may result in the failure to identify critical security loopholes before a CBS product is shipped to market.

In addition to the approach vendors take to security testing, the volume of previously identified security vulnerabilities should be key to evaluating a Core Banking System and verifying the application security capability of its vendor. Once again a number of the vendors must improve in these areas to establish a solid base for application security.

Security Incident Response

A secure CBS demands rapid communication of security incidents to customers to ensure necessary mitigation of risk linked to security vulnerabilities in the CBS software. Most of the vendors have a Software Security Incident Response Process with a procedure and communication channel in place to inform customers about security vulnerabilities. A few vendors exhibit such preparedness despite seeing the channel as redundant, having never once had to use it to report an incident to a customer.

(Technical) standards and best practices for application security

A secure CBS demands the knowledge and usage of existing technical application security standards and guidelines. (Technical) application security best practices and standards for web technologies like OWASP, ÖNORM A 7700 (Security requirements for web applications), etc. are already important for vendors and this importance will increase. Data privacy standards for applications like EuroPriSe are not yet well known or understood. Until now there are no certifications conducted on application security though this may change in the future depending upon consumer demand.

8 | SECURITY CRASH TEST OF SELECTED CBS PRODUCTS

The following section summarizes the results of security 'crash tests' on three Core Banking Systems out of the first part of the study.

As stated before this study was planned in two parts, the first being a survey of vendors covering specific criteria and the second involving a security crash test of the products provided by the vendor with a respective test system.

Despite offering vendors a free security quality check from our specialized security consultants in return for their participation in the second part of the study, we did not receive a single so called 'Permission to Attack'. Some vendors did show serious initial interest but did not follow through. As a result we were not permitted access to any of the CBS test systems.

However, in the process of conducting the study we attracted the interest of a number of banks which were using CBS that had been included in the survey. Finally, three banks allowed us access to their test systems with a respective 'Permission to Attack'. Thanks to the cooperation of these banks we acquired the material that we required (i.e. not only paper-based answers but real-world experience of CBS application security).

As the security tests were not performed for all of the surveyed vendors and the scope of the security crash tests carried out were limited, a direct comparison between the systems is not shown in this study.

The crash tests were performed on three products from vendors included in the first part of the study. The tested CBS versions have been in operation at their respective banks. Although some of the products tested were not the latest release of the software in question, all security upgrades provided and recommended by the vendor for these releases had been applied. As such all tested systems fairly represent the banks' actual setup and the crash test is a similarly accurate simulation of the threats a bank might be facing if insecure, 'toxic' software is implemented.

The approach taken is referred to as a 'blackbox', meaning the security test was performed without access to source code being granted to the experts while they conducted the security test.

A 'blackbox' approach is limited to a rather low assurance level for application security but gives a good first indication of the level of insecurity, if obvious security issues exist. The application security experts were provided with a low-privilege, standard user account (e.g. of a back office assistant role) to test the escalation of privileges.

The effort invested in the test was very limited compared to the multimillion lines of source code which comprise a CBS. Statistically speaking approximately 20 person-days per CBS have been used to produce the results below, touching less than 1% of the functionality of a CBS. Admittedly, this is a limited approach but this type of crash test is only intended to gain a first and quick impression of the maturity of application security.

For all three of the tested products severe and critical security vulnerabilities had been found. None of these vulnerabilities had been discovered and remediated by the quality assurance of the respective vendors. Rather, all three vendors have shipped their software release to the banks with the hidden vulnerabilities still included. All security vulnerabilities described below are proven, exploitable loopholes. Used by an attacker, the majority of these vulnerabilities can seriously damage the bank that is operating the vulnerable CBS.

In order to protect participating banks as well as other customers using a similar product release we will not disclose the name of the bank, the vendor or the product names that have been tested for security in this study.

Vulnerabilities identified in security crash test

The following security vulnerabilities have been found in the three security 'crash tests' conducted in this study on releases of CBS products at three customer banks:

1) Cross Site Scripting (XSS) – Stealing the identity and spy a CBS user

A Cross Site Scripting security vulnerability is used to steal the identity information of a CBS user. First the attacker writes an email to this user with a malicious link, including hidden script code (a very short software program). The user receives the email and clicks on that link. The malicious script runs in (the context of) the web browser of the attacked user. With this (client side) script the attacker is now able to remotely control the web browser. Due to this Cross Site Scripting security vulnerability of the CBS, the attacker can do whatever the user is allowed to do. The attacker can record all of the user's activities or initiate changes (e.g. change target account numbers of a transaction on the fly).

2) Privilege Escalation – Become a more powerful CBS user

A Privilege Escalation security vulnerability is used to give low-privilege users additional rights to become a more powerful user of the CBS. In the tested CBS low-privilege users are presented with the same control menus as higher-privilege users, except that certain items to which they should not have access are hidden (but not actually withheld) from them. By simply changing certain 'clear text parameters' sent from the Internet browser to the server the attacker reveals the menu items which should be concealed to them and becomes a more powerful user as well as gaining access to administrative functionality. With this achieved the attacker can misuse the CBS by performing high-privilege transactions and functions. This vulnerability is a general design flaw in the respective CBS.

3) Weak encryption - stealing the password of a CBS user

A security vulnerability based on weak encryption is used to gain access to user's account data. First the attacker traces the data traffic between the CBS client and the CBS server. Due to the weak encryption security vulnerability of the CBS the attacker can bypass the login mechanism and take control of the user's account. This vulnerability is a general design flaw in the respective CBS.

4) SQL-Injection – Direct reading of the database of CBS

A Standard Query Language (SQL) Injection security vulnerability is used to directly extract data from the database without further authorization. Firstly, the attacker prepares a script code (very short software program) including database statements for later use. Due to this SQL Injection security vulnerability of the CBS, the attacker can enter this script code in a data field which is originally a harmless field such as a search expression or an amount in Euros etc. Due to insufficient input validation in the CBS software with respect to this harmless field, the injected script code bypasses the authorization mechanisms and extracts valuable data from the database the attacker is interested in.

5) Direct OS Command Execution – Remote control of the server of the CBS

A direct Operating System (OS) Command Execution security vulnerability is used to get full control of the system running the CBS. Firstly, the attacker prepares a script code (very short software program) containing operating-system commands for later use. Due to this direct (OS) Command Execution security vulnerability of the CBS, the attacker can enter this script code in a data field which is originally a harmless field such as a search expression or an amount in Euros, etc. The injected script code bypasses the authorization mechanisms.

Contrary to the SQL Injection as described above, direct Operating System (OS) Command Execution allows the attacker not only access to the database but to gain control over the operating system of the server of the CBS as well. The system can be shut down, wiped or manipulated by the hacker using unsuitable data. Data from the server can be copied to a repository belonging to the attacker. Additionally, this vulnerability can be used to attack other systems belonging to the bank.

All three of the CBS releases, we tested, exhibited failings in terms not only of surprisingly poor quality assurance for application security but also by way of significant issues with insecure coding and inappropriate security design.

Due to the identified vulnerabilities these three CBS releases fail to comply with several commonly accepted security standards for (web) application security [e.g. Open Web Application Security Project (OWASP), WASC, BSI ISi-Reihe (Germany), ÖNORM A 7700 (Austria) etc.]. Consequently none of three CBS releases we tested achieved state-of-the-art application security.

Selected security vulnerabilities found in the three tested systems might:

- Enable unauthorized access
- Disable segregation of duties
- Circumvent the effectiveness of auditing and logging
- Circumvent the effectiveness of strict access control and enable privilege escalation

These security vulnerabilities may cause the CBS to be in violation of compliance requirements such as OCC 2008-16, Basel II, SAS70, ISO/IEC 27001, national data privacy protection laws or other national banking specific laws and regulations.

Given the limited scope of the testing this should by no means be understood as an exhaustive list of the potential security vulnerabilities within the systems we tested, or those offered by other vendors included within this study. The respective vendors have been and/or will be informed by the participating banks about the security vulnerabilities identified by these security crash tests.

9 | REMEDIATION OF RISKS

Application security is vital to CBS products and vendors and CBS customers must address their risks with increased intensity. It is not yet clear how rapidly the threat level for cyber-crime and cyber terrorism will escalate in the coming years but there is no doubt at all that it will continue to rise.

Mitigation of risks for vendors

Any data breach or other security incident caused by security vulnerabilities in a CBS belonging to a banking customer will have a secondary impact upon the vendor. The trust in the CBS product will be eroded and financial and/or legal consequences are likely.
Figure 7 shows the overview of a generic lifecycle model developed by SEC Consult in 2010 to show the erosion of trust in vendors of insecure software products.

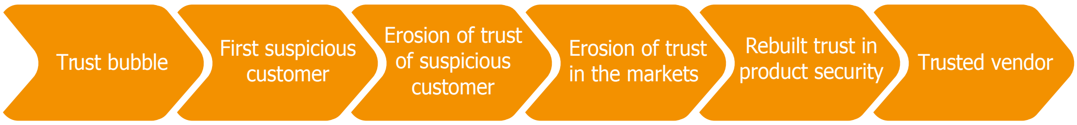

Trust bubble
First suspicious customer
Erosion of trust of suspicious customer
Erosion of trust in the markets
Rebuilt trust in product security
Trusted vendor

Figure 7: Lifecycle Model – Erosion of Trust in Software Vendors of Insecure Products (LM-ETSVIP)

- **Trust bubble:** In the first phase of the lifecycle customers and vendors of insecure software products hold trust in each other based on their individual assumptions. The customer assumes that the vendor does enough to achieve state-of-the-art application security while the software vendor assumes that a lack of complaints from customers indicates that their application security is sufficient.

- **First suspicious customer:** A first 'suspicious customer' testing the application security for toxic software is often the first means by which security deficiencies are revealed. From a customer's perspective the implicit promise on the part of the vendor regarding application security is not mirrored by the actual product they have been sold. For a software vendor such suspicious customers offer a tremendous opportunity to improve product quality and to initiate a comprehensive application security improvement program.

- **Erosion of trust of suspicious customer/in the markets:** An erosion of trust in the market is brought about when vendors ignore the larger issues indicated by the concerns of 'suspicious customers' and in an ad-hoc fashion repair only the individual vulnerabilities reported rather committing to an overhaul of the system as a whole. More and more customers will recognize the hidden security deficiencies and the ignorance of the vendor concerning the application security of toxic products.

- **Rebuilt trust in product security:** This phase starts after a commitment is made by the vendor to rebuild trust in product security and to discontinue the shipping of toxic software products. An increased effort over a significant time span is required by the vendor to improve not only the level of application security but to rebuild the trust of the vendors' customers.

  Recommended selected elements of a mitigation strategy for insecure, 'toxic' applications are:

  - Performance of application security tests with high assurance level for all product components.
  - Enforcement of Software Security Assurance in the whole software development process.
  - Ongoing reevaluation of the maturity of application security.
  - Maintenance of trust in the market through a long-term, fundamental approach to application security rather than a reactive and opportunistic one.

- **Trusted vendor:** This phase and status should be the objective of all vendors with regards to application security. Achieving and maintaining this status will require intense and sustained effort on the part of the vendor but will offer a definite advantage in the software marketplace.

Investments in application security are investments in the quality of the product and are therefore mutually beneficial to both vendors and customers of CBS.

Mitigation of risks for banks

Banks, rather than software vendors, will bear the brunt of any data breach or security incident caused by security vulnerabilities in a CBS. Therefore, it is critical that banks do not implement heavily insecure, 'toxic' applications, especially in the case of 'toxic' CBS.

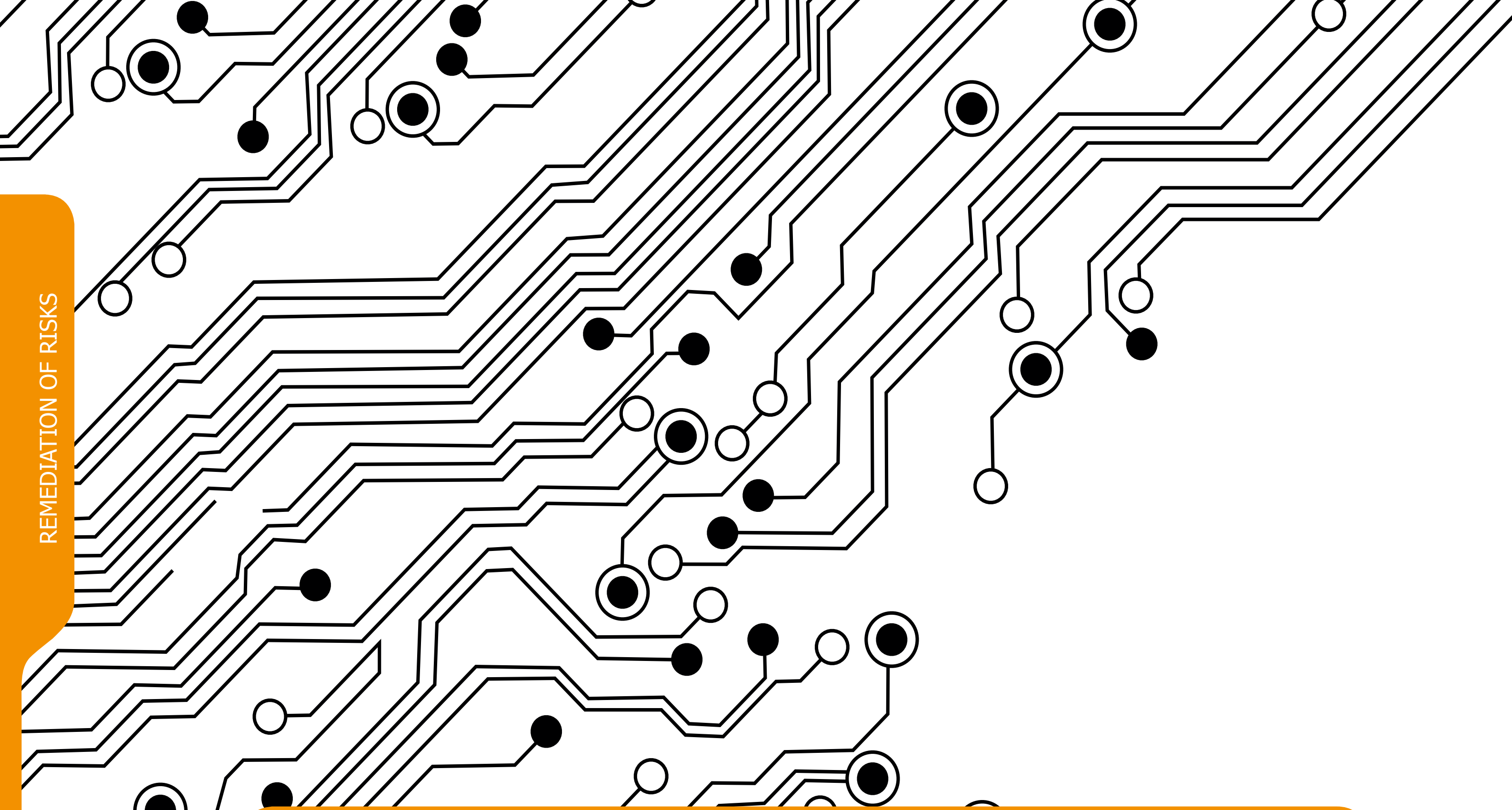
We strongly recommend the adoption of the following, selected elements of a mitigation strategy to circumvent risks from insecure, 'toxic' applications:

- **Demand state-of-the-art-application security for CBS**
  Enter contracts solely with vendors that fulfill mandatory state-of-the-art application security requirements

- **Penalties and cost sharing for security**
  Define penalties for not achieving state-of-the-art application security requirements and cost sharing for unsuccessful application security tests

- **Prove the vendor claims and promises by testing application security of CBS**
  Implement application security tests (Security Quality Gates) to attain transparency regarding the fulfillment of the state-of-the-art requirement

- **Establish additional, multiple lines of defense**
  Allocate budget and implement measures to, at least temporarily, mitigate some of the risks of an insecure CBS on other levels of defense (infrastructure, organizational, awareness of users etc.)



The only guaranteed way to ensure that applications are secure is to build security into the entire development process, including requirements, design and code and to develop IT systems to the highest security standards from the start.

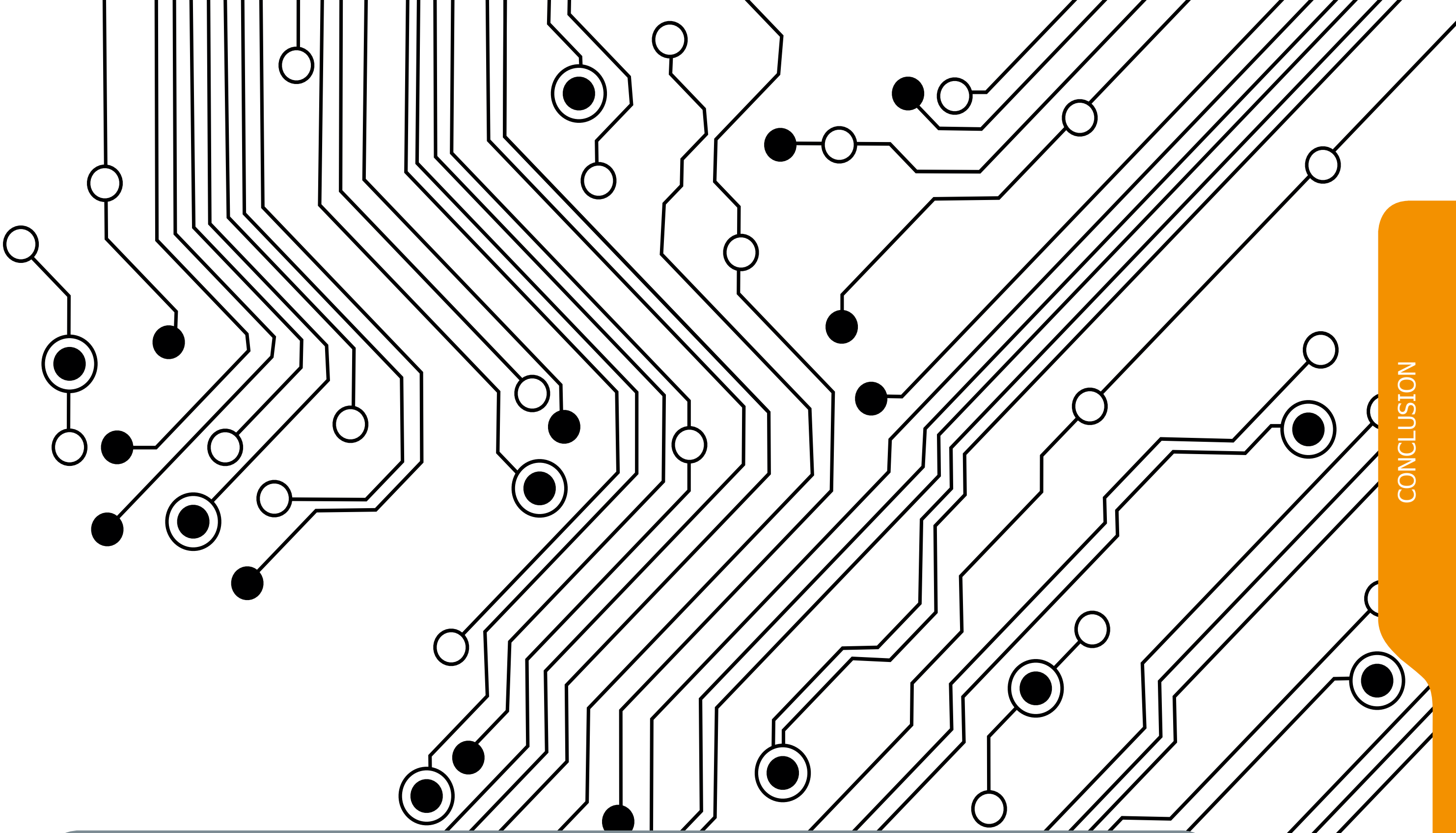
Capgemini, Sogeti, HP, World Quality Report 2011/2012

Recommendation No. 1: Demand software quality and security from suppliers. Many organizations today don't formally require their software suppliers to provide quality and secure software. This has resulted in an industry where time-to-market trumps all other considerations. If the buyer side starts to demand quality and security from software producers, this will incent them to invest more in application security measures. If you‘re a security professional in a buyer‘s organization, you need to encourage your sourcing colleagues to view application security maturity as an essential vendor selection criteria and as much as you can, demand software security and quality at the contractual level.

Forrester Research, Research Report 'Application Security: 2011 And Beyond', April 2011

Recommendation No. 2: Perform stringent acceptance tests for third-party code. Beyond contractual demands and careful vendor selection, another measure is acceptance tests for supplied code. **Remember, without actual tests, there‘s no way of validating your vendor‘s claims for quality and security.** Perform penetration tests on the supplied code to check for common security vulnerabilities, such as cross-site scripting, code injection, and buffer overflows. You can perform the tests internally or contract a trusted third party. There are also code analysis services available for more in-depth assessments on binary code. Use such a service if your supplier consents.

Forrester Research, Research Report 'Application Security: 2011 And Beyond', April 2011



10 | CONCLUSION

The objective of this study was to compare the promises and commitments of selected core banking vendors with the reality of their performance in hands-on security test. Additionally, this study aimed to become a building block for the continuous improvement of applications security of Core Banking System by outlining the necessary actions:

The study's major findings:

1. All of the vendors surveyed described their products as state-of-the-art in terms of application security.

2. All vendors surveyed in this study invest in the improvement of application security (e.g. threat modeling, training, software process improvements etc.) of their products.

3. Quality assurance approaches to security, especially security testing, differ significantly from vendor to vendor.

4. Three out of the eight products have undergone a security crash test conducted by application security experts. Severe security vulnerabilities have been found in all three tested products. All three tested products failed to demonstrate state-of-the-art application security due to poor applications security to the point of causing an operational risk to the banks concerned.

5. Banks must demand state-of-the-art application security from the vendors and validate it with application security tests. Failure to do so can result in the implementation of 'toxic' software products and incur operational risks.

6. Several of the surveyed vendors will have to significantly improve the application security of their product, starting with comprehensive application security tests before shipping to avoid shipping insecure 'toxic' applications.

SEC Consult

ADVISOR FOR YOUR INFORMATION SECURITY

Vienna – Frankfurt/Main – Montreal – Singapore – Vilnius

office@sec-consult.com

www.sec-consult.com