Mobile Application Security

Assignment

Submitted By

Raj Shekhar

Enrolment no.: 240545002004

M.Sc. in Cyber Security

Semester II

Guided By

Mrs. Meena Lakshmi

(Assistant Professor)

School of Cyber Security & Digital Forensics

National Forensic Science University

Bhopal Campus, M.P., 462001, India

MobSF Static Analysis Report Summary : PIVAA APK

MobSF (Mobile Security Framework) is an open-source, automated security testing tool designed to comprehensively assess the security posture of Android, iOS, and Windows Mobile applications. It employs a multi-layered approach, combining static and dynamic analysis techniques to identify vulnerabilities, malware risks, and compliance issues.

It capabilities demonstrated in this Report are: Static Analysis, Permission Mapping, Behavioral Analysis, Malware Indicators, Automated Reporting.

Security Score: 40/100 (*Medium Risk*) | **Grade:** B

Key Findings

1. Certificate & Signing Issues : Signed with a **debug certificate, Uses insecure v1 signature scheme,** and **Weak SHA1 with RSA algorithm**.

2. Manifest Vulnerabilities : **Debuggable flag enabled**, Targets outdated **Android 4.4**, Exported components (Service, Broadcast Receiver, Content Provider) are **unprotected**.

3. Permissions & Data Risks : Requests **dangerous permissions** (e.g., camera, location, contacts), Logs sensitive information and uses **hardcoded secrets** in code.

4. Code-Level Weaknesses : **SQL injection risks** due to raw SQL queries, Uses **weak cryptography** like ECB (is predictable), CBC with PKCS (is vulnerable to padding oracle)

5. Network & Behavior : SSL pinning detected, and also Records audio/video and accesses files.

The above is the finding of the MobSF report (check below) for PIVAA, we can also all verify that using drozer (some of them are shown below figures). Also, this app exhibits significant security gaps, particularly in cryptography, permissions, and component protection. Addressing these issues would elevate its security score and reduce exploitation risks.

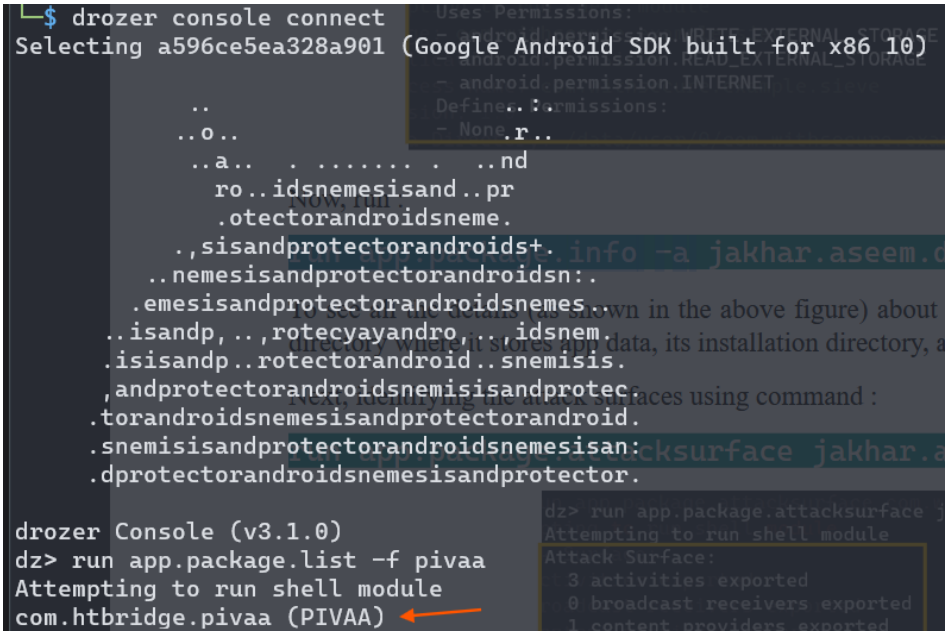

Fig: using drozer console connected with drozer agent find pivaa

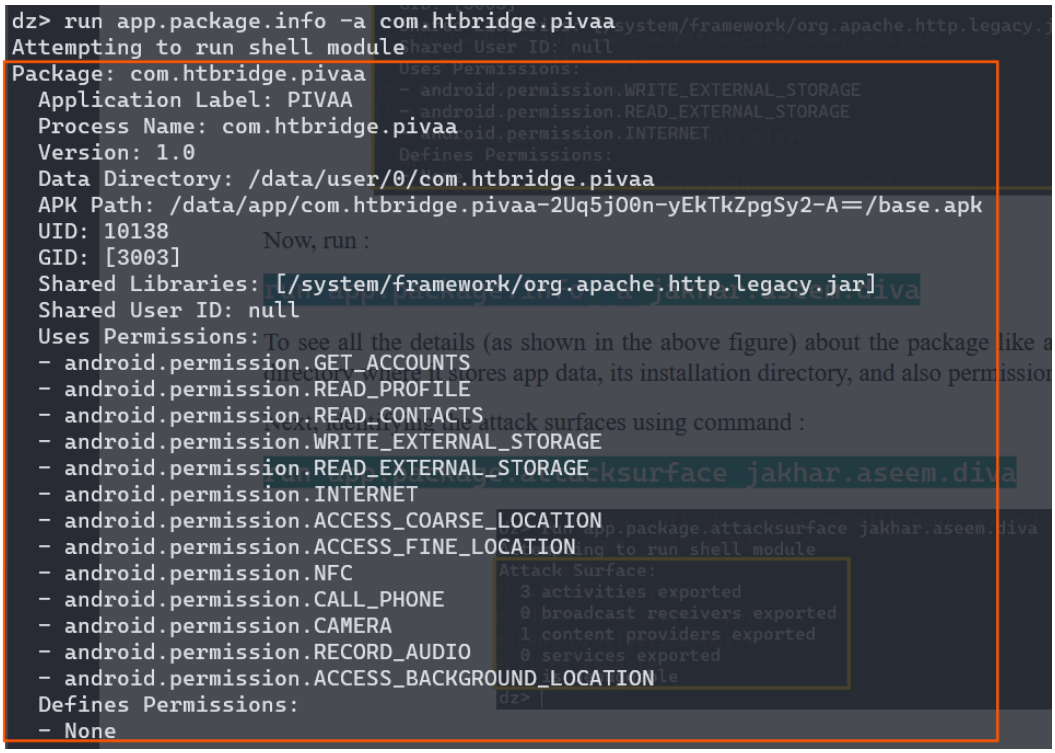

Fig: Retrieving information about pivaa package

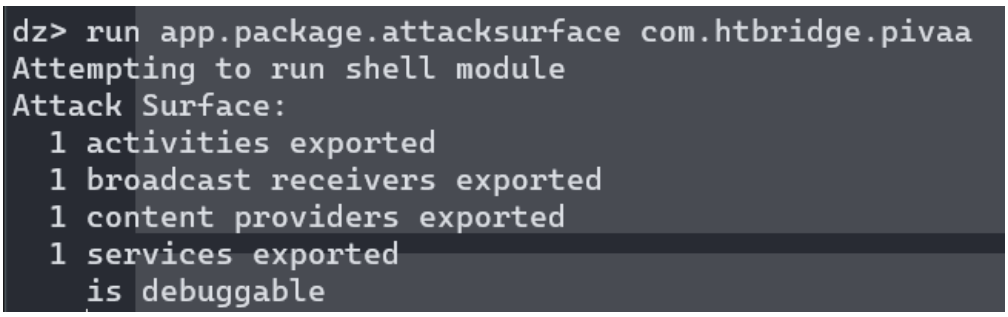

Fig: Checking for attack surfaces like activities, content providers and services

```
dz> run app.provider.info -a com.htbridge.pivaa
Attempting to run shell module
Package: com.htbridge.pivaa
  Authority: com.htbridge.pivaa
    Read Permission: null
    Write Permission: null
    Content Provider: com.htbridge.pivaa.handlers.VulnerableContentProvider
    Multiprocess Allowed: False
    Grant Uri Permissions: True
```

Fig: Checking for exported Content Providers in pivaa

These are some information that are enough to cross check the MobSF report and the report cover the contents as follows :

a. FILE & APP INFORMATION
b. APP COMPONENTS
c. CERTIFICATE INFORMATION & ANALYSIS
d. APPLICATION PERMISSIONS
e. APKID ANALYSIS
f. MANIFEST ANALYSIS
g. CODE ANALYSIS
h. BEHAVIOUR ANALYSIS & PERMISSIONS MANIPULATED
i. DOMAIN MALWARE CHECK
j. HARDCODED SECRETS
k. LOGS

Below table suggests some urgent fixes must be done asap :

Category	**Issue**	**Fix**
Certificate	Weak SHA1withRSA hashing	Upgrade to SHA256withRSA
Manifest	Debuggable flag enabled	Set android:debuggable="false" in manifest
Cryptography	Insecure random number generator	Replace with SecureRandom
Code Quality	Hardcoded secrets in code	Remove hardcoded values; use encrypted keystore
	SQL injection risks	Use parameterized queries
Network Security	HTTP communication risks	Enforce HTTPS and implement SSL certificate pinning

File Name:	pivaa.apk
Package Name:	com.htbridge.pivaa
Scan Date:	April 14, 2025, 7:49 a.m.
App Security Score:	40/100 (MEDIUM RISK)
Grade:	B

FINDINGS SEVERITY

HIGH	MEDIUM	INFO	SECURE	HOTSPOT
7	12	1	2	1

FILE INFORMATION

File Name: pivaa.apk
Size: 3.02MB
MD5: eaade08f941e47b08cfbce65c37895d6
SHA1: 2bc8ccf3185f5387097c29bfd79453bdb08b4457
SHA256: 57887e1d1e119939eec0e929801b049f8037cf90d2accab479a48f0d4dd2c19a

APP INFORMATION

App Name: PIVAA
Package Name: com.htbridge.pivaa
Main Activity: com.htbridge.pivaa.MainActivity
Target SDK: 26
Min SDK: 19
Max SDK:
Android Version Name: 1.0
Android Version Code: 1

APP COMPONENTS

Activities: 10
Services: 1
Receivers: 1
Providers: 1
Exported Activities: 0
Exported Services: 1
Exported Receivers: 1
Exported Providers: 1

CERTIFICATE INFORMATION

Binary is signed
v1 signature: True
v2 signature: True
v3 signature: False
v4 signature: False
X.509 Subject: CN=Android Debug, O=Android, C=US
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2017-11-21 14:47:12+00:00
Valid To: 2047-11-14 14:47:12+00:00
Issuer: CN=Android Debug, O=Android, C=US
Serial Number: 0x1
Hash Algorithm: sha1
md5: 0197d3002ac9ac35e4f3947e5e5f456a
sha1: 6747332aa82c3ec0e9b74d9cfbeaec56a094c563
sha256: 918b23e3bf7c966db1906bd25d0deaeb42ed8ef76ad609a79c90c7f59ac8dcfd
sha512: e0930a700698ad149fd5dec650745cb0f7420f6e3c62bc62c134233b8f10caaa6368966e00617eb575e4fbdc5fbe60fe29f3fe40ca5e805412604b40e52c6c30
PublicKey Algorithm: rsa
Bit Size: 1024
Fingerprint: 701570d3726da7399372db54956f75edadf0e193179f35e94cc80532e14dbc13
Found 1 unique certificates

APPLICATION PERMISSIONS

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.GET_ACCOUNTS	dangerous	list accounts	Allows access to the list of accounts in the Accounts Service.
android.permission.READ_PROFILE	dangerous	read the user's personal profile data	Allows an application to read the user's personal profile data.
android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people.
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows an application to read from external storage.
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.
android.permission.NFC	normal	control Near-Field Communication	Allows an application to communicate with Near-Field Communication (NFC) tags, cards and readers.

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.CALL_PHONE	dangerous	directly call phone numbers	Allows the application to call phone numbers without your intervention. Malicious applications may cause unexpected calls on your phone bill. Note that this does not allow the application to call emergency numbers.
android.permission.CAMERA	dangerous	take pictures and videos	Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.
android.permission.RECORD_AUDIO	dangerous	record audio	Allows application to access the audio record path.

APKID ANALYSIS

FILE	DETAILS
assets/com/htbridge/raw/ExternalCode.jar!classes.dex	FINDINGS: Compiler — DETAILS: unknown (please file detection issue!)
classes.dex	FINDINGS: Compiler — DETAILS: dx (possible dexmerge) FINDINGS: Manipulator Found — DETAILS: dexmerge

NETWORK SECURITY

NO	SCOPE	SEVERITY	DESCRIPTION

CERTIFICATE ANALYSIS

HIGH: 1 | WARNING: 2 | INFO: 1

TITLE	SEVERITY	DESCRIPTION
Signed Application	info	Application is signed with a code signing certificate
Application vulnerable to Janus Vulnerability	warning	Application is signed with v1 signature scheme, making it vulnerable to Janus vulnerability on Android 5.0-8.0, if signed only with v1 signature scheme. Applications running on Android 5.0-7.0 signed with v1, and v2/v3 scheme is also vulnerable.
Application signed with debug certificate	high	Application signed with a debug certificate. Production application must not be shipped with a debug certificate.
Certificate algorithm might be vulnerable to hash collision	warning	Application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues. The manifest file indicates SHA256withRSA is in use.

MANIFEST ANALYSIS

HIGH: 2 | WARNING: 4 | INFO: 0 | SUPPRESSED: 0

NO	ISSUE	SEVERITY	DESCRIPTION
1	App can be installed on a vulnerable upatched Android version Android 4.4-4.4.4, [minSdk=19]	high	This application can be installed on an older version of android that has multiple unfixed vulnerabilities. These devices won't receive reasonable security updates from Google. Support an Android version => 10, API 29 to receive reasonable security updates.
2	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.
3	Application Data can be Backed up [android:allowBackup=true]	warning	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.
4	Service (com.htbridge.pivaa.handlers.VulnerableService) is not Protected. [android:exported=true]	warning	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.
5	Broadcast Receiver (com.htbridge.pivaa.handlers.VulnerableReceiver) is not Protected. [android:exported=true]	warning	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.
6	Content Provider (com.htbridge.pivaa.handlers.VulnerableContentProvider) is not Protected. [android:exported=true]	warning	A Content Provider is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.

</> CODE ANALYSIS

HIGH: **4** | WARNING: **6** | INFO: **1** | SECURE: **1** | SUPPRESSED: **0**

NO	ISSUE	SEVERITY	STANDARDS	FILES
1	The App logs information. Sensitive information should never be logged.	info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3	com/htbridge/pivaa/AboutActivity.java com/htbridge/pivaa/ContentProviderActivity.java com/htbridge/pivaa/DatabaseActivity.java com/htbridge/pivaa/EncryptionActivity.java com/htbridge/pivaa/LoadCodeActivity.java com/htbridge/pivaa/MainActivity.java com/htbridge/pivaa/WebviewActivity.java com/htbridge/pivaa/handlers/Authentication.java com/htbridge/pivaa/handlers/Encryption.java com/htbridge/pivaa/handlers/LoadCode.java com/htbridge/pivaa/handlers/ObjectSerialization.java com/htbridge/pivaa/handlers/VulnerableContentProvider.java com/htbridge/pivaa/handlers/VulnerableReceiver.java com/htbridge/pivaa/handlers/VulnerableService.java com/htbridge/pivaa/handlers/about/AboutAdapter.java com/htbridge/pivaa/handlers/database/DatabaseAdapter.java com/htbridge/pivaa/handlers/database/DatabaseHelper.java

NO	ISSUE	SEVERITY	STANDARDS	FILES
2	Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	warning	CWE: CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	com/htbridge/pivaa/Configuration.java com/htbridge/pivaa/handlers/Authentication.java
3	App can read/write to External Storage. Any App can read data written to External Storage.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/htbridge/pivaa/handlers/Authentication.java com/htbridge/pivaa/handlers/VulnerableService.java
4	App creates temp file. Sensitive information should never be written into a temp file.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/htbridge/pivaa/handlers/Authentication.java
5	The file or SharedPreference is World Writable. Any App can write to the file	high	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/htbridge/pivaa/handlers/Authentication.java
6	App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.	warning	CWE: CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	com/htbridge/pivaa/handlers/database/DatabaseHelper.java
7	Ensure that user controlled URLs never reaches the Webview. Enabling file access from URLs in WebView can leak sensitive information from the file system.	warning	CWE: CWE-200: Information Exposure OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: MSTG-PLATFORM-7	com/htbridge/pivaa/BroadcastReceiverActivity.java com/htbridge/pivaa/WebviewActivity.java
8	Debug configuration enabled. Production builds must not be debuggable.	high	CWE: CWE-919: Weaknesses in Mobile Applications OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: MSTG-RESILIENCE-2	com/htbridge/pivaa/BuildConfig.java

NO	ISSUE	SEVERITY	STANDARDS	FILES
9	The App uses an insecure Random Number Generator.	warning	CWE: CWE-330: Use of Insufficiently Random Values OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-6	com/htbridge/pivaa/handlers/Encryption.java
10	The App uses ECB mode in Cryptographic encryption algorithm. ECB mode is known to be weak as it results in the same ciphertext for identical blocks of plaintext.	high	CWE: CWE-327: Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-2	com/htbridge/pivaa/handlers/Encryption.java
11	The App uses the encryption mode CBC with PKCS5/PKCS7 padding. This configuration is vulnerable to padding oracle attacks.	high	CWE: CWE-649: Reliance on Obfuscation or Encryption of Security-Relevant Inputs without Integrity Checking OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-3	com/htbridge/pivaa/handlers/Encryption.java
12	This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.	secure	OWASP MASVS: MSTG-NETWORK-4	com/htbridge/pivaa/handlers/API.java

NIAP ANALYSIS v1.3

NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION

BEHAVIOUR ANALYSIS

RULE ID	BEHAVIOUR	LABEL	FILES
00013	Read file and put it into a stream	file	com/htbridge/pivaa/handlers/Authentication.java com/htbridge/pivaa/handlers/ObjectSerialization.java
00022	Open a file from given absolute path of the file	file	com/htbridge/pivaa/handlers/LoadCode.java com/htbridge/pivaa/handlers/VulnerableService.java
00195	Set the output path of the recorded file	record file	com/htbridge/pivaa/handlers/VulnerableService.java
00199	Stop recording and release recording resources	record	com/htbridge/pivaa/handlers/VulnerableService.java
00198	Initialize the recorder and start recording	record	com/htbridge/pivaa/handlers/VulnerableService.java
00194	Set the audio source (MIC) and recorded file format	record	com/htbridge/pivaa/handlers/VulnerableService.java
00197	Set the audio encoder and initialize the recorder	record	com/htbridge/pivaa/handlers/VulnerableService.java
00007	Use absolute path of directory for the output media file path	file	com/htbridge/pivaa/handlers/VulnerableService.java
00196	Set the recorded file format and output path	record file	com/htbridge/pivaa/handlers/VulnerableService.java
00041	Save recorded audio/video to file	record	com/htbridge/pivaa/handlers/VulnerableService.java
00089	Connect to a URL and receive input stream from the server	command network	com/htbridge/pivaa/handlers/API.java

ABUSED PERMISSIONS

TYPE	MATCHES	PERMISSIONS
Malware Permissions	9/25	android.permission.GET_ACCOUNTS, android.permission.READ_CONTACTS, android.permission.WRITE_EXTERNAL_STORAGE, android.permission.READ_EXTERNAL_STORAGE, android.permission.INTERNET, android.permission.ACCESS_COARSE_LOCATION, android.permission.ACCESS_FINE_LOCATION, android.permission.CAMERA, android.permission.RECORD_AUDIO
Other Common Permissions	1/44	android.permission.CALL_PHONE

Malware Permissions:
Top permissions that are widely abused by known malware.
Other Common Permissions:
Permissions that are commonly abused by known malware.

OFAC SANCTIONED COUNTRIES

This app may communicate with the following OFAC sanctioned list of countries.

DOMAIN	COUNTRY/REGION

DOMAIN MALWARE CHECK

DOMAIN	STATUS	GEOLOCATION
www.htbridge.com	ok	**IP:** 172.67.206.240 **Country:** United States of America **Region:** California **City:** San Francisco **Latitude:** 37.775700 **Longitude:** -122.395203 **View:** Google Map
xss.rocks	ok	**IP:** 104.21.2.163 **Country:** United States of America **Region:** California **City:** San Francisco **Latitude:** 37.775700 **Longitude:** -122.395203 **View:** Google Map
google.com	ok	**IP:** 216.58.210.142 **Country:** United States of America **Region:** California **City:** Mountain View **Latitude:** 37.405991 **Longitude:** -122.078514 **View:** Google Map

HARDCODED SECRETS

POSSIBLE SECRETS
013602400437880159360205O5

SCAN LOGS

Timestamp	Event	Error
2025-04-14 07:49:33	Generating Hashes	OK
2025-04-14 07:49:33	Extracting APK	OK
2025-04-14 07:49:33	Unzipping	OK
2025-04-14 07:49:33	Parsing APK with androguard	OK
2025-04-14 07:49:33	Extracting APK features using aapt/aapt2	OK
2025-04-14 07:49:33	Getting Hardcoded Certificates/Keystores	OK
2025-04-14 07:49:36	Parsing AndroidManifest.xml	OK
2025-04-14 07:49:36	Extracting Manifest Data	OK
2025-04-14 07:49:36	Manifest Analysis Started	OK

2025-04-14 07:49:36	Performing Static Analysis on: PIVAA (com.htbridge.pivaa)	OK
2025-04-14 07:49:36	Fetching Details from Play Store: com.htbridge.pivaa	OK
2025-04-14 07:49:36	Checking for Malware Permissions	OK
2025-04-14 07:49:36	Fetching icon path	OK
2025-04-14 07:49:36	Library Binary Analysis Started	OK
2025-04-14 07:49:36	Reading Code Signing Certificate	OK
2025-04-14 07:49:37	Running APKiD 2.1.5	OK
2025-04-14 07:49:38	Detecting Trackers	OK
2025-04-14 07:49:39	Decompiling APK to Java with JADX	OK
2025-04-14 07:49:55	Converting DEX to Smali	OK
2025-04-14 07:49:55	Code Analysis Started on - java_source	OK

2025-04-14 07:49:55	Android SBOM Analysis Completed	OK
2025-04-14 07:50:26	Android SAST Completed	OK
2025-04-14 07:50:26	Android API Analysis Started	OK
2025-04-14 07:50:58	Android API Analysis Completed	OK
2025-04-14 07:50:59	Android Permission Mapping Started	OK
2025-04-14 07:51:00	Android Permission Mapping Completed	OK
2025-04-14 07:51:01	Android Behaviour Analysis Started	OK
2025-04-14 07:51:02	Android Behaviour Analysis Completed	OK
2025-04-14 07:51:02	Extracting Emails and URLs from Source Code	OK
2025-04-14 07:51:02	Email and URL Extraction Completed	OK
2025-04-14 07:51:02	Extracting String data from APK	OK

2025-04-14 07:51:02	Extracting String data from Code	OK
2025-04-14 07:51:02	Extracting String values and entropies from Code	OK
2025-04-14 07:51:03	Performing Malware check on extracted domains	OK
2025-04-14 07:51:03	Saving to Database	OK

Report Generated by - MobSF v4.3.2

Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis.