Internet Banking System
Web Application Penetration Test Report

Kiev - 2014

1. Executive Summary

This report represents the results of the Bank (hereinafter – the Client) Internet Banking Web Application (hereinafter – the Application) penetration test conducted by Berezha Security between 28th of December and 16th of November 2014.

As a result of the penetration test, we have concluded that the overall security of the Application is in an *acceptable* state, while there are certain improvements that we recommend to implement in order to apply generally acceptable good practice for security of internet-facing online financial systems.

2. Immediate Remedial Actions

No immediate remedial actions are required to be performed over the Application's security.

3. Scope and Goals

The goal of the penetration test was to assess the security posture of online internet banking web application located by the URL

https://www.thebank.com/payme/

and hosted on the system with external IP address

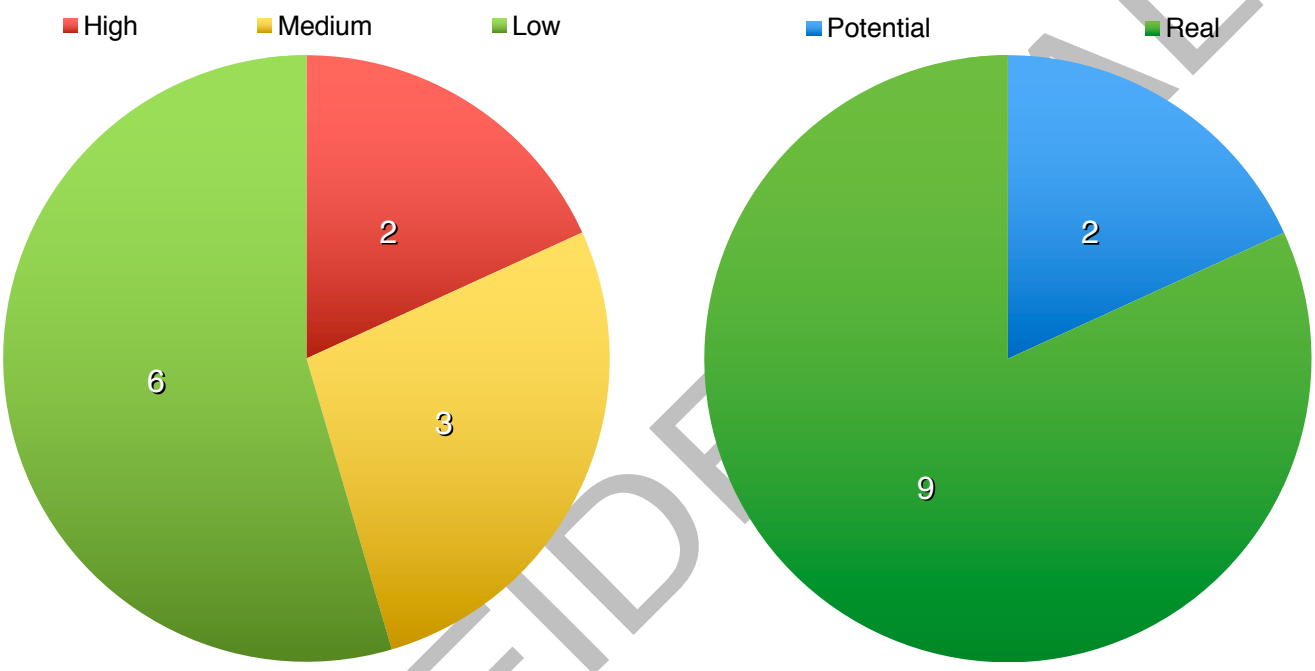
X.X.X.X

The Application was tested in two ways: without user access to the Application and with a valid user account provided by the Client. Since no significant access control vulnerabilities have been found during the penetration test, this report does not differentiate the findings by the manner of their discovery (e.g. authenticated or unauthenticated).

During the penetration test we have experienced multiple occasions of unpredictable temporary increases of response time of the Application while the latency of the hosting system remained within acceptable limits. We assume this behavior to be caused by the fact that the Application still remains in its development phase and may have been changed during the penetration test. We recommend the Client to investigate this issue and insure that no such behavior is demonstrated by the Application in production.

4. Overview of the Results

During the penetration test we have discovered 11 significant vulnerabilities that we recommend remediating, including 2 vulnerabilities of High risk, 3 vulnerabilities of Medium risk, and 6 vulnerabilities of Low risk. 2 of the discovered vulnerabilities have Potential status, and the status of 9 vulnerabilities is Real.



We recommend remediating the vulnerabilities of High risk before moving Application to production. In case required changes and extensions are not acceptable from the time and budget point of view, the Client's responsible management should accept related risks and plan remediation activities after production release.

5. Technical Results

This section contains detailed descriptions of the vulnerabilities we have discovered during the penetration test along with the remediation steps we recommend to take in order to eliminate them.

The issues are outlined in order of descending Risk value. Please refer to Appendix A for details on the risk assessment methodology used in this report.

The Status value of each vulnerability is either Potential or Real. Potential vulnerabilities do not pose direct risk to the Application but might be used by a malicious person in order to perform certain attacks. Though, there is no sufficient evidence that Potential attacks may be effectively used to attack the Application.

In contrast, Real vulnerabilities were verified and are backed by sufficient evidence. However, this status does not signal that an attacker may easily and readily use the vulnerability: the probability of such event is indicated by the Risk value.

Please see Appendix B for the evidence of each vulnerability and related explanation.

Table 1. Vulnerabilities and Recommendations

Index	VULN-01
Title	One-Factor User Authentication
Description	User authentication mechanism of the Application does not involve a second factor of authentication.
Risk	High
Status	Potential
Remediation	Consider adding a second authentication factor to the access control subsystem of the Application.
Comments	For a web-application that allows bank account manipulations (e.g. transfers of funds) it is essential to strengthen the user authentication process by a second factor such as hardware or software token, or a one-time password delivered out-of-band.

Index	VULN-02
Title	No Lock-Out After Failed Authentication Attempts
Description	Although we observed that multiple failed authentication attempts using the same user name result in the secure lock-out of a corresponding user account, it is still possible to mount password guessing attacks using different user names.
Risk	High
Status	Potential
Remediation	Consider adding a mechanism of secure source blocking after multiple failed authentication attempts using different user names.
Comments	Secure lock-out is used by the Application to protect user accounts against targeted password guessing attacks. This approach is efficient assuming that Application user names cannot be easily guessed. However, during the penetration test we have been provided with a user account with a numerical user name. Based on that, we assumed that the user name convention of the Application is the sequence of 4 digits. Also the user names might as well be sequential 4-digit numbers. In case if these assumptions hold true, it is easy for an attacker to mount “username-guessing” attacks rotating through predictable user name space and trying a few most popular passwords against each user account. We have performed an imitation of the attack explained above. Although the imitation had no positive outcome within a short time frame of a few hours, it has demonstrated that this attack vector is available and can be used by any internet user.

Index	VULN-03
Title	SSL Certificate Is Not Trusted
Description	The server's certificate is not trusted, which means any trusted SSL Certification Authority did not sign it. Instead, SSL certificate is signed by a test CA run by the VeriSign company for trial use.
Risk	Medium
Status	Real

Remediation	Although in general SSL certificate issues may constitute a significant risk of information disclosure or disruption, we assume that the web server is going to be configured with a valid SSL certificate before moving to production.
Comments	SSL helps to protect the confidentiality and integrity of information in transit between the browser and server, and to provide authentication of the server's identity. To serve this purpose, the server must present an SSL certificate, which is valid for the server's hostname, is issued by a trusted authority and is valid for the current date. If any one of these requirements is not met, SSL connections to the server will not provide the full protection for which SSL is designed.

Index	VULN-04
Title	The POODLE Attack (SSLv3 Is Supported)
Description	The SSL server (port: 443) encrypts traffic using a vulnerable version of SSLv3.0. An attacker may be able to exploit this problem to conduct man-in-the-middle attacks and decrypt communications between the affected service and clients.
Risk	Medium
Status	Real
Remediation	It's recommended to disable SSLv3 and replace it with TLSv1.0 as soon as compatibility with legacy clients is no longer required. (The only browser that does not support TLSv1.0 is Internet Explorer 6). To disable SSLv2 and SSLv3: For Apache: SSLProtocol all -SSLv2 -SSLv3
Comments	Websites that support SSLv3 and CBC-mode ciphers are potentially vulnerable to an active MITM (Man-in-the-middle) attack. This attack, called POODLE, is similar to the BEAST attack and also allows a network attacker to extract the plaintext of targeted parts of an SSL connection, usually cookie data. Unlike the BEAST attack, it doesn't require such extensive control of the format of the plaintext and thus is more practical. Any website that supports SSLv3 is vulnerable to POODLE, even if it also supports more recent versions of TLS. SSLv3 will be disabled by default in Firefox 34, which will be released on Nov 25 2014.

Index	VULN-05
Title	Session Token In URL
Description	9 instances of this issue were identified, at the following locations: *[URLs redacted]*
Risk	Low
Status	Real
Remediation	The application should use an alternative mechanism for transmitting session tokens, such as HTTP cookies or hidden fields in forms that are submitted using the POST method. As it is shown in the evidence section, the application uses tokens in URLs that point to non-sensitive data that can be accessed without authentication based on session tokens. It is recommended to store non-sensitive web-site information (e.g. JavaScript modules, static image files) in the areas that do not require user authentication.
Comments	Sensitive information within URLs may be logged in various locations, including the user's browser, the web server, and any forward or reverse proxy servers between the two endpoints. URLs may also be displayed on-screen, bookmarked or emailed around by users. They may be disclosed to third parties via the Referrer header when any off-site links are followed. Placing session tokens into the URL increases the risk that an attacker will capture them.

[The rest of vulnerability descriptions were redacted.]

Appendix A. Risk Evaluation Methodology

Issue Risk is defined as a product of Issue Impact and Issue Exploitability:

Risk = Impact * Exploitability

Further risk evaluation is performed according to the following taxonomy and selection principles.

Table 2. Issue Impact

Value	Description
Critical	The issue can pose a very high security threat such as allow an attacker to gain full administrative access to the system, allow all traffic to pass through the security control device unfiltered etc.
High	The issue poses significant security threat, but has some limitations on the extent to which they can be exploited. User level access to the system or a DoS vulnerability in a critical service would fall into this category.
Medium	The issue has significant limitations on the impact it can cause. Typically such issues would include significant information leakage, denial of service or those that allow limited access to the system.
Low	The issue represents a low level security threat. A typical issue would involve information leakage that could be useful to an attacker, such as a list of users or software version details.

Table 3. Issue Exploitability

Value	Description
Trivial	The issue requires little-to-no knowledge on behalf of an attacker and can be exploited using standard operating system tools.
Easy	The issue requires some knowledge for an attacker to exploit, which could be performed using standard operating system tools or tools downloaded from the Internet.
Moderate	The issue requires specific knowledge on behalf of an attacker. The issue could be exploited using a combination of operating system tools or publicly available tools downloaded from the Internet.
Challenging	A security issue that falls into this category would require significant effort and knowledge on behalf of the attacker. The attacker may require specific physical access to resources or to the network infrastructure in order to successfully exploit it. Furthermore, a combination of attacks may be required.
N/A	The issue is not directly exploitable.

Table 4. Risk Level Calculation

Exploitability/Impact	5 - Trivial	4 - Easy	3 - Moderate	2 - Challenging	1 - N/A
4 - Critical	20	16	12	8	4
3 - High	15	12	9	6	3
2 - Medium	10	8	6	4	2
1 - Low	5	4	3	2	1

Table 5. Risk Value

Risk Level	Risk Value
15-20	High
6-12	Medium
1-5	Low

Appendix B. Evidence

Table 6. Penetration Test Evidence

VULN-01	One-Factor User Authentication The Application login form does not require entry of any additional authentication information other than user ID and password.
	[Image redacted]

VULN-02	No Lock-Out After Failed Authentication Attempts Screenshots demonstrate that the attack imitation has been able to try passwords of different user accounts over 22 000 times.
	[Image redacted.]
	[Image redacted.]

VULN-03	SSL Certificate Is Not Trusted The web server SSL certificate is not signed by a trusted Certification Authority as shown on the screenshot below.
	[Image redacted]

VULN-04	The POODLE Attack (SSLv3 Is Supported) The Application web server supports SSL version 3 which is vulnerable to the POODLE Attack.
	[Image redacted]