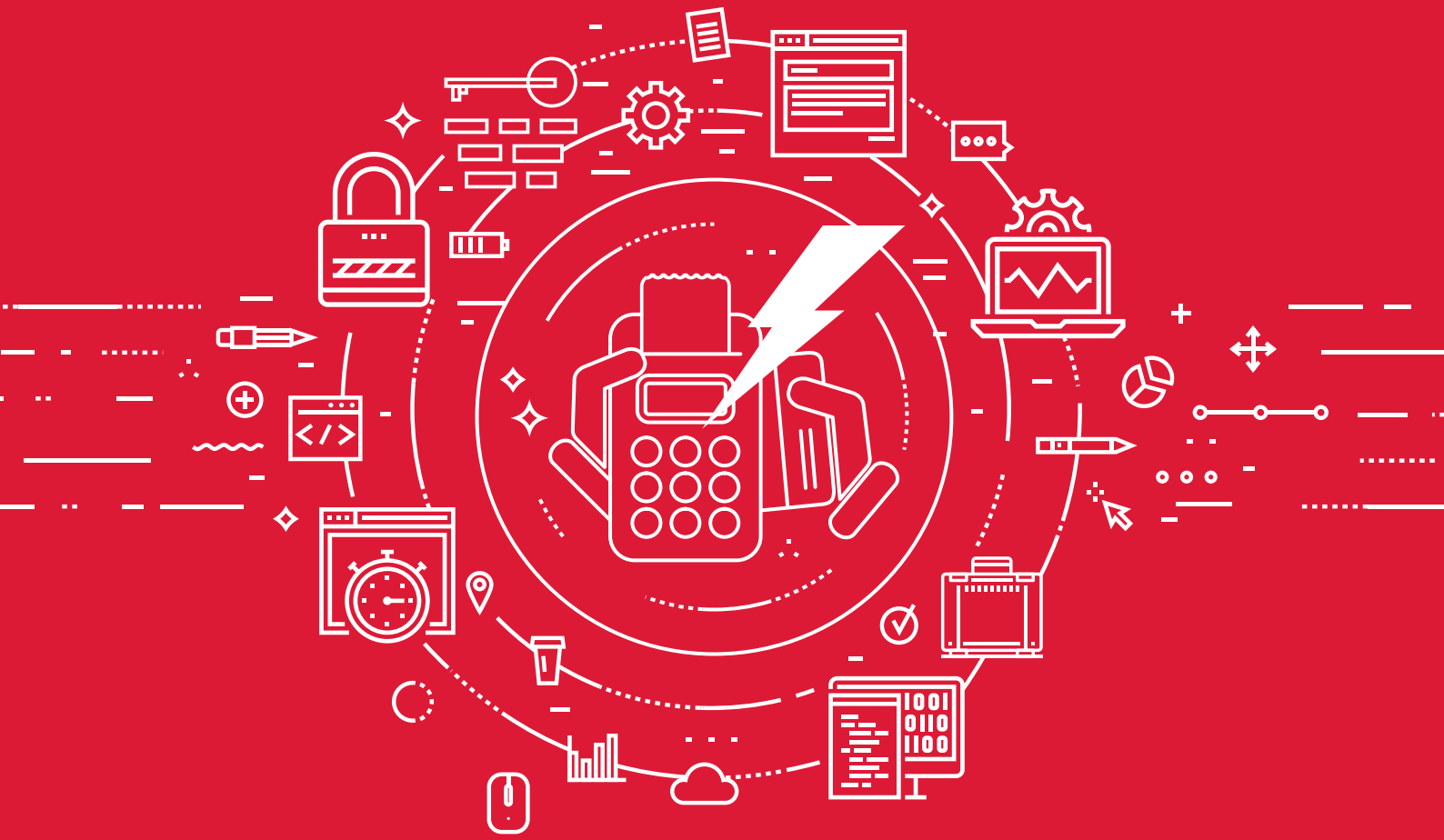
SECURITY TRENDS & VULNERABILITIES REVIEW

FINANCIAL SYSTEMS



2016

POSITIVE TECHNOLOGIES

Contents

Introduction

Today the banking industry seeks to implement the latest technologies to provide its clients with on-demand banking services from any location. To gain access to the majority of banking services, you just need to receive a card and sign an online banking service contract. Most financial institutions opt to offer remote services instead of traditional consumer banking services in a physical building.

This approach has many advantages. It is faster and more convenient for users, and it is less costly for banks as it does not require a large number of operation personnel or the creation of branch offices, and eliminates clients physically waiting in a queue that should lead to a higher customer satisfaction rate.

However, this connectivity and access lead to increased needs in terms of OLB security maintenance and risks mitigation. As OLB systems are public web and mobile applications, they are subject to the corresponding vulnerabilities and IS threats listed in numerous classifications, for example Web Application Security Consortium Threat Classification (WASC TC v. 2). Theft of funds is considered the most dangerous threat among attacks against OLB systems. Additional risks include unauthorized access to payment card data, personal data, and bank secrets, denial of service, and many other attacks that may cause significant financial and reputational loss.

This report provides statistics gathered during the OLB penetration testing performed by Positive Technologies in 2015. This research also identifies similarities and differences in the findings compared to those from 2014 and 2013. Thus, it is possible to track the dynamics of information systems development in the context of delivering information security.

1. Executive Summary

All OLB systems are vulnerable

Despite the decreasing number of critical vulnerabilities in OLB systems, they are still present in almost every application. Only 10% of the OLB systems do not contain such flaws (though they still suffer from medium-severity vulnerabilities).

A potential attacker could take control over a DBMS and steal confidential data from 55% of the systems examined, transfer funds with application user privileges in 25% or without — in 5%. In addition, an attacker may disrupt service functionality by gaining control over the server OS and exploiting other vulnerabilities (40%).

Off-the-shelf systems cannot guarantee security

In comparison to the previous years, the number of high severity vulnerabilities in applications supplied by vendors has fallen by half. However, all OLB products are critically flawed (e.g., SQL Injection, XML External Entities (XXE)). Additionally, OLB systems supplied by dedicated developers contain on average 1.5 – 2 times more source code bugs than the systems developed by on-site programmers.

Production OLB systems are also vulnerable

Every OLB system at the development stage has a significantly higher number of vulnerabilities than production applications do. The majority of threats are of medium or low severity. However, 40% of all flawed OLB systems in operation have critical errors. The situation is worse than in the test applications.

Issues with security tools are still present

The predictable format of identifiers is characteristic of all OLB systems, and users can change it in only 60% of the cases.

Two-factor authentication (2FA) for user logon and transaction manipulations reduce risks significantly in terms of stealing user funds, but 24% of the systems do not use this mechanism at all and 29% of the systems implement it incorrectly. 45% of the in-house systems are vulnerable and even commercial solutions have flaws (33%).

Additionally, one third of the systems does not provide sufficient session protection to prevent interception and further exploitation by possible attackers.

Mobile OLB systems for iOS are vulnerable

One third of security flaws in iOS applications are critical. These bugs are triggered by the storing and transferring of data in clear-text. Similar flaws are detected in Android OLB systems.

The iOS systems still have higher resilience to outsider attacks compared to the Android ones (75% of them are exposed to critical vulnerabilities); however, iOS still cannot guarantee security.

2. Research Data

The experts at Positive Technologies assessed 20 OLB systems in 2015. This research data set includes several financial service systems written in 1C with vulnerabilities similar to OLB systems. The study only includes those OLB systems that have undergone a complete analysis including an operation logic audit. It does not include the results of tests performed on systems that are still under development as these are only assessed for application code vulnerabilities, without any analysis of unauthorized transactions.

Most systems are designed for personal online banking (75%).

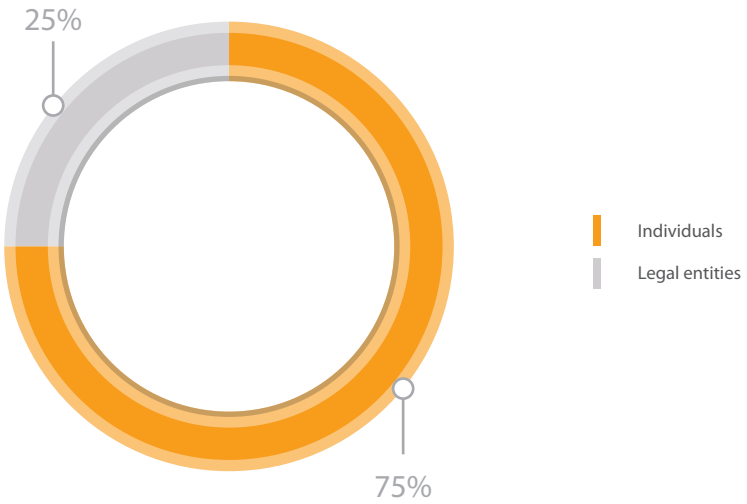


Figure 1. OLB distribution by service sectors

65% of the OLB systems studied are bank in-house applications. The other systems are based on platforms developed by well-known vendors. In order to comply with our responsible disclosure policy regarding vulnerabilities, no companies are named in this report.

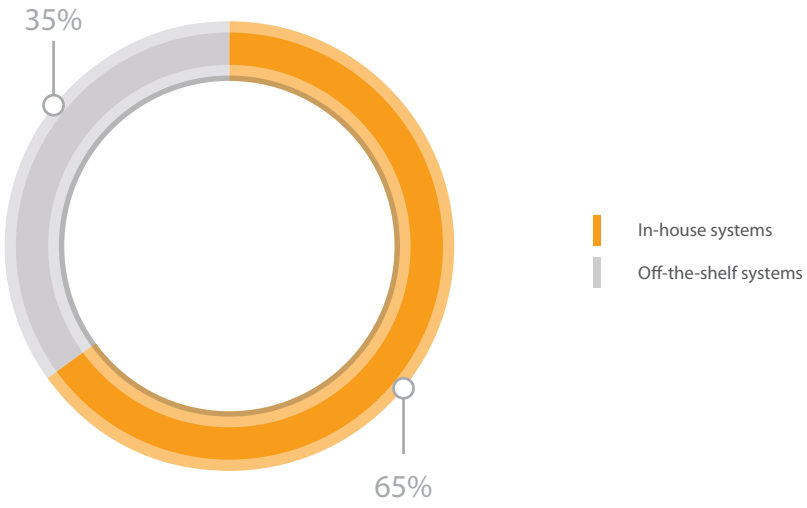


Figure 2. Systems by developers

In most cases, solutions developed by banks themselves are written in Java, and only 8% of the systems are written in 1C.

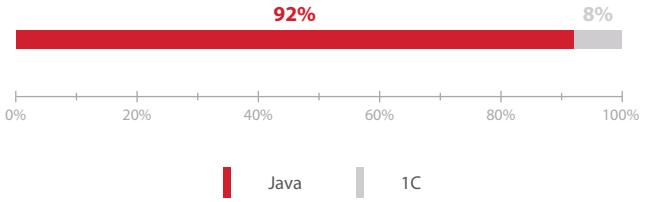


Figure 3. Systems by programming languages

The study includes mobile banking systems consisting of server and client components (35%).

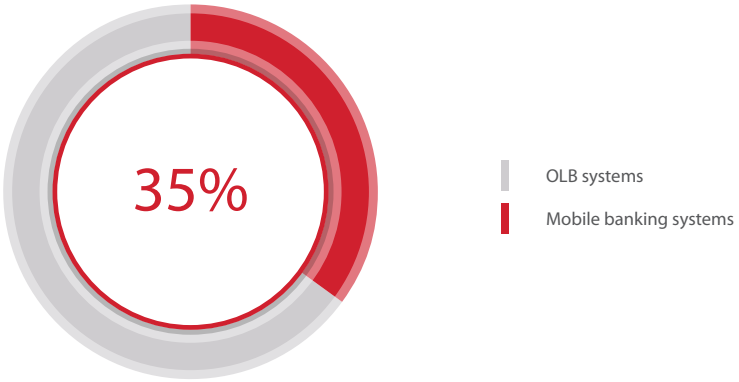


Figure 4. The percentage of mobile applications compared to all OLB systems tested

The OLBs studied are at different development stages. Most OLB systems (75%) are operational and accessible to clients. 25% of the systems are represented by testbeds ready for commissioning.

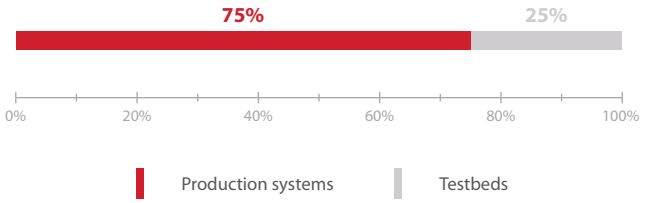


Figure 5. Systems by the stage of development

Most OLB systems developed by well-known vendors are operational (57%).

3. Results

3.1. Most Common Vulnerabilities and Related Threats

Our security analysis of the OLB systems in 2015 revealed security vulnerabilities in every system. In total, 171 vulnerabilities were found during testing. The majority of vulnerabilities discovered are classified as low severity (39%). Vulnerabilities classified as high (30%) and medium (31%) severities are found in almost equal proportions. The percentage of high-severity vulnerabilities has decreased by 14% as compared to 2013 and 2014. The rate shows the tendency towards rising OLB protection, but a more detailed analysis argues that the security level of OLB systems remains low.

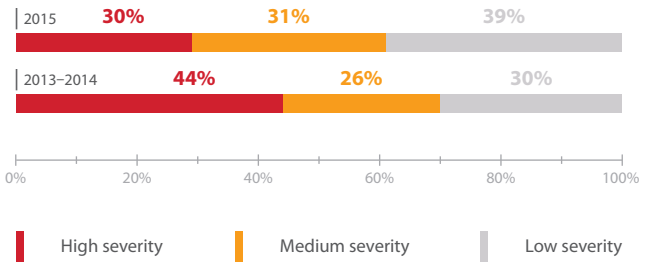


Figure 6. Vulnerabilities by severity

All OLB systems contain at least medium-severity vulnerabilities, and 90% of the systems has a critical vulnerability. These results are much worse than in 2013 and 2014.

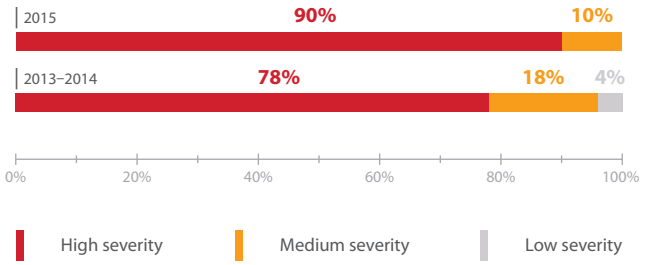


Figure 7. System distribution by maximum severity of the vulnerabilities detected

The majority of detected vulnerabilities (36%) are flaws in protection mechanisms, almost the same results were obtained in the previous years (42%). This includes flaws in identification, authentication, and authorization. 36% of the vulnerabilities detected are bugs in application source code such as XML External Entities or Cross-Site Scripting (XSS). The rest (27%) relate to OLB misconfiguration.

As opposed to the results of the previous two years, a vulnerability related to missing security updates (CVE-2015-1635) was detected in 2015. This vulnerability exists in the HTTP protocol stack of Windows OS, when it improperly parses specially crafted HTTP requests. The detailed description of the vulnerability is provided in Microsoft Security Advisory MS15-034 published on April 14, 2015. The vulnerability could allow arbitrary code execution or DoS attacks, if an attacker sends a specially crafted HTTP request to an affected system.

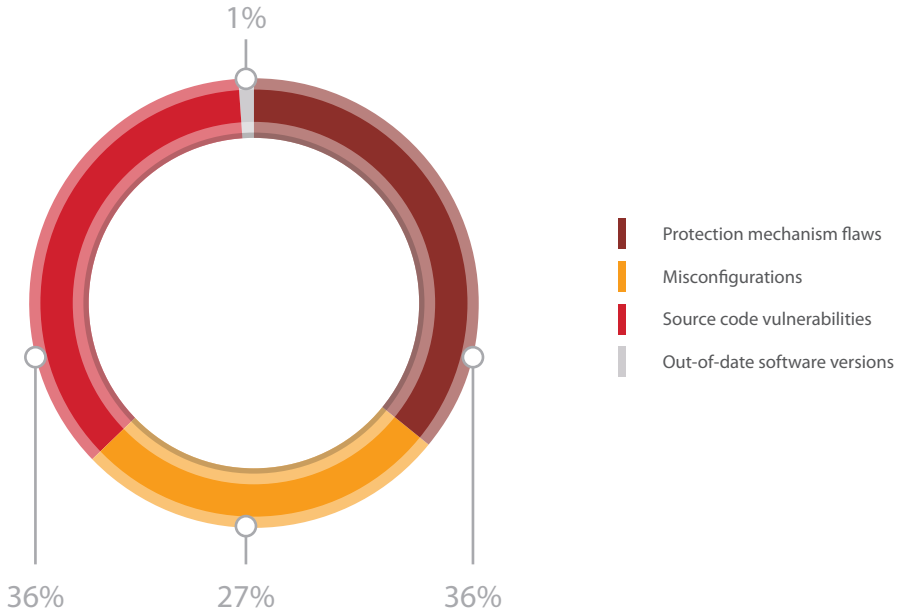


Figure 8. Vulnerabilities by categories

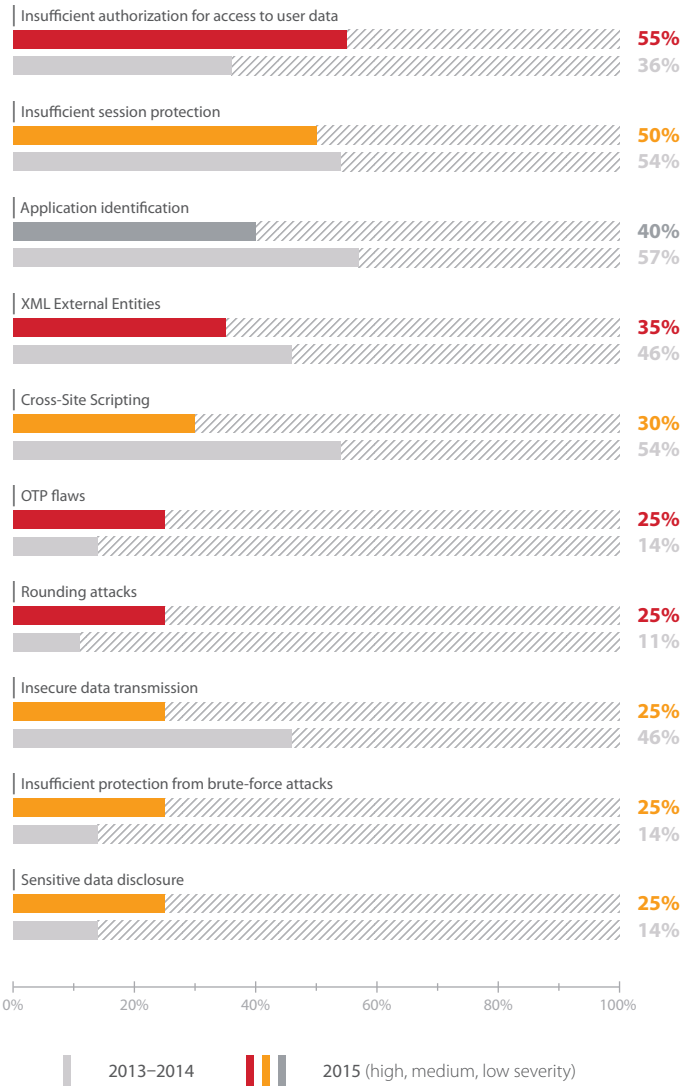


Figure 9. Most common OLB vulnerabilities (across systems)

It is worth noting that the small percentage of this vulnerability in the OLBs does not mean that the updates have been installed in a timely manner in all the systems investigated. Security analysis of most systems is conducted by black-box and gray-box testing, i.e. with privileges identical to those of a potential attacker. Thus, OLB systems could contain obsolete software versions.

Most of the systems tested allow attackers to obtain unauthorized access to user data, and this includes flaws in authorization. More than a half of the systems studied (55%) are exposed to critical vulnerabilities. It confirms that developers do not pay enough attention to the correct implementation of security mechanisms.

The second most common flaw is insufficient session protection. This medium-severity vulnerability includes improper session termination, cookies misconfiguration, multiple concurrent sessions run by the same user, and no association between user sessions and client IP addresses.

The third most common flaw is the low-severity vulnerability Application Identification (Fingerprinting). 40% of the systems contain this vulnerability, while previously 57% of the applications were vulnerable. It is worth mentioning that low-severity vulnerabilities topped the list of flaws in 2014.

XXE and Cross-Site Scripting are also in the top five most common vulnerabilities.

A combination of the detected vulnerabilities may cause critical security issues. The testing demonstrates the most serious threats to the security of the investigated systems.

5% of the OLB systems studied allows an external attacker to exploit a combination of various vulnerabilities (insufficient session protection, lack of two-factor authentication) to steal money. 25% of the OLBs are vulnerable to critical flaws such as stealing money by an authorized user as an intruder can conduct Rounding and SQL Injection attacks or gain an unauthorized access to other user operations. That may cause financial and reputation damage. Additionally, 10% of the systems allow access to the OS of an OLB system server. More than half of the systems (55%) allow unauthorized access to a DBMS with important personal and financial information.

Systems developed in-house and off-the-shelf systems are vulnerable. It is possible to exploit security threats found in all OLB systems regardless of their development stage: both testbeds and operational systems are vulnerable.

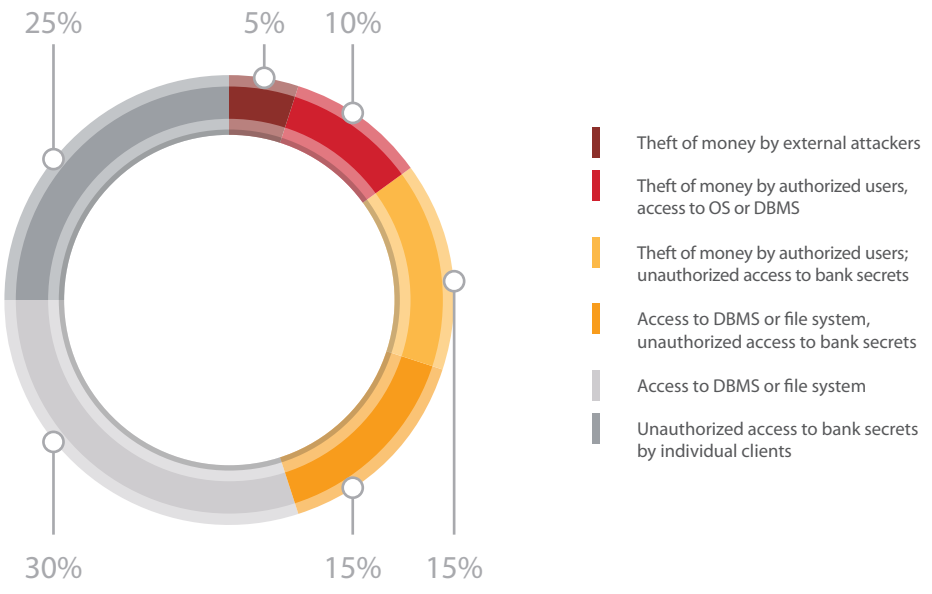


Figure 10. OLB security issues (across systems)

3.2. Vulnerabilities in Personal and Commercial OLBs

Both systems for personal and commercial banking contain similar flaws in security control mechanisms (one third of OLB systems). The number of commercial systems with such flaws has decreased from 47% to 31%.

There are significant changes in the frequency of other types of vulnerabilities. The personal systems contain twice as many misconfiguration vulnerabilities than commercial systems, though they have 2 times fewer source code vulnerabilities. Our security analysis has revealed obsolete software versions only in the commercial OLBs.

The percentage of the personal systems with misconfiguration flaws has grown from 22% to 38%.

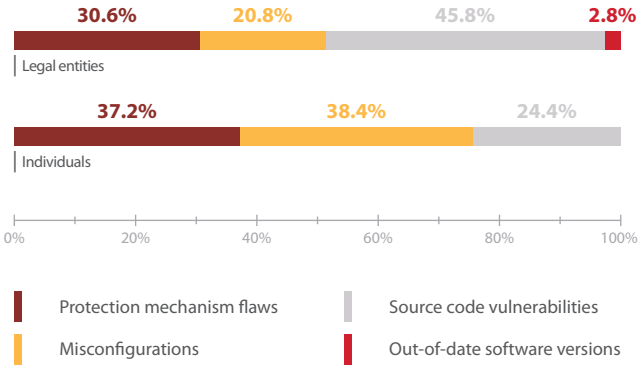


Figure 11. Systems by vulnerabilities of various categories (designed for individuals and legal entities)

The percentage of vulnerabilities of various severity levels in OLB systems designed for use by individuals and organizations does not vary significantly (see Figure 12). The number of critical vulnerabilities in both types of systems has decreased (by 19% for commercial and 12% for personal systems). The percentage of medium- and low-severity vulnerabilities has increased.

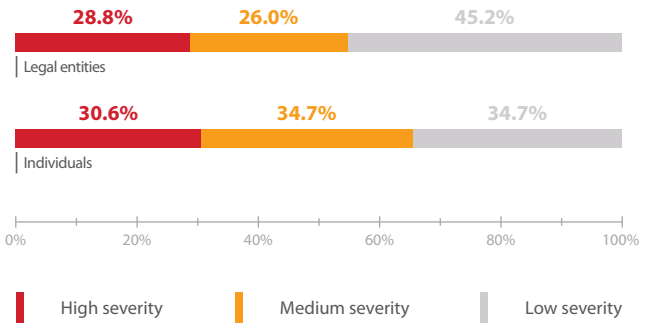


Figure 12. Vulnerability distribution by severity for systems aimed at individuals and legal entities

This tendency indicates improvements in OLB protection, though the general security level remains low. All resources studied are vulnerable, while all commercial OLBs contain critical flaws. The same results were obtained in the previous years. All personal OLB systems contain at least medium-severity vulnerabilities.

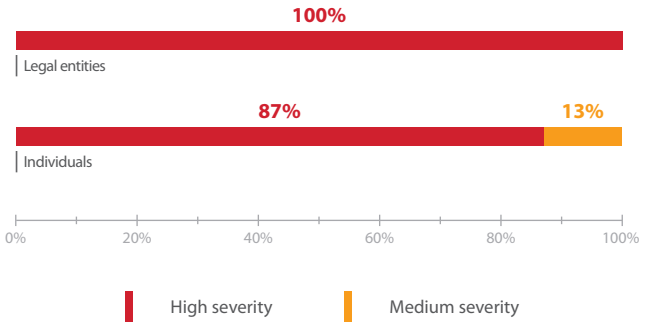


Figure 13. Systems by maximum severity of the vulnerabilities detected (for legal entities and individuals)

The percentage of high- and low-severity vulnerabilities detected in the commercial OLBs in 2015 is almost identical to the results obtained in 2013 and 2014, while the number of medium-severity vulnerabilities per system has increased.

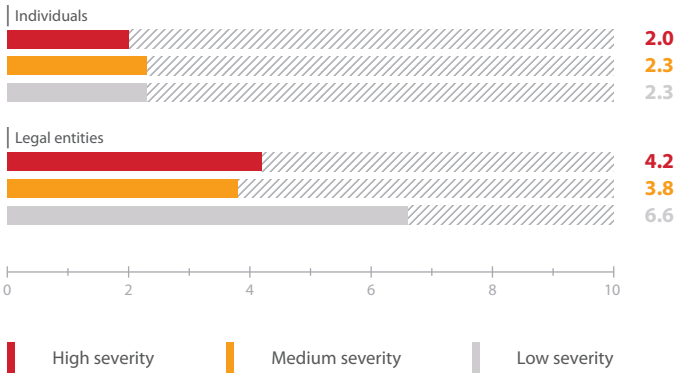


Figure 14. Average number of different severity vulnerabilities per system aimed at individuals and legal entities

The results show that the OLB systems aimed at businesses has decreased significantly, while the security level of the OLBs aimed at individual consumers remains low.

3.3. Vulnerabilities in In-House and Off-the-Shelf OLBs

The off-the-shelf systems contain more vulnerabilities related to source code errors than the in-house systems (40% versus 28%). The in-house systems have on average more misconfiguration vulnerabilities in comparison to the off-the-shelf systems (35% versus 27%). The percentage of vulnerabilities in security control mechanisms are almost equal for both types of the OLB systems.

The vendor's systems are exposed to vulnerabilities related to missing security updates (MS15-034). This vulnerability is excluded from the classification because it cannot be referred directly to the software provided by the vendor.

It is important to mention that the percentage of OLBs with critical vulnerabilities has almost doubled to 27% compared to the previous years (14%). There are no significant changes for other types of vulnerabilities. The percentage of in-house OLBs with source code vulnerabilities remains the same. The number of other vulnerabilities has also remained similar. The security level of both types of the OLB systems remains low.

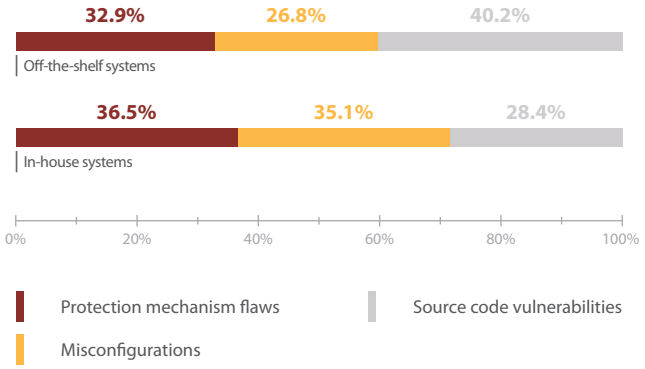


Figure 15. Vulnerabilities of various categories by off-the-shelf and in-house systems

It is worth noting that there is a decrease in the number of vulnerabilities per system. There is a tendency to detect more vulnerabilities in off-the-shelf OLBs. Banks pay more attention to the development process of their own products, while the vendors' systems are more vulnerable.

Every OLB system supplied by well-known vendors has 1.5 – 2 times more errors. The results in 2013 and 2014 showed the same findings. We think this is primarily because banks using third-party software rely on the vendor's QA. As OLBs are crossplatform and have complicated architectures and multiple features, they do not provide vendors with sufficient security at the source code level.

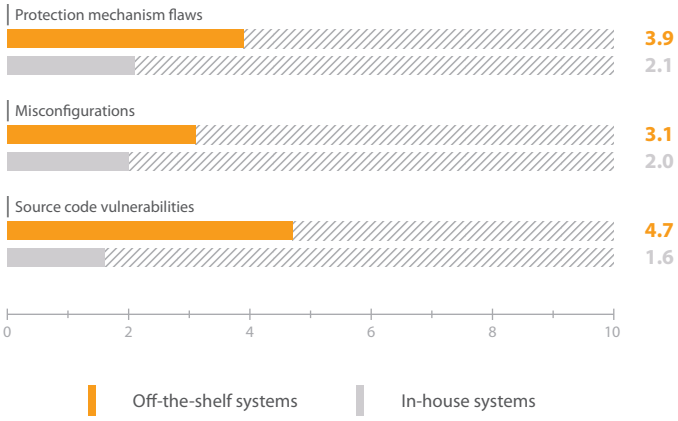


Figure 16. Average number of vulnerabilities per system

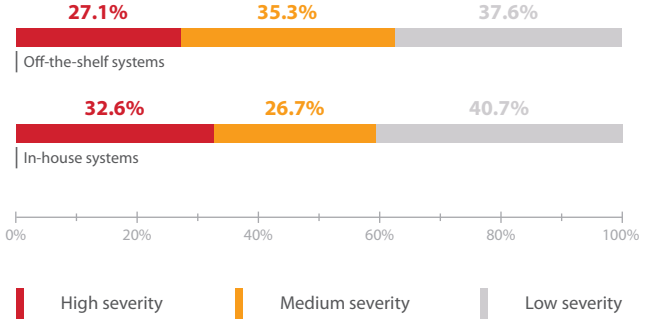


Figure 17. Vulnerability distribution by severity for off-the-shelf and in-house systems

27% of the vulnerabilities detected in the off-the-shelf systems are critical. One third of all vulnerabilities detected in the in-house OLBs are of high severity. The average number of vulnerabilities of all categories for both systems does not vary significantly. There is a tendency toward reducing the number of critical vulnerabilities. The amount of the vendor's OLBs with critical vulnerabilities almost halved in 2015.

Critical vulnerabilities are detected in all off-the-shelf OLB systems studied, though the ratio of such flaws to the total number has dropped. The results for the systems provided by vendors are worse than the 2014 figures, when 17% of systems did not contain critical vulnerabilities, and 6% of the resources studied contained only low-severity weaknesses. The figure for the in-house systems in 2015 has not changed significantly.

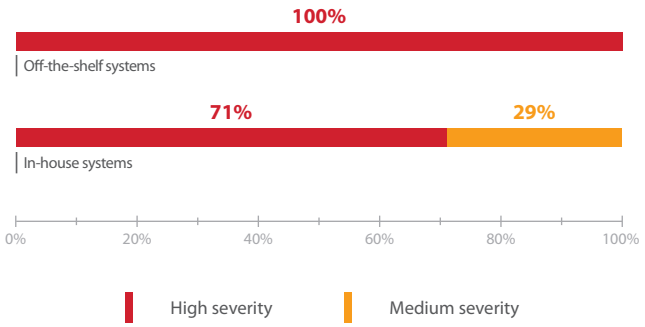


Figure 18. Systems by maximum severity of the vulnerabilities detected (for off-the-shelf and in-house systems)

An average vendors' OLB system has 3.3 critical vulnerabilities, while an average in-house OLB has 2.2 critical flaws. This is just one third of the errors found in 2014 and represents a significant improvement (see Figure 19).

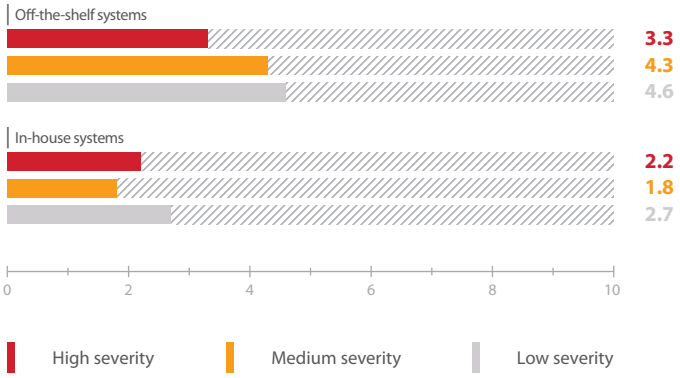


Figure 19. Average number of vulnerabilities per system (off-the-shelf and in-house systems)

Therefore, an average off-the-shelf OLB system has more vulnerabilities, because in-house systems are developed for a particular architecture and have a set of functionalities, which make them simpler and, thus, less vulnerable.

The results achieved do not imply that all existing third-party OLB systems are as vulnerable as the ones tested by us. They only show that a purchase of a vendor provided system does not automatically guarantee good security.

Vendors try to eliminate vulnerabilities detected during OLB security assessments and provide service packs. It is important that preventative protection means are implemented (Web Application Firewalls) in order to prevent vulnerability exploitation in OLB production systems until it is patched by the vendor. We recommend you to analyze the OLB security at all development stages and regularly (e.g., twice a year) in the course of operational use, and then control the process of vulnerability elimination.

3.4. Vulnerabilities in Test and Production OLB Systems

This section provides the results of the security analysis of test and production OLB systems. According to the analysis, the test systems contain more vulnerabilities related to flaws in security mechanisms and misconfiguration than to code errors.

In 2013-2014, 49% of the testbeds and 41% of the production systems contained vulnerabilities in security control mechanisms.

The test systems are exposed to vulnerabilities related to missing security updates (MS15-034).

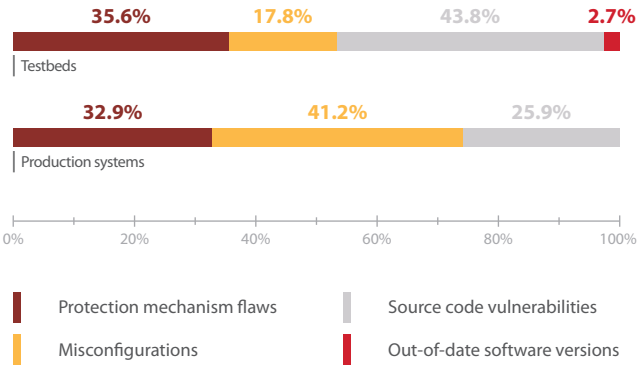


Figure 20. Vulnerabilities of various categories by test and production systems

On average, the testbeds contain more vulnerabilities of all categories than the production systems. This can be attributed to the fact that many security checks are carried out during pre-production testing. A security assessment is usually conducted to detect such vulnerabilities.

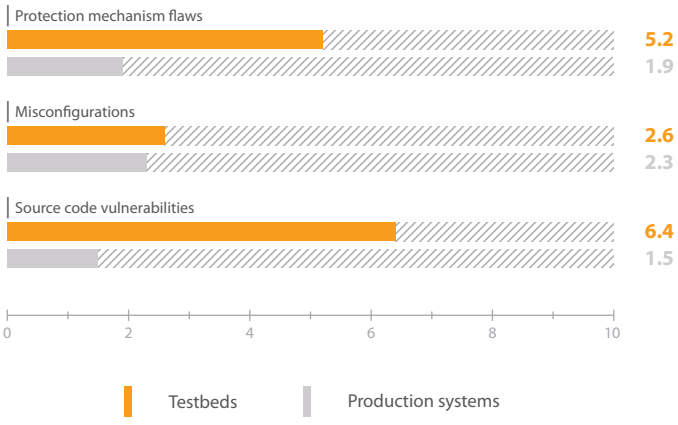


Figure 21. Average number of various vulnerabilities per system (test and production systems)

The percentage of protection mechanism flaws and misconfigurations in the testbeds is almost the same as in the previous years. The number of source code vulnerabilities for the test systems has doubled as compared to the results in 2013-2014, while the percentage of such flaws in the production OLB systems has decreased by a factor of five. This finding may be explained by the fact that banks focus more on vulnerability detection during OLB commissioning.

This does not demonstrate an improvement in the production OLB system security, because the severity level of the vulnerabilities detected remains high. The production systems contain more critical vulnerabilities than the testbeds (see Figure 22).

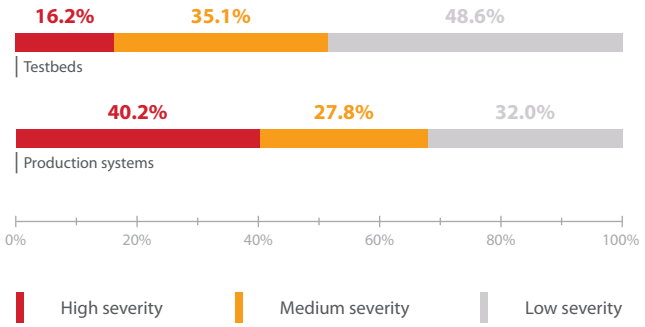


Figure 22. Distribution of various severity vulnerabilities in test and production systems

The percentage of vulnerabilities of all severity levels is almost the same for the production systems. The percentage of critical vulnerabilities detected in the test systems has halved. Due to a large amount of low- and medium-severity flaws, the average number of vulnerabilities per test system is huge.

All OLB systems contain at least medium-severity vulnerabilities. Almost all production systems are exposed to critical vulnerabilities (see Figure 23). The results obtained argues that the implementation of Secure Systems Development Life Cycle and regular security assessments of operational systems is critical.

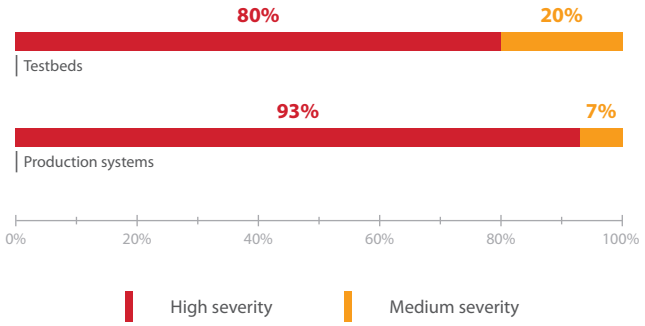


Figure 23. Systems by maximum severity of the vulnerabilities detected (for testbeds and production systems)

The figure shows that the test systems have more low- and medium-severity vulnerabilities than critical flaws. The average number of high-severity vulnerabilities per system is almost the same for both system types.

The 2015 results demonstrate that the production systems have fewer vulnerabilities than the testbeds. These findings indicate that banks dedicate more effort to the operational systems security. The security level of the production systems is not high, because all production systems contain critical vulnerabilities.

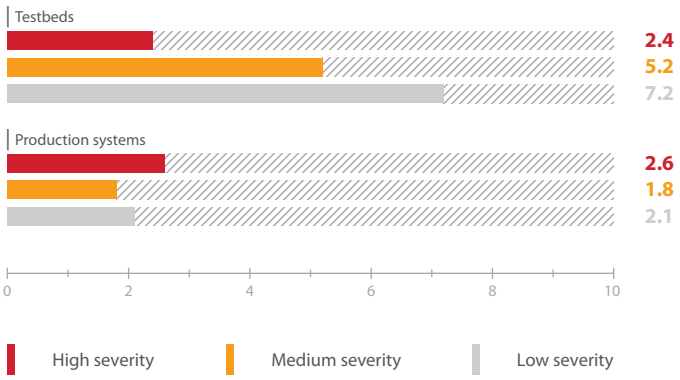


Figure 24. Average number of various vulnerabilities per system (test and production systems)

4. Review of the Most Critical Vulnerabilities

This section covers high-severity vulnerabilities that are detected in the majority of the OLB systems examined.

The OLB systems are still vulnerable to XXE and insufficient authorization. The percentage of the systems with source code errors vulnerable to SQL Injection decreased in 2015, while more OLBs became vulnerable to arbitrary code execution and rounding attacks. An intruder may exploit these vulnerabilities not only to obtain full control over the server (as a result of code execution), but also to attack application logic and steal money.

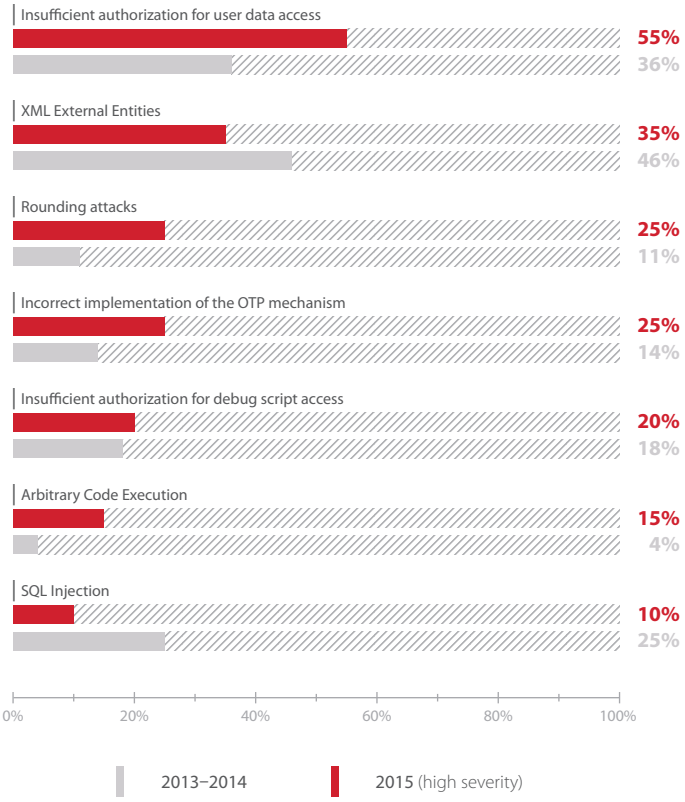


Figure 25. The most common critical vulnerabilities found in OLBs

More vulnerabilities related to the incorrect implementation of the OTP mechanism are detected. When there is no need for a one-time password to change template details, an intruder can use a session and steal money from other users. Besides, an attacker can conduct several transactions using one OTP (Race Condition attacks).

4.1. Insufficient Authorization for Accessing User Data

This vulnerability relate to flaws in the user authorization mechanism. 55% of the OLB systems have incorrect or no user privilege check, allowing an attacker to obtain a direct access to some scripts and files. Depending on the system's features and functions, an attacker can obtain unauthorized access to the following information:

+ Personal data of users
+ Information about user transactions and payment cards
+ Files uploaded into OLB
+ Certificates data
+ Specific information about clients (accounts, payment templates, etc.)

For example, one of the personal OLB systems does not conduct correct authorization assessments, and an attacker can obtain access to various system procedures and functions, including OLB settings and online statement of account, which do not need two-factor authentication.

4.2. XML External Entities

This security weakness allows attackers to obtain the content of server files or execute requests to the local network of the attacked server. The application does not check data received from a user properly, so an attacker can change request logic via DTD Schema injection into an XML document. It allows an attacker to obtain sensitive information, source code, and configuration files and conduct a DoS attack.

35% of the systems studied in 2015 are vulnerable to XXE. In 2013-2014, XXE was the most common critical vulnerability (46%) in all OLB systems.

4.3. Rounding Attacks

One quarter of the systems suffers from application logic attacks. This vulnerability allows an attacker to receive a sum of money larger than what is provided by the current exchange rate.

Attack example:

1. An attacker converts 0.29 RUB (Russian Ruble) to USD (United States Dollar). If the price of 1 USD is 80 RUB, then 0.29 RUB equals to 0.003625 USD. This sum is rounded up to the hundredths place, i.e. to 0.01 USD (one cent).
2. Then attacker converts 0.01 USD back to rubles and gets 0.8 RUB. The attacker's bonus is 0.51 RUB.

Thus, an attacker can automate this procedure and steal more money. For example, if there is no limitation on the number of transactions per day and on the minimum sum, then a malicious user exploiting these bugs in combination with a Race Condition attack, may obtain about 100 RUB from each account daily. Thus, a hacker with necessary technical tools and access to several accounts can steal a huge amount of money.

This type of attacks is well-known, but not all OLB systems have effective protection against it. To prevent this type of attacks and minimize the risks of embezzlement, it is important to restrict minimum sum for exchange within one transaction (for example, 5-10 RUB and 2-3 USD or other currency).

5. Identification Flaws

The most common identification flaws in the OLB systems are predictable UID format and UID disclosure.

These flaws allow an attacker to predict algorithms of UID generation and make a list of registered user identifiers. Using this information, an attacker can obtain unauthorized access to OLB systems (as a result of a brute-force attack). If a system contains an insufficient authorization vulnerability, then an intruder can gain unauthorized access to user personal or financial information. Despite the low severity of these vulnerabilities, an attacker, who gains identifier data together with the flaws of authentication, authorization and transaction security (see sections 6 and 7), can then obtain an unauthorized access to user personal accounts and conduct transactions.

5.1. Predictable UID Format

All OLB systems studied contain a predictable format of account identifiers.

In order to obtain access to a personal account provided by his or her bank, a client uses a login and password generated in accordance with a set algorithm. An attacker can figure out the algorithm. In 10% of the systems studied, UIDs coincide with user's mobile number. One of the OLBs provides IDs consisting of 6 digits, and each following identifier differs from the previous in one.

In order to reduce the risk of compromising systems caused by brute-force attacks, there must be the ability to change the password and identifier. It is important to warn a user to change the logon data provided by the bank for initial authorization, because not all users are experts in information security. For this purpose, there should be a special message on the authorization page of OLB systems forcing the user to change default authorization credentials or logon data after specified periods of time (90 days). It is important to pay special attention to the password policy that should forbid the use of simple combinations in passwords.

Only 60% of the OLBs provide an opportunity to change the identifier provided by a bank.

5.2. UID Information Disclosure

If an identifier does not exist in an OLB system, the server displays a corresponding error message. The vulnerable system sends a message confirming that the identifier has not been registered. Some OLB systems do not disclose registration information directly in messages, but a system may send back different server answers depending on the existence of the ID entered. An attacker analyzes server responses and determines registered identifiers. Information disclosure is common for new user registration scripts or password change scripts.

Every third OLB system was exposed to this vulnerability in 2013-2014. Only one OLB system of all studied was vulnerable in 2015. We suppose this is primarily because this flaw is easy to detect, while the elimination of other medium- and high-severity vulnerabilities often requires source code alteration.

6. Analysis of Authentication Mechanisms

Most systems investigated (60%) use mandatory two-factor authentication that demands a token or a one-time password. The remaining OLB systems implement authentication based on a user ID and password.

The detected OLB authentication flaws can be divided into the following categories:

+ Lack of mandatory two-factor authentication when accessing personal accounts and/or conducting transactions
+ Insufficient authentication
+ Weak password policy
+ Inadequate protection from brute-force attacks

Vulnerabilities triggered by missing or incorrect 2FA (required for access to a personal account or executing transaction) were the most common system vulnerabilities in 2015. Weak password policies were the most common vulnerabilities in 2013–2014.

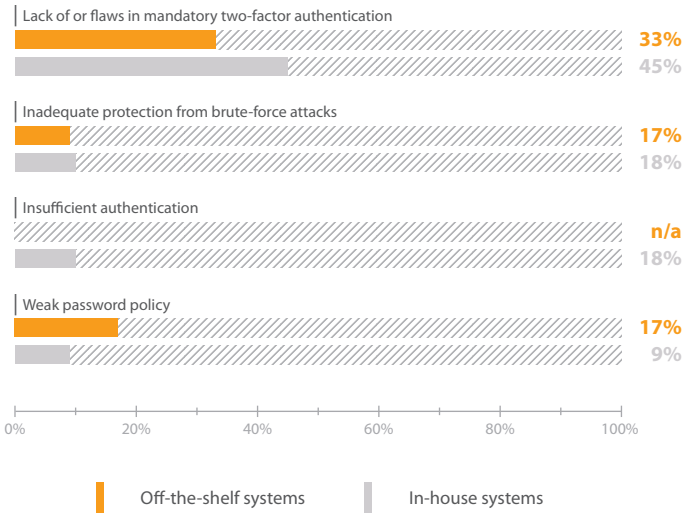


Figure 26. Authentication flaws found in different OLB systems

6.1. Two-Factor Authentication Flaws

45% of the in-house OLB systems as well as 33% of the vendor's OLBs contain certain two-factor authentication flaws. One third of systems lacks mandatory two-factor authentication required to access personal accounts or conduct transactions (see Figure 27).

An attacker can obtain access to user accounts (for example, as a result of credentials bruteforcing or session interception) and freely conduct transactions if there is no 2FA.

6.1.1. One-Time Password

Incorrect implementation of the OTP mechanism was discovered in 27% of the in-house OLBs studied. This weakness was detected in 17% of the vendor's OLBs. According to the research, the most common weaknesses in the OTP mechanism are as follows:

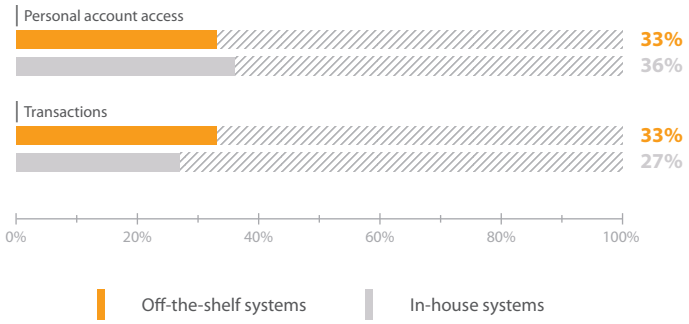


Figure 27. Different OLB systems distributed by 2FA vulnerabilities

+ OTP brute-force attacks
+ No confirmation of transactions or other operations by the OTP (e.g., an attacker can change operation templates without OTP confirmation)
+ Ineffective protection from unauthorized change of SIM cards (an attacker may receive an OTP)

6.1.2. Other Mechanisms

One OLB system has a vulnerable two-factor authentication mechanism that uses e-signatures with keys. The mechanism does not ensure sufficient integrity control, so due to the vulnerability an attacker could inject arbitrary data in a signed certificate, i.e. change certificate data or forge the root certificate.

6.2. Insufficient Protection from Brute-Force Attacks

18% of the in-house OLB systems as well as 17% of the off-the-shelf OLBs are exposed to brute-force attacks. The following weaknesses were found:

+ Lack of CAPTCHA or CAPTCHA bypass vulnerabilities
+ No temporary account blocking after several failed login attempts

Though temporary or permanent account lockout after several failed authentication attempts is often used as a protection against brute-force attacks, this mechanism doesn't provide full protection. If a password policy allows the use of weak passwords and there is no 2FA, an attacker can bruteforce UIDs by using one single password.

Incorrect locking may result in denial of service to legitimate bank users, if an attacker has received identifier information. It can trigger substantial reputational loss.

When there is no two-factor authentication at logon, developers should use the CAPTCHA technology to protect their systems from brute-force attacks. If incorrect credentials are entered several times, this mechanism requires a user to input a series of characters that are displayed on screen to obstruct a malicious user form conducting automated attacks and avoid account lockout.

Using the CAPTCHA technology together with a strict password policy will strengthen the system's resistance to brute-force attacks if there is no two-factor authentication. If blocking is used, banks should take into account such factors as the time interval between consecutive login attempts, resource IP address, and incidents of both password and identifier bruteforcing.

6.3. Insufficient Authentication

Insufficient authentication occurs when a web application permits an attacker to access sensitive data or system functionality without proper authentication. For example, in one of the systems exploitation of this bug allows a malicious user to obtain access to the script vulnerable to XXE. The application handles an XML document at first and then checks credentials. This combination of two various vulnerabilities allows an attacker to read the content of server files.

It is necessary to set strong access restrictions for application features and restrict administrative access to OLB system scripts and features by checking user access privileges.

6.4. Password Policy Flaws

Certain password policy vulnerabilities are detected in 12% of all the OLB systems, including weaknesses connected with lack of or incorrectly implemented restrictions on:

+ password length
+ password complexity
+ password age

These flaws together with insufficient protection from brute-force attacks and lack of two-factor authentication allow a hacker to conduct attacks aimed at obtaining unauthorized accesses to the personal accounts of bank clients.

Positive Technologies experts recommend using passwords that are at least 8 characters long and include upper and lowercase letters, figures, and special characters. Common dictionary passwords (e.g., P@ssw0rd) and combinations of keys that are set close to each other on a keyboard should be forbidden. Password age also should be restricted (e.g. no more than 90 days) and a user should not be able to reuse one of three previous passwords.

7. Flaws in Authorization Mechanisms and Transaction Security

In the OLB systems studied, users are authorized via a session generating mechanism. Some systems require not only session IDs, but also additional characteristics such as unique request tokens for the authorization. In all the systems, a session ID has a sufficient entropy to obstruct brute-force attacks. However, the sessions of several systems are not strongly protected from hijacking and further exploitation.

The majority of the systems studied use OTP authentication to protect transactions.

The most common critical transaction protection flaws are:

+ Insufficient authorization, including when accessing user data (see section 4.1) and debug scripts (70% of the OLBs).
+ Incorrect implementation of the OTP mechanism, which allows an attacker to conduct several transactions using one OTP or intercept the OTP value (24% of the OLB systems).
+ Lack of two-factor authentication for transactions (29% of OLB systems).

The majority of authorization weaknesses are found in the in-house OLBs. The most critical vulnerabilities are caused by improper verification of access rights when conducting critical operations. This includes the ability to change a password or disable the OTP mechanism by sending a special request to an application without providing the password.

However, 35% of the systems studied are not strongly secured from hijacking (e.g. via XSS) and further exploitation. According to the research, the most common weaknesses in the OTP mechanism are:

+ A session identifier is not linked to an IP address.
+ Cookie parameters containing session IDs and other important data are not secured properly.

8. Source Code Vulnerabilities

8.1. Overall Statistics

This section provides an overview of the vulnerabilities related to OLB source code errors. The percentage of the systems with vulnerabilities in application source code decreased in 2015 from 82% to 68%. Additionally, all source code errors are of medium severity or higher. More than a half of the systems (63%) have high-severity vulnerabilities (see Figure 28); in 2013-2014, 68% of the systems were vulnerable.

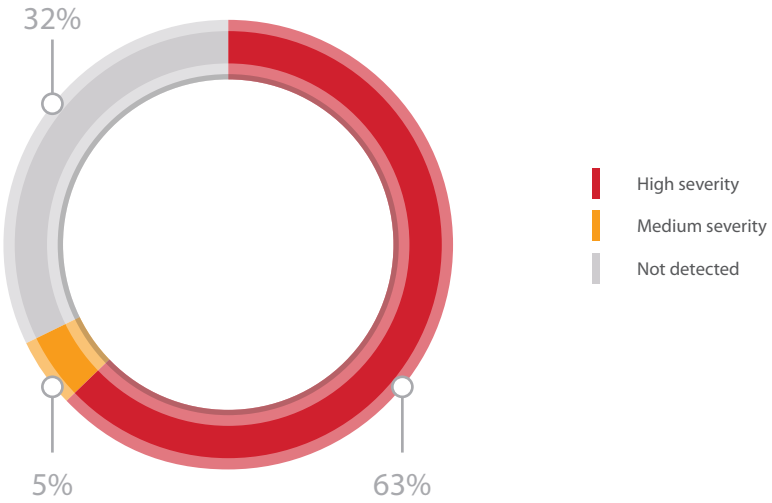


Figure 28. Systems with source code flaws by maximum severity

The percentage of vulnerable systems has decreased, but more than half of the systems are exposed to high-severity flaws.

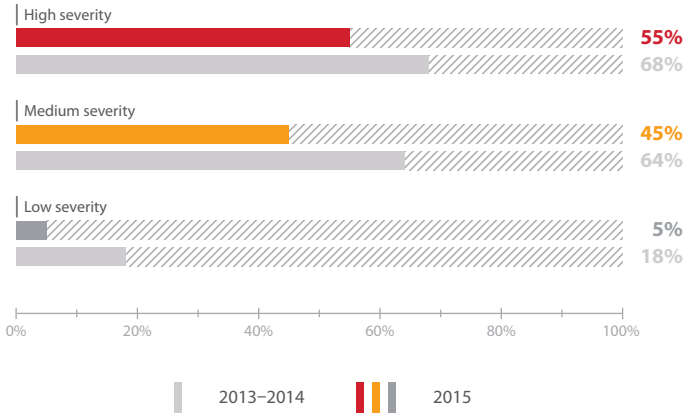


Figure 29. The percentage of OLBs with various severity vulnerabilities

Severity of source code errors detected in 2015 has not changed significantly in comparison to the severity of bugs found in 2013 and 2014. 42% of source code vulnerabilities are critical. This category includes the most common critical OLB vulnerability — XXE (see section 4.2). 55% of the detected vulnerabilities are classified as medium severity.

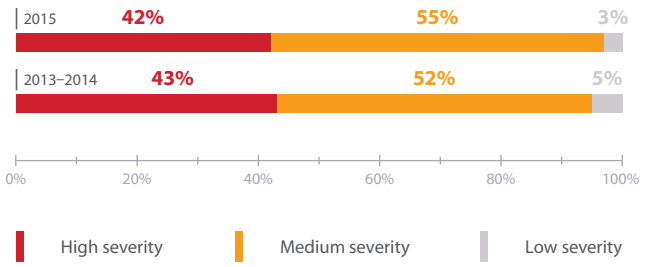


Figure 30. Source code vulnerabilities by severity

Figure 31 shows the average number of source code vulnerabilities of various severity levels per system. The results in 2015 are twice as better as the same value in the previous years. For example, an average system contained 3.2 critical vulnerabilities in 2013-2014.

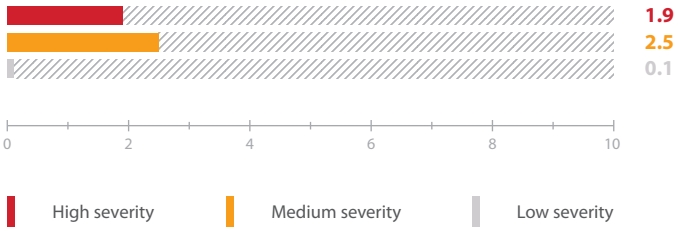


Figure 31. Average number of source code vulnerabilities per system

8.2. Application Vulnerabilities in In-House and Off-the-Shelf Systems

The off-the-shelf systems contain more source code vulnerabilities (see section 3.3) on average than in-house ones. 55% of the vulnerabilities detected in the vendors' systems are classified as high severity (see Figure 32), very similar results to those of the previous years.

All in-house OLB systems contain at least medium-severity vulnerabilities, while 27% of the systems are exposed to critical flaws. This figure was almost the same (25%) in 2013-2014.

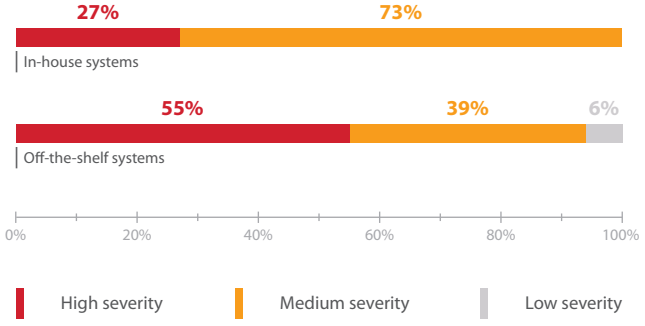


Figure 32. Severity of source code errors by developers

An average off-the-shelf OLB system contains 2.6 critical source code vulnerabilities. The in-house systems had fewer critical vulnerabilities in 2015 (see Figure 33), and the results in 2013 and 2014 showed the same findings.

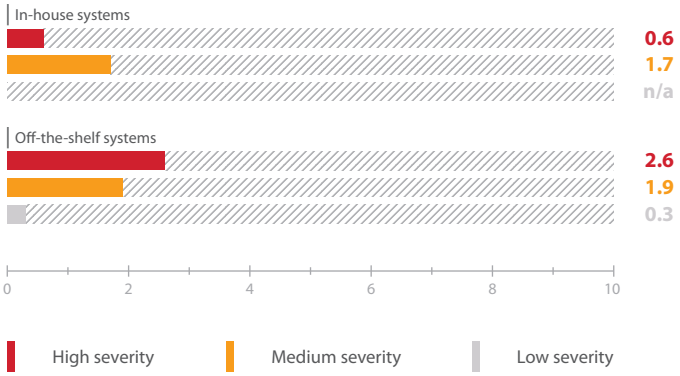


Figure 33. Average number of source code vulnerabilities per system (by developers)

The in-house OLB systems are developed for a particular architecture and have a predefined set of functionalities, which makes them simpler and less vulnerable (see section 3.3).

8.3. The Most Common Vulnerabilities

The most widespread code vulnerabilities are XXE, Cross-Site Scripting, and Arbitrary Code Execution. The rating of the most common vulnerabilities has four critical flaws.

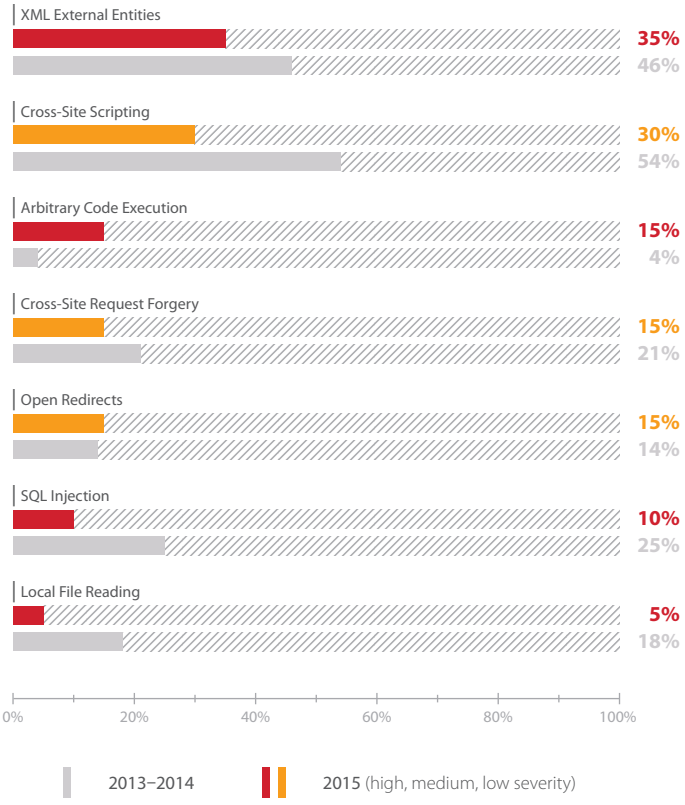


Figure 34. Most common source code vulnerabilities

The percentage of systems vulnerable to arbitrary code execution increased in 2015. This security flaw allows attackers to execute OS commands on an application server and escalate privileges. Attackers may obtain access to sensitive data on servers and conduct various attacks including DoS attacks.

Arbitrary code execution, URL Redirector Abuse, and Local File Reading were not detected in the off-the-shelf systems. The vendors' systems have such critical vulnerabilities as XXE and SQL Injection.

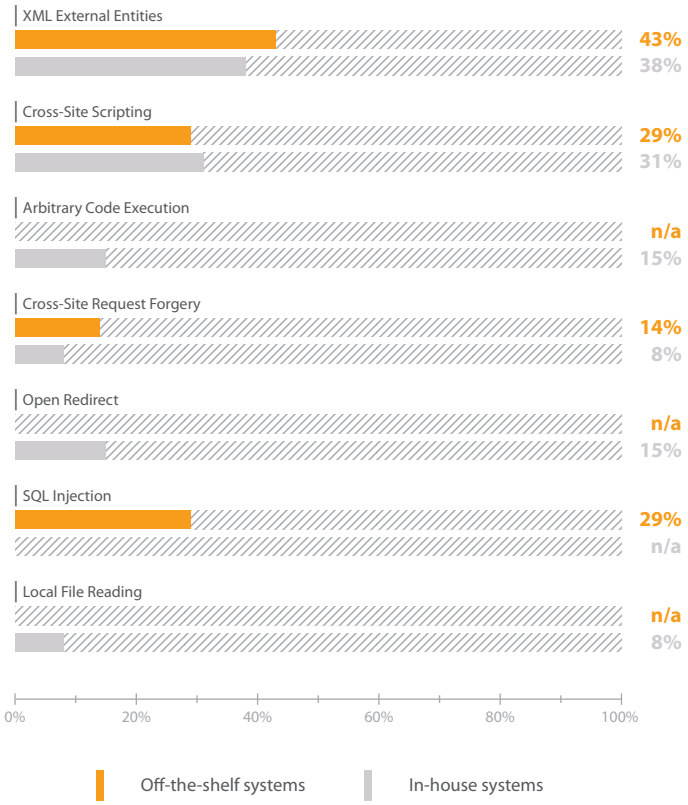


Figure 35. Systems by application code vulnerabilities (for in-house and off-the-shelf systems)

To reduce the risks of exploitation of source code vulnerabilities, it is important for developers to implement secure development procedures, analyze security of applications regularly (ideally by source code analysis), and eliminate all vulnerabilities detected immediately. We recommend you to implement preventative protection means (Web Application Firewalls) to prevent exploitation of vulnerabilities in off-the-shelf OLB systems until they are patched by the vendor or in in-house OLBs when it is impossible to eliminate flaws immediately.

9. Configuration Flaws

9.1. Overall Statistics

This category of flaws is caused by the incorrect configuration of operating systems, DBMS, web servers, and web application components. 27% of all vulnerabilities detected in this study are of this type (see Figure 8). Most vulnerabilities of this category (57%) are classified as medium severity. In 2015, none of critical vulnerabilities, such as application excessive privileges, were detected.

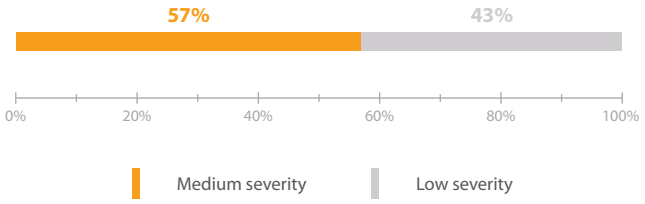


Figure 36. Configuration flaws by severity

90% of the systems studied contain at least one configuration flaw. Most OLB systems (65%) contain medium-severity vulnerabilities.

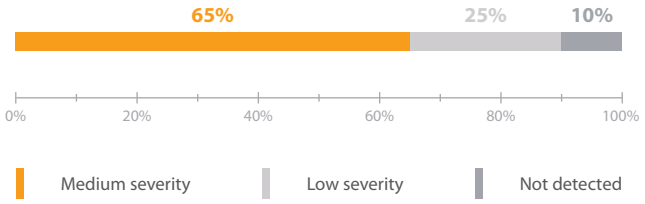


Figure 37. Systems with configuration flaws by maximum severity

On average, every OLB system contains at least one medium and one low-severity vulnerability caused by misconfiguration.

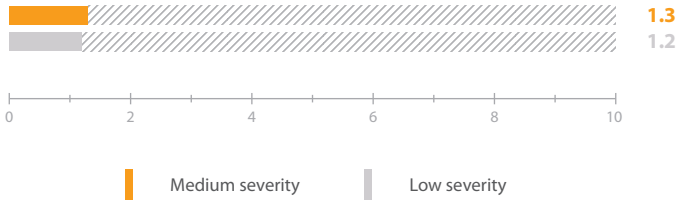


Figure 38. Average number of configuration vulnerabilities per system

The most common vulnerabilities of this type are:

+ Session attacks (50%)
+ Fingerprinting (40%)
+ Important data disclosure (25%)
+ Insecure data transfer (25%)

For example, if a cookie parameter that transmits a session ID is not configured to "secure" then a browser can transfer it via both HTTPS and HTTP. If HTTP or weak encryption algorithms are used, an attacker can hijack user data.

9.2. Configuration Flaws in Off-The-Shelf and In-House Systems

The majority of the configuration vulnerabilities (57%) found in the in-house systems are medium-severity. The ratio of low- and medium-severity vulnerabilities is the same in the off-the-shelf systems.

Figure 40 identifies the average number of configuration vulnerabilities per system. Both types of the systems have a small amount of configuration flaws. Although, the off-the-shelf systems have more vulnerabilities of this type that the in-house OLBs.

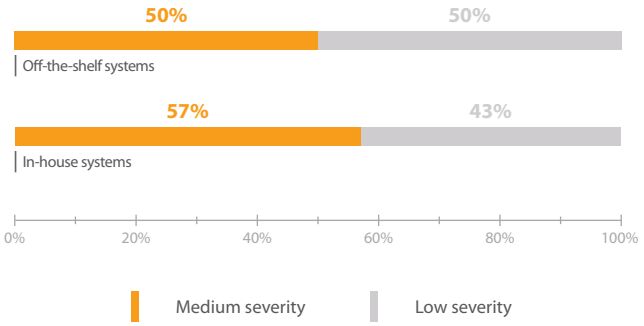


Figure 39. Severity of configuration vulnerabilities (by developers)

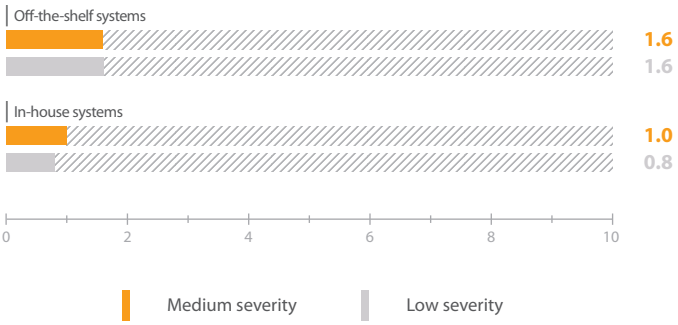


Figure 40. Average number of configuration vulnerabilities per system

10. Denial of Service

DoS attacks are one of the most urgent security threats. In terms of OLB systems, this attack can be aimed at disrupting the availability of both the system server and the user's personal account with obviously financial and reputational implications for the bank concerned.

DoS attacks can be the result of insufficient data processing or exploitation of vulnerabilities such as XXE, XPath Injection, arbitrary code execution, and other. 40% of the OLBs are vulnerable to DoS attacks.

If an ID format is predictable, an attacker can lockout a user account by entering invalid data into the password field multiple times (if there is an automated account lockout). Only one system studied in 2015 is exposed to this type of attack.

11. Mobile OLB Vulnerabilities

Android applications are more vulnerable than iOS applications as in the previous years. High-severity vulnerabilities were detected in 75% of the Android applications and in 33% of the iOS applications (see Figure 41).

It is worth noting that though storing and transferring data in clear text is a medium-severity vulnerability, in particular cases it was reclassified as critical as this was recommended by the owners of the relevant systems. Such high-severity flaws as SQL Injection, XXE, and arbitrary code execution were not discovered in 2015 at all.

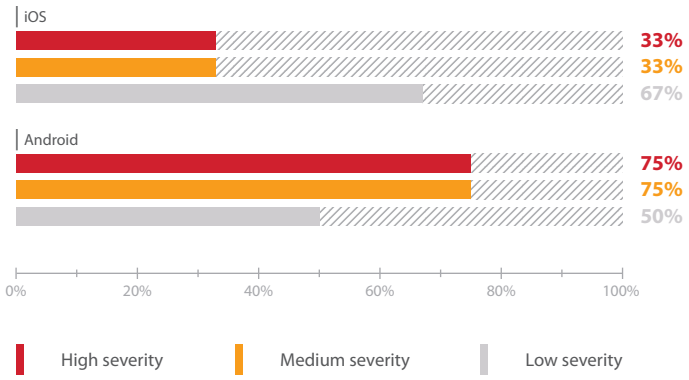


Figure 41. Mobile OLB systems by vulnerabilities of various severity

All systems investigated in 2015 are vulnerable; the Android OLBs contain at least medium-severity vulnerabilities. 33% of the mobile applications for iOS contain only low-severity vulnerabilities.

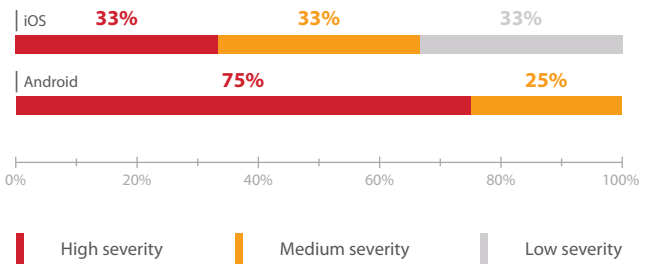


Figure 42. Mobile system distribution by maximum severity of the vulnerabilities detected

An average Android application studied in 2015 contains at least 3.8 vulnerabilities; this figure is almost equal to the results of 2013-2014 (3.7). An average iOS application has 1.6 vulnerabilities. There is an improvement because an average iOS application had 2.3 flaws previously. Both iOS and Android applications are exposed to Fingerprinting.

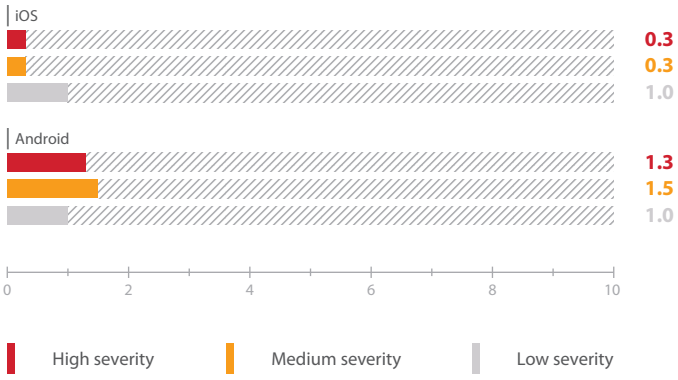


Figure 43. Average number of different severity vulnerabilities in mobile OLB per system

The most common mobile OLB vulnerabilities are related to insecure data storage and transfer as well as to the lack of session protection, though the percentage of systems with such flaws has reduced.

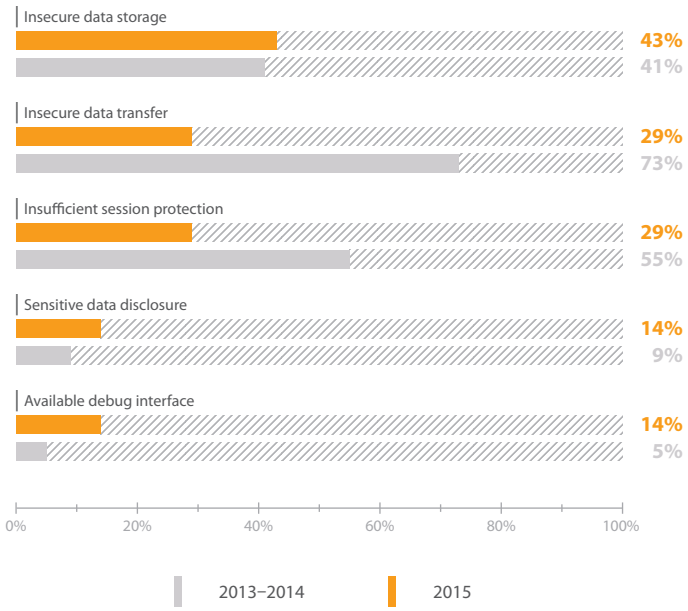


Figure 44. Most common mobile banking vulnerabilities

The majority of the vulnerabilities discovered are classified as medium severity, but a combination of them could have a critical impact on a system. Owners of OLB systems consider these vulnerabilities as critical. For example, an attacker can change a web server response by exploiting incorrect implementation of the short PIN-code mechanism in combination with a session identifier stored in file systems. Therefore, the server response will be always true even if a PIN code is incorrect. Thus, an attacker can obtain full control over a user account, alter configurations, and perform transactions. In one of the applications, an attacker can obtain access to user banking due to the lack of data transfer protection. The system allows the use of self-signed certificates for data transfer via HTTPS.

11.1. Insecure Data Storage

Most of the mobile applications studied store sensitive data in clear text, 43% of the systems are vulnerable. Sensitive information includes clients' personal data, users' identifiers and passwords, session IDs, debug information and other, which can be used for attacks.

OLBs should use encryption or forbid saving sensitive data in public directories. It is better to set a strong access policy, adhering to the principle of providing the minimum necessary privileges.

11.2. Insecure Data Transfer

Mobile application vulnerabilities related to insecure data transfer fall into the following categories:

+ Data transfer via insecure HTTP
+ Lack of SSL certificate verification
+ Adding self-signed certificates to the trust list on the user's device
+ Transferring sensitive data (e.g., the user password) in clear text without hashing

By exploiting these vulnerabilities, an attacker can gain access to sensitive data transferred by the application, e.g., IDs, user passwords, geolocation data, and other critical information. For instance, if an OLB user connects to a public Wi-Fi, an attacker can conduct a man-in-the-middle (MiTM) attack and intercept data transferred in clear text. An attacker can also automate the attack, interfere with the transaction process and replace the payment details in real-time mode.

OLBs are recommended to use secure data transfer protocols (e.g., TLS) and certificates signed by a certification authority.

11.3. Insufficient Session Protection

The OLB systems investigated have a number of vulnerabilities that allow a hacker to attack user sessions:

- When a user logs out, the application does not send a request to terminate the session. A hacker can intercept a user session and gain access to user data and payment cards.
- When a user logs out, the application sends a server request to terminate the session, but the session stays active for 20 minutes. A hacker can exploit this bug to conduct an attack or extend session activity by sending requests to the server.
- A session ID is not bound to an IP address. An attacker can intercept the session ID and gain unauthorized access to the user account.
- The "secure" property is not configured for the cookie parameter, which allows a hacker to intercept a session ID via insecure HTTP.

When logging out, it is important to terminate an active session to protect it. A session identifier should not be active when the user has logged out. Simultaneous connection of various devices should be restricted and when another device tries to log in, an active session should be terminated. The cookie parameters containing session IDs and other sensitive data should be secured properly.

Summary

This research demonstrates that the security level of OLB systems remains low, though the percentage of critical vulnerabilities has decreased. Both systems, personal and commercial, in-house and off-the-shelf are vulnerable.

The security bugs found in the systems already put into production indicate the importance of secure software development lifecycle processes. OLB security audit should be performed not only at each stage of application development and prior to commissioning, but also during the course of its operational use. It is important to analyze security regularly (e.g., twice a year) and control vulnerability elimination. Off-the-shelf OLB systems are often more vulnerable than in-house systems and require more control. Banks should use preventive protection means like Web Application Firewalls. In particular, when using vendor provided systems, a WAF is required until the third-party vendor releases an update, which prevents attackers from exploiting known vulnerabilities.

According to the research, identification, authentication, and authorization flaws are still among the most common weaknesses. These flaws may cause critical security issues, unauthorized transactions, and acquisition of control over an OLB system. To access a user account, a hacker needs to use well-known flaws like insufficient session protection. OLBs must ensure the correct implementation of security mechanisms. To reduce any risks related to application vulnerabilities, vendors should implement secure development procedures and provide comprehensive testing at the acceptance stage.

We also recommend performing a regular quality assessment of application source code via the white-box testing (including automated tools). Finally, performing a regular and comprehensive OLB security assessment will reduce the risk of security violation and help to avoid financial and reputational losses.

About Positive Technologies

Positive Technologies is a leading global provider of enterprise security solutions for vulnerability and compliance management, incident and threat analysis, and application protection. Commitment to clients and research has earned Positive Technologies a reputation as one of the foremost authorities on Industrial Control System, Banking, Telecom, Web Application, and ERP security, supported by recognition from the analyst community. Learn more about Positive Technologies at ptsecurity.com.

POSITIVE TECHNOLOGIES

pt@ptsecurity.com ptsecurity.com