



APP SECURITY ANALYSIS REPORT

Northcap

Napp v1.1.0

Document date: 09/19/2024

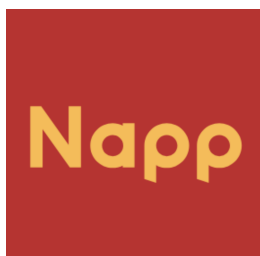


TABLE OF CONTENTS

1. HOW TO READ THE REPORT
2. EXPLORE BEYOND
3. DYNAMIC ANALYSIS: SUMMARY
4. EVIDENCES
5. SECURITY SCORE

STRICTLY CONFIDENTIAL

1. HOW TO READ THE REPORT

Each vulnerability is assigned a criticality score ranging from 0 to 10 based on the CVSS v3.1 standards. This score considers the ease of exploit implementation and the potential risk in case of a successful exploit. The score is then recalculated by factoring in parameters that account for the client's reputation, business, and policy areas. This results in a real risk score that accurately measures the level of risk. Based on the resulting score, the vulnerability is assigned a risk level, which is indicated in the table under the column "Range".

LEVEL	DESCRIPTION	RANGE
	No vulnerability	0
	Low vulnerability	0.1 – 3.9
	Medium vulnerability	4 – 6.9
	High vulnerability	7 – 8.9
	Critical vulnerability	9 – 10

Mobisec has come up with three risk flags that show when evidence, regardless of its score, is the result of a **bug**, when it might pose a **risk to the business**, and when it needs **special attention** because it could have a negative impact on compliance, best practices, market standards, or more.

Any personal, sensitive, or confidential data collected during testing may appear in dynamic analysis evidence.

This data will potentially be obfuscated using the following notation: ***.

All the vulnerabilities we have found are grouped according to the OWASP MASVS v2.1 standard.

MOBISec FLAGS



Bug



Business Risk



Caution

STANDARD MASVS

MASVS - STORAGE

MASVS - PLATFORM

MASVS - CRYPTO

MASVS - CODE

MASVS - AUTH

MASVS - RESILIENCE

MASVS - NETWORK

MASVS - PRIVACY

STRICTLY CONFIDENTIAL

2. EXPLORE BEYOND

We investigate the application from a unique perspective, exactly as an external hacker would do. This analysis reveals points of interest, vulnerabilities and strengths, providing a strategic advantage not available internally. This privileged view is critical to accurately and confidently address risks. This is essential to prevent, address and overcome cyber security challenges. An outside view to understand current vulnerabilities and anticipate future threats ensures an intelligent and proactive security path.

Our analysis

The application is native in Swift and includes external links that, by invoking the system browser (Safari), lead to the display of pages such as the privacy policy, FAQs, or data modification. This method of opening external links introduces risks related to the browser, which could be reduced by using a WebView through WKWebView, although the most secure solution would still be to create native screens (activities) for managing this data.

The app allows access to all the services offered by the company, including digital payments, viewing recent purchases, geolocation through MapKit and REST API, viewing nearby points of interest, accessing new promotions, and inviting other people to sign up.

All network communications take place over the HTTPS protocol, ensuring that the certificate is signed by a CA. REST API calls are primarily made using the GET method, while the POST method is used for sensitive operations such as login, session token renewal, and inputting promotional codes.

Payments are handled via proprietary PayPal REST APIs, which support both PayPal accounts and credit cards. The payment process uses a QR code and a token that changes with each session, further strengthened by PayPal's server-side proxy detection, reducing the risk of Man-In-The-Middle (MITM) attacks.

The app's filesystem is partially managed correctly: files and cache are encrypted, but user preference files are not protected by encryption, which could expose sensitive information. However, the integrity of this data is guaranteed by a consistency check performed by the app at each startup: if the local files do not match those on the server, they are re-downloaded.

The initial login is protected by a maximum attempt limit, reducing the risk of brute-force attacks and making the app less vulnerable to phishing or identity theft techniques.

Notes

When attempting to use the **Recover Password** function, the message *"The email has been sent successfully"* is always returned, even when entering email addresses that are not present in the database. While this is a security advantage, as it prevents potential attackers from discovering which emails are in the database, we suggest considering a different and more realistic communication if the user enters an incorrect or non-existent email address.

STRICTLY CONFIDENTIAL

3. DYNAMIC ANALYSIS: SUMMARY

The first column shows the evidence number in the risk analysis section. The second column shows the actual severity, indicated in the synopsis (SE), with the CVSS v3.1 score. The third column indicates the simplicity of possible exploit implementation (EX) on a scale of 1 to 4, with a higher value indicating greater difficulty of implementation.

ID	SE	EX	DETAILS
NAP-007	9.9	1	Authorization control is not implemented correctly and allows access to other users' data.
NAP-008	6.5	2	The application has no certificate pinning, so it is vulnerable to Man-in-the-Middle (MITM) attacks capable of reading and modifying data exchanged during communications.
NAP-011	3.4	4	The certificate of the app is signed with SHA-1, a cryptographic hash function that has been defined as insecure since February 2017. Given this problem, there could be the possibility of an attacker signing the app by impersonating the same developer.
NAP-016	0	-	The app lacks sufficiently effective checks on the compromise status of the device.

4. EVIDENCES

VULN ID: NAP-007

MASVS-PLATFORM-1



Score 9.9

Critical vulnerability

Vector string

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

Confidentiality

AT RISK

Integrity

AT RISK

Availability

AT RISK

Broken access control

There is no proper authorization control for access to resources. Some APIs only check the validity of the session to allow reading and writing of confidential data of other users. A registered app user could, in fact, query a REST API with a database of emails obtained from third parties, receive in response, the codes of customers registered with those emails that are actually in the database, and then query a second API to obtain all the information of the users whose codes they have obtained.

This issue could create a potentially massive data breach that would almost certainly lead to the intervention of the Privacy Authority and the imposition of a fine under GDPR regulations.

Request: GET https://domain.tld/app/1.0/getAnagrafica?comp=1&app=APP01&v=2.0.0&os=android&cliente=C999&codFIscale=AAAAAA00A00A000A

Response: [...], "cognome": "****", [...], "email": "****", [...], "nome": "****", "localita": "CARMAGNOLA", "via": "VIA ****", "numero_cellulare": "****", [...], "provincia": "TO", [...], "comune_nascita": "****", [...], "provincia_nascita": "TO", "sesso": "M"}

Possible remediation

Return only the data belonging to the user associated with the session authorization token. Identify the user based on the same token, not a query/API parameter.

STRICTLY CONFIDENTIAL

VULN ID: NAP-008

MASVS-NETWORK-1



Score 6.5

Medium vulnerability

Vector string

CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:L

Confidentiality

AT RISK

Integrity

AT RISK

Availability

AT RISK

Lack of certificate pinning

The app has a security check that ensures the presence of a valid certificate on the server it communicates with, which prevents communication through self-signed proxies. However, if the server has a certificate issued by a trusted Certificate Authority (CA), it is seen as secure, which can allow Man-in-the-Middle (MITM) attacks to read any client-server network communication.

Domain: domain.tld

Possible remediation

Implement certificate pinning on the app (e.g., by introducing the server's public key into the code) allowing only Point-To-Point (P2P) communications between the device and the server, discarding any communication involving another device in between.

STRICTLY CONFIDENTIAL

VULN ID: NAP-011

MASVS-CRYPTO-1



Score 3.4	Low vulnerability				
Vector string	CVSS:3.1/AV:A/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:N				
Confidentiality	AT RISK	Integrity	AT RISK	Availability	OK

SHA-1 signed the app certificate

As announced by Google, in collaboration with CWI of Amsterdam, in February 2017 an internal failure of the SHA-1 function was discovered and a hash collision attack was carried out, whereby it was confirmed that two different pieces of data could generate the same digest. The fact that the app is signed with SHA-1 enables an attacker to make changes to publish a malicious app, impersonating a certified developer.

Algorithm: [SHA1withRSA]

Signature: -----

Possible remediation

To mitigate this vulnerability, it is advisable to renew the certificate by changing the signature algorithm to a more secure one, such as SHA-256. Alternatively, for platforms like iOS, one can rely on the code-signing services provided by the ecosystem (e.g., Apple App Store), which utilize modern hashing algorithms to ensure the security of distributed applications.

STRICTLY CONFIDENTIAL

VULN ID: NAP-016

MASVS-RESILIENCE-1



Score 0

No vulnerability

MASTG-TEST-0088 - Lack of jailbreak detection

The application does not perform device compromise checks. In the event of a positive result, it should display a risk alert to the user regarding running applications on a device that allows the execution of code with root permissions. This is because data and information exfiltration is more likely to happen, compared to a normal device with the default permission management.

Moreover, detecting these devices facilitates a more straightforward investigation in case of an incident.

Possible remediation

Implement a jailbreak detection system to identify and alert users about the risks associated with using a compromised device.

STRICTLY CONFIDENTIAL

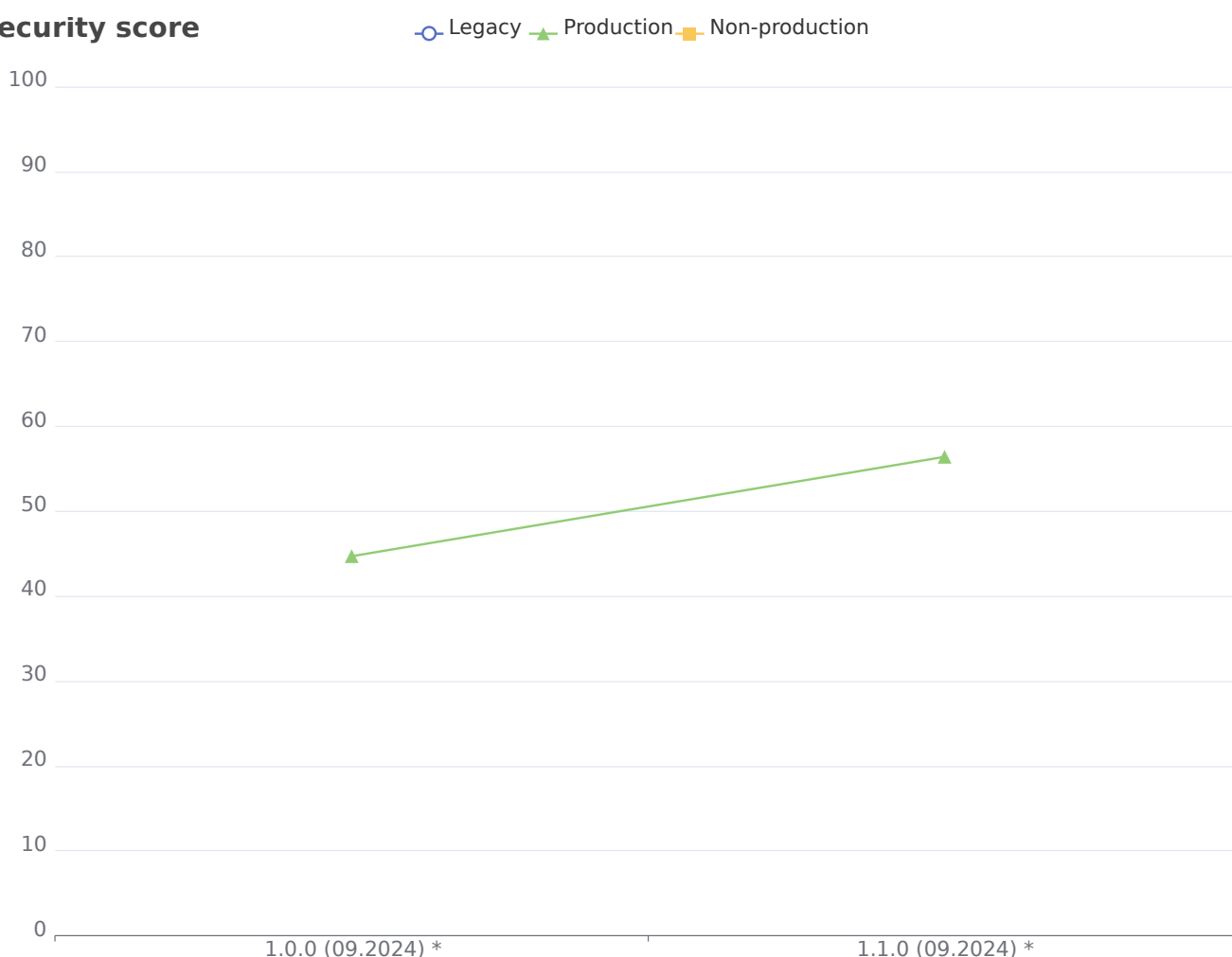
5. SECURITY SCORE

We have also put together a single security index that sums up all the vulnerabilities and security issues in the report. This application security index is a score between 0 and 100, where the higher the value, the better the security.

The difference between the *legacy* and newer scores is down to more testing and the way the results are broken down, which has changed the scale. It would be best to treat the previous *legacy* score as almost equal to the first point in the new data set. In any case, it is probably not a good idea to compare the *legacy* and newer data sets.

Security score: 56.4 /100

Security score



STRICTLY CONFIDENTIAL



THANK YOU

Mobisec Italia srl

Viale della Repubblica, 22
31020 Villorba (TV) – Italia

email: info@mobisec.com

P.Iva e C.F.
04735010268