DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Report

Black Box Testing Report

Alexis Engelke, Johannes Fischer, Ralph Schaumann,
Saurabh Nawalgaria

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Report

Black Box Testing Report

Author:	Alexis Engelke, Johannes Fischer, Ralph Schaumann, Saurabh Nawalgaria
Team:	9
Lecture:	Secure Coding, Phase 2
Submission Date:	November 24, 2015

Executive Summary

Contents

1 Time Tracking

Table 1.1: Time Tracking Table

Name	Task	Time
Alexis Engelke	Setting up LaTeX template	1
Foo	Fixing all issues	10

2 Vulnerabiliteis Overview

Through our testing, we identified the following vulnerabilities as the most critical for the Online Banking application and the SecureBank:

2.1 Online Banking

2.1.1 Stored XSS in Registration and Transaction Description

- Likelihood: high
- Implication: high
- Risk: high

With stored cross site scripting attacks it is possible to inject JavaScript code, which is run whenever an employee logs in and opens the list of unapproved accounts or transactions. It is also possible to inject script from other sites.

2.1.2 Missing check for amount in transactions from batch file

- Likelihood: medium
- Implication: high
- Risk: high

It is possible to get money from another client of the bank by filling in a negative number in the amount field of a transaction batch file. Therefore, one client can generate an infinite amount of money, while reducing the amount of money of other clients.

2.1.3 SQL injection in transaction batch file

- Likelihood: medium
- Implication: high

- Risk: high

The application is vulnerable to SQL injections in the transaction batch files. Therefore, it is possible to perform transactions while using any unused TAN in the system, which is not known to the attacker and might come from another client.

2.1.4 Some critical vulnerability

- Likelihood: high
- Implication: high
- Risk: high

The web application is vulnerable.

2.2 SecureBank

3 Tools

4 Detailed Report

4.1 Configuration and Deploy Management Testing

4.1.1 Test File Extensions Handling for Sensitive Information

Online Banking

Observation	We found various files which are served as plain text but are PHP source files. One of these files contains the credentials of the mail server. We were also able to download the compiled executable as well as the source code of the batch file parser.
Discovery	Using the OWASP ZAP tool, we used the forced browse functionality on `/InternetBanking/`. We received a list of files which were found using this tool, see below.
Likelihood	This can be tested by anyone who enters specific strings into the address bar of a browser. However, the likelihood of this vulnerability is much higher if the attacker uses specific tools which test specific paths systematically.
Impact	The attacker can get sensitive information, e.g. credentials to the mail server or the database. He can analyze the source of the parser and find vulnerabilies there.
Access Vector	Network
Access Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Unchanged
Confidentiality	High
Intigrity	No Impact
Availability	No Impact

TODO: Forced browsing results.
TODO: SecureBank and comparison

Acronyms

TUM Technische Universität München.