

# Mobile Application Penetration Testing Report

Prepared by:

Technology Governance & Cyber Security Management Department  
Banglalink Digital Communication Limited

Prepared for:

Banglalink Digital Communication Limited

March, 2022

**Limitation of disclosure and use of the document:**

This report contains information about potential vulnerabilities of the MyBL (My Banglalink) ) mobile application. The document gives no warranties concerning 100% accuracy, reliability, quality, correctness, or freedom from error or omission of this work. This report is delivered "as is", it shall not be liable for any inaccuracy in any condition. This paper also does not assure any form of warranty or guarantee that the system is 100% secure from security attacks that are not mentioned in the report.

**Document Details:**

Title: Mobile Application(Black Box) Penetration testing.

Application Name: MyBL (My Banglalink)

Application Type: Android

Package Name: com.arena.banglalinkmela.app

Application Version: 9.0.0.3

Project Duration: 20/02/2021 - 10/03/2022

# Contents

<b>1</b>	<b>EXECUTIVE SUMMARY:</b>	<b>1</b>
<b>2</b>	<b>Mobile APPLICATION VULNERABILITIES RESULTS:</b>	<b>3</b>
<b>3</b>	<b>SCANNING AND TESTING</b>	<b>6</b>
3.1	Automated Application scan and crawling: . . . . .	6
3.2	Manual Application Testing: . . . . .	7
<b>4</b>	<b>Web Application Detailed vulnerabilities Findings:</b>	<b>9</b>
4.1	User authentication Using the Bruteforce method: . . . . .	10
4.2	Content management admin login panel accessible via public IP. . . . .	11
4.3	Insecure 1024-bit RSA Signing-key. . . . .	13
4.4	Weak Password Policy for User Accounts. . . . .	15
4.5	Certificates signed by non-trusted are accepted. . . . .	16
4.6	Backdated Apache Server . . . . .	17
4.7	WAF set-up . . . . .	18
<b>5</b>	<b>Conculation</b>	<b>20</b>

# 1. EXECUTIVE SUMMARY:

The primary purpose of this mobile application (Black box) penetration testing was to determine any possible areas of concern associated with the mobile application and identify to which extent the system can be exploited by an attacker possessing a particular skill and motivation. This web application penetration testing was performed following the The OWASP Mobile Application Security Verification Standard (MASVS v1.4.2) and The OWASP Mobile Security Testing Guide (MSTG v1.4.0). The penetration testing was conducted from February 20 to March 10, 2020. All testing activities were conducted as an external attacker without prior knowledge of the environment and were completely isolated from the production data.

During the course of this assessment, some critical vulnerabilities were found that could lead to serious damage to the system. Along with that, some medium and low severity issues were found, which should be addressed. As some high-security flaws were found, it is recommended to remediate all high-security issues detected to mitigate against the possible risk of a sensitive data compromise. The

remediation of the low severity findings is not so urgent due to the low probability of their successful exploitation. However, the presence of these known issues could decrease the system's overall security.

**The scope of the assessment included the following:**

Components and interfaces of [My BL\(My Banglalink\)](#) android application.

Testing was performed using industry-standard penetration testing tools and frameworks. Including, Burp Suite, OWASP ZAP, Drozer, ADB, PID Cat and MobSF.

## 2. Mobile APPLICATION VULNERABILITIES RESULTS:

### **OWASP MSTG and MASTVS:**

The OWASP Mobile Security Testing Guide is an OWASP flagship project led by Carlos Holguera and Sven Schleier which defines the industry standard for mobile application security.

<https://owasp.org/www-project-mobile-security-testing-guide/>

The OWASP MASVS (Mobile Application Security Verification Standard) is a standard that establishes the security requirements for mobile app security.

<https://github.com/OWASP/owasp-mstg/>

ID	Risk Category	Number of Alerts
MSTG-1	Architecture, Design and Threat Modeling	1 Alerts
MSTG-2	Data Storage and Privacy	N/A
MSTG-3	Cryptography	2 Alerts
MSTG-4	Authentication and Session Management	3 Alerts
MSTG-5	Network Communication	N/A
MSTG-6	Platform Interaction	2 Alerts
MSTG-7	Code Quality and Build	1 Alerts
MSTG-8	Resilience	4 Alerts
	Total =	13

**Figure 2.1:** Checklist Using OWASP MSTG v1.4.0 and MASTVS v1.4.2

ID	MSTG-ID	Findings	L1	L2	R	Risk
4.6	MSTG-AUTH-6	Able to authenticate on the application Using the Bruteforce method.	✓			High
6.3	MSTG-PLATFORM-3	CMS admin login panel of this application is accessible via public IP.	✓			High
1.4	MSTG-ARCH-4	Information considered sensitive in the context of security of the mobile app is identified.		✓		Mid
3.4	MSTG-CRYPTO-4	The app uses the weak encryption mode CBC with PKCS5/PKCS7 padding.		✓		Mid
4.5	MSTG-AUTH-5	Weak password policy for user accounts.		✓		Mid
5.4	MSTG-NETWORK-4	Certificates signed by non-trusted are accepted.		✓		Mid
5.6	MSTG-NETWORK-6	The app runs on backdated Apache server version 2.4.29, which is venerable to different attacks.		✓		Mid
6.1	MSTG-PLATFORM-1	Application has set insecure Permissions.		✓		Mid
7.1	MSTG-CODE-1	The application uses a1024-bit RSA Signing-key, which is considered a security risk.		✓		Mid
8.4	MSTG-RESILIENCE-4	The application can be installed and run after reverse engineered.			✓	Mid
3.6	MSTG-CRYPTO-6	Use of Insufficiently Random Values.		✓		Low
4.11	MSTG-AUTH-11	The app does not inform the user of all sensitive activities with their account. There is no feature for users to view the list of devices, view contextual information (IP address, location, etc.), and block specific devices.		✓		low
8.1	MSTG-RESILIENCE-1	The app does not detect and responds to the presence of a rooted or jailbroken device either by alerting the user or terminating the app.			✓	Low
8.2	MSTG-RESILIENCE-2	The app does not prevent debugging and/or detects and responds when a debugger is attached.			✓	Low
8.5	MSTG-RESILIENCE-5	The app does not detect and respond to being run in an emulator.			✓	Low
Overall Risk = High						

Figure 2.2: Summary of Vulnerabilities Found



## **3. SCANNING AND TESTING**

### **3.1 Automated Application scan and crawling:**

Several commercial tools were used to scan the targeted scope and identify potential vulnerabilities. The automated scanning software identifies application-level vulnerabilities. The web application crawling covered the following:

- Parameter Injection
- SQL Injection
- Cross-Site Scripting
- Directory Traversal
- Parameter Overflow
- Buffer Overflow
- Parameter Addition
- Path Manipulation
- Character Encoding
- Site Search
- SSL Strength

- Sensitive Developer Comments
- Web Server Identification
- Web Package Identification
- Permissions Assessment
- Brute Force Authentication attacks

## 3.2 Manual Application Testing:

Using the information created by the automated testing software, some manual testing was done to identify and exploit additional vulnerabilities in the targeted application and eliminate false positives produced by the automatic scanning methods. Following actions were performed as part of manual testing:

- The gathered information about the application
- Checked types and placement of security controls
- Mapped application content and analyzed it
- Tested application authentication, session management and access control
- Tested application for input based vulnerabilities
- Tested application for client data validation issues
- Checked for application server vulnerabilities
- Tested HTTP/URL manipulation
- Tested SQL command injection
- Tested cross-site scripting

- 
- Tested Cross-site request forgery
  - Tested DOS attacks
  - Tested Parameter overflow and handling

## **4. Web Application Detailed vulnerabilities Findings:**

## 4.1 User authentication Using the Brute force method:

**Classification:** High Risk

**CWE ID:** CWE-307

Improper Restriction of Excessive Authentication Attempts.

**OWASP MSTG-ID:** MSTG-AUTH-6

**Description:**

The software does not implement sufficient measures to prevent multiple failed authentication attempts within a short time frame for password-based login, making it susceptible to brute force attacks. The application allowed multiple authentication requests when tested against a brute force cluster bomb attack using Burp Suite intruder. As the number is fixed for a particular user, that makes it very easy for an attacker to use regex in the password and perform a successful brute force attack.

**Solution:**

Solution: An IP address should be blocked for a time span for password-based login after 3 to 5 failed authentication attempts. It is recommended that for the time being, only OTP-based login should be enabled before fixing the password-based login system.

38	Mns78931.	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
39	Mns78933.	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
40	Mns78932.	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
41	aghahs1234	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	631
42	12123asdf	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
43	www1234	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
44	Mns78931	01404400712	200	<input type="checkbox"/>	<input type="checkbox"/>	2876
45	5654absa	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
46	yety5656	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
47	treyeq456	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611
48	gfgfs5657	01404400712	401	<input type="checkbox"/>	<input type="checkbox"/>	611

Request	Response
---------	----------

Raw	Headers	Hex
-----	---------	-----

```

HTTP/1.1 200 OK
Date: Tue, 01 Mar 2022 05:30:03 GMT
Server: Apache/2.4.41 (Unix)
X-Powered-By: PHP/7.3.11
Cache-Control: no-cache, private
Access-Control-Allow-Origin: https://mybl.banglalinkr.net
Access-Control-Allow-Headers: X-PINGOTHER, Content-Type, authorization, api-client-pass, pl
Access-Control-Allow-Methods: POST, GET, OPTIONS
Keep-Alive: timeout=5, max=73
Connection: Keep-Alive
Content-Type: application/json
Content-Length: 2399

{"status": "SUCCESS", "status_code": 200, "message": "Login
Successfully", "data": {"token": {"token_type": "Bearer", "expires_in": 31536000, "access_token": "s

```

Figure 4.1: login using brute force

## 4.2 Content management admin login panel accessible via public IP.

**Classification:** High Risk

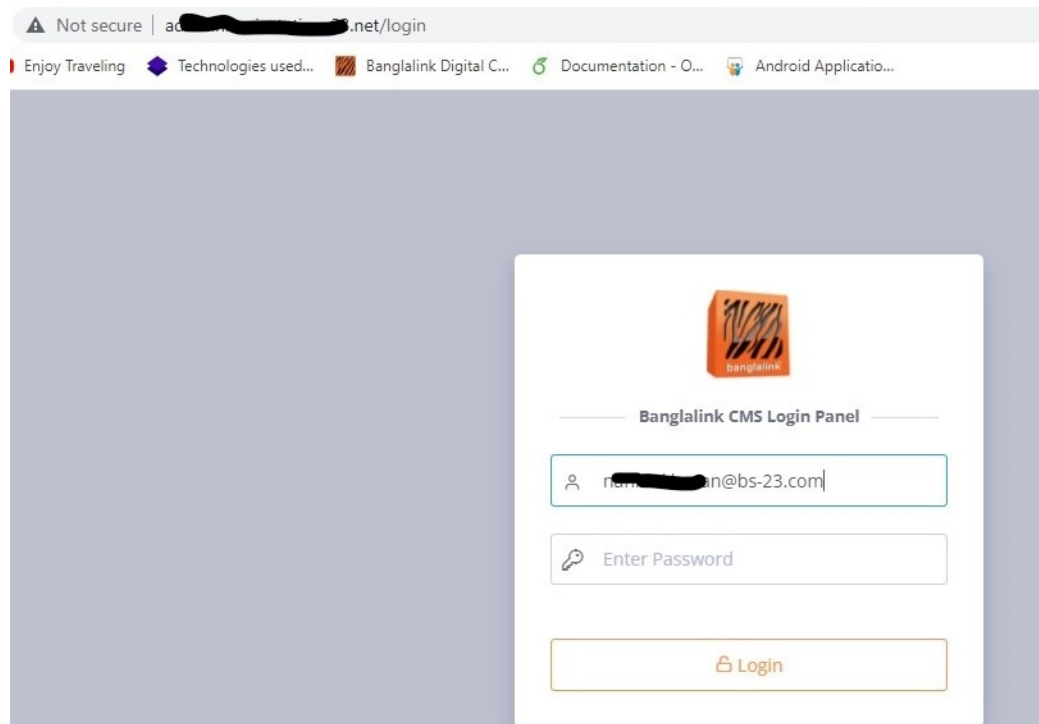
**CWE ID:** 284

Improper Access Control

**OWASP MSTG-ID:** MSTG-PLATFORM-3

### Description:

This application's content management system admin panel is accessible via any



**Figure 4.2:** CMS Admin Login panel

public IP address. If we dig further, it reveals some sensitive information that might help a remote attacker break into the admin panel of the CMS of this Application. So the site does not use a secure transmission protocol.

**Solution:**

Consult the virtual host configuration and check if this virtual host should be publicly accessible.

Server/Request Data	
REDIRECT_STATUS	"200"
HTTP_HOST	"admin.brainstation-23.net"
HTTP_X_FORWARDED_SCHEME	"http"
HTTP_X_FORWARDED_PROTO	"http"
HTTP_X_FORWARDED_FOR	"103.67.158.163"
HTTP_X_REAL_IP	"103.67.158.163"
HTTP_CONNECTION	"close"
CONTENT_LENGTH	"88"

3/2/22, 10:44 AM	Whoops! There was an error.
HTTP_CACHE_CONTROL	"max-age=0"
HTTP_UPGRADE_INSECURE_REQUESTS	"1"
HTTP_ORIGIN	"http://admin.brainstation-23.net"
CONTENT_TYPE	"application/x-www-form-urlencoded"
HTTP_USER_AGENT	"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, l
HTTP_ACCEPT	"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
HTTP_REFERER	"http://admin.brainstation-23.net/login"
HTTP_ACCEPT_ENCODING	"gzip, deflate"
HTTP_ACCEPT_LANGUAGE	"en-US,en;q=0.9"
HTTP_COOKIE	"XSRF-TOKEN=eyJpdii6I1JNqlwOUx50GRjQmLFNTRQQXBHlWvQT09IiwidmFsdWUiOiJ
PATH	"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
SERVER_SIGNATURE	"<address>Apache/2.4.29 (Ubuntu) Server at admin.brainstation-23.net P
SERVER_SOFTWARE	"Apache/2.4.29 (Ubuntu)"
SERVER_NAME	"admin.brainstation-23.net"
SERVER_ADDR	"192.168.104.238"
SERVER_PORT	"80"
REMOTE_ADDR	"172.16.229.109"
DOCUMENT_ROOT	"/var/www/html/bl_cms/public"
REQUEST_SCHEME	"http"
CONTEXT_PREFIX	""
CONTEXT_DOCUMENT_ROOT	"/var/www/html/bl_cms/public"
SERVER_ADMIN	"[REDACTED]@bs-23.com"

Figure 4.3: CMS Admin panel information

## 4.3 Insecure 1024-bit RSA Signing-key.

**Classification:** Medium Risk

**CWE ID:** 327

Use of a Broken or Risky Cryptographic Algorithm

**OWASP MSTG-ID:** MSTG-CODE-1



```

APK is signed
v1 signature: True
v2 signature: True
v3 signature: True
Found 1 unique certificates
Subject: C=BD, ST=dhaka, L=dhaka, O=banglalink, OU=banglalink, CN=banglalink
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2014-02-25 15:27:42+00:00
Valid To: 2039-02-19 15:27:42+00:00
Issuer: C=BD, ST=dhaka, L=dhaka, O=banglalink, OU=banglalink, CN=banglalink
Serial Number: 0x530cb66e
Hash Algorithm: sha1
md5: deb75dad27cf4da195559333c3aab0c1
sha1: eb748a82d4398c4df6962f7380988ea9d4e073dc
sha256: a1a6d472561dae3b8816cb36f4dbe560b601a847288aa4e926093110ca3ffed9
sha512: 1b252f8f10364c37b735ce8bdce394a82fb5d7be125e799b1dd844114980532b36dfec4aa7213d5
PublicKey Algorithm: rsa
Bit Size: 1024
Fingerprint: 21188bafd3bcad0e1ec04f2fc5467f444dbec47c26093a7309388fd6b0d66a8e|
Is CN=banglalink, OU=banglalink, O=banglalink, L=dhaka, ST=dhaka, C=BD correct?
[no]: yes
Generating 1,024 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 2,542 days
for: CN=banglalink, OU=banglalink, O=banglalink, L=dhaka, ST=dhaka, C=BD
Warning:
The generated certificate uses a 1024-bit RSA key which is considered a security risk. This key size will be disabled in a future update.

```

**Figure 4.4:** Application digital signature

### Description:

The application uses a 1024-bit RSA Signing-key according to MobsF, which is no longer considered as a secure key for application certificate signing. The use of a non-standard algorithm is dangerous because a determined attacker may break the algorithm and compromise whatever data has been protected.



### Solution:

Solution: The application's digital signature key should be updated with a more secure signing algorithm.


Change Password

Password must be minimum 8 characters and contain at least one letter and one number

Old Password

••••••••  

---

 Password must be minimum 8 characters and contain at least one letter and one number

**Figure 4.5:** Password Policy of MyBL

## 4.4 Weak Password Policy for User Accounts.

**Classification:** Medium Risk

**CWE ID:** 521

Weak Password Requirements

**OWASP MSTG-ID:** MSTG-AUTH-5

### **Description:**

This application does not require that users have strong passwords, making it easier for attackers to compromise user accounts. Therefore, this password must be of sufficient complexity and impractical for an adversary to guess. Selecting the correct password requirements and enforcing them through implementation are critical to the overall success of the authentication mechanism. The user account password in this application is not mandatory to use upper case, lower case or special characters, making it easy for brute force attacks.

**Solution:**

Password should contain lower case, upper case alphabet, special character, along with the current policy to make it stronger.

## 4.5 Certificates signed by non-trusted are accepted.

**Classification:** Medium Risk

**CWE ID:** 295

Improper Certificate Validation

**OWASP MSTG-ID:** MSTG-AUTH-5

**Description:**

While reverse-engineering the software, the application accepted and worked with the digital signing certificate provided.

**Solution:**

There should be a mechanism to prevent application certificates signed from non

```
>>> Signer
X.509, CN=banglalink, OU=banglalink, O=banglalink, L=dhaka, ST=dhaka, C=BD
Signature algorithm: SHA256withRSA, 1024-bit key (weak)
[trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.
The RSA signing key has a keysize of 1024 which is considered a security risk. This key size will be disabled in a future update.
D:\Pen_Testing Tool\Reverse eng apk\dex-tools-2.1\mybl\dist>
```

**Figure 4.6:** Signing MyBL App

trusted, or there should be some warning for non-trusted certificated applications during installation.

## 4.6 Backdated Apache Server

**Classification:** Medium Risk

**CWE ID:** 295

Improper Certificate Validation

**OWASP MSTG-ID:** MSTG-AUTH-5

### **Description:**

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application. CSP can be used to prevent and/or mitigate attacks that involve content/code injection, such as cross-site scripting/XSS attacks, attacks that require embedding a malicious resource, attacks that involve malicious use of iframes, such as clickjacking attacks, and others. It is a good thing that the website is well protected to different injection attacks.

### **Solution:**

It's recommended to implement Content Security Policy (CSP) into your web

Web Server
Details
Request headers
GET / HTTP/1.1 Referer: https://www.banglalink.net/ Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.banglalink.net User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21 Connection: Keep-alive

**Figure 4.7: CSP not Found**

application. Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

## 4.7 WAF set-up

**Classification:** low Risk

**Source:** www.banglalink.net

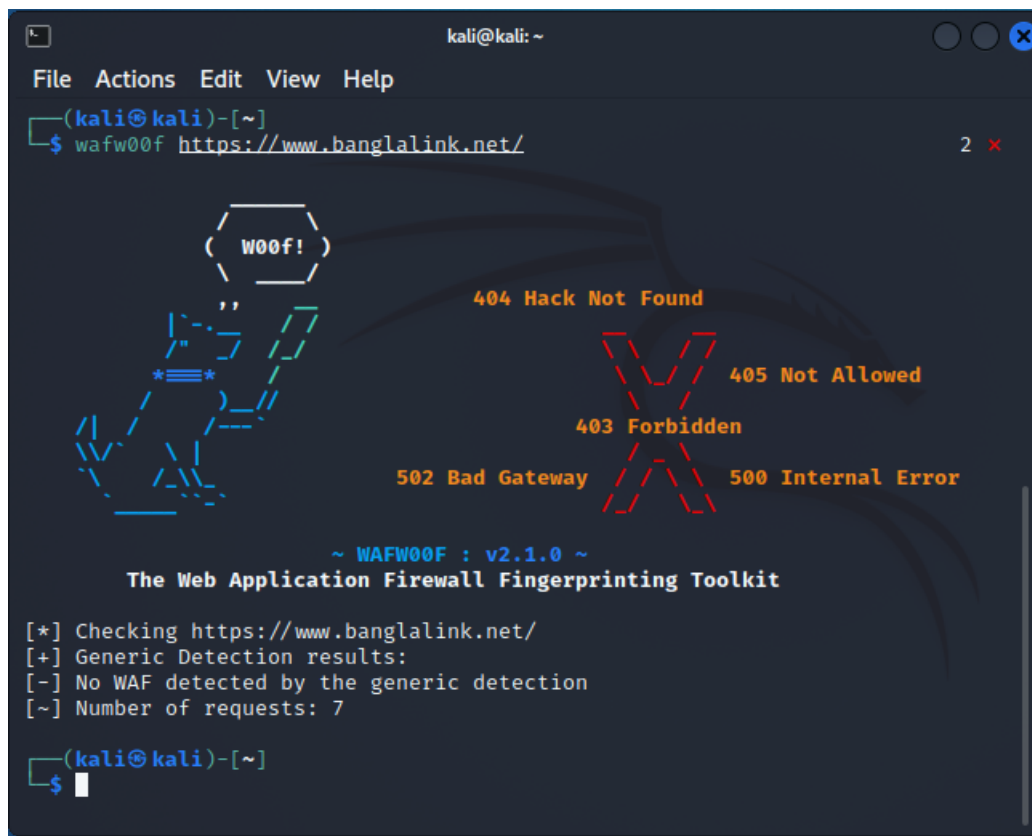
**Alert Tag:** OWASP 2021 A05

**Description:**

WAF is a "safety net" and may provide "virtual patching" until the application code is fix. A misconfigured WAF will also bring a false sense of security. A well configure WAF will provide more time for developer to fix their code. It seems WAF is not configured correctly on the web application.

**Solution:**

Firewall should configure correctly to ensure better security.



### Figure 4.8: WAFW00F Scan Result

## 5. Conculation

All the testings in this report were done based on the external knowledge of the web application as part of black-box testing. All the security issues discovered during that activity were explored and defined in this report. Please note that as technologies and risks alter over time, the vulnerabilities associated with the operation of systems described in this report and the actions necessary to reduce the exposure to such vulnerabilities will also change.