Finders Keepers Mobile Application Vulnerability Assessment & Penetration Testing Findings Report

Business Confidential

Date: November 27th, 2024

Version 1.0

Table of Content

Table of Contents	2
Confidentiality statement	3
Disclaimer	3
Contact information	3
Assessment Overview	4
Assessment components	4
Application penetration test	4
Finding severity Ratings	5
Risk Factors	5
Likelihood	5
Impact	5
Scope	6
Scope exclusions	6
Client allowances	6
Executive summary	7
Testing summary	7
Tester Notes and Recommendations	7
Vulnerability Summary and Report Card	8
Application penetration test findings	8
Technical findings	9
API App Penetration Test Findings	9
Mobile App Penetration Test Findings	35

Confidentiality Statement

This document is the exclusive property of Finders Keepers. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form requires the consent of Finders Keepers.

Finders Keepers share this document with auditors under non-disclosure agreements to demonstrate compliance with penetration test requirements.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment, not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a complete evaluation of all security controls. The testers prioritized the assessment to identify the weakest security controls an attacker would exploit. The testers recommend conducting similar assessments on a biannual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

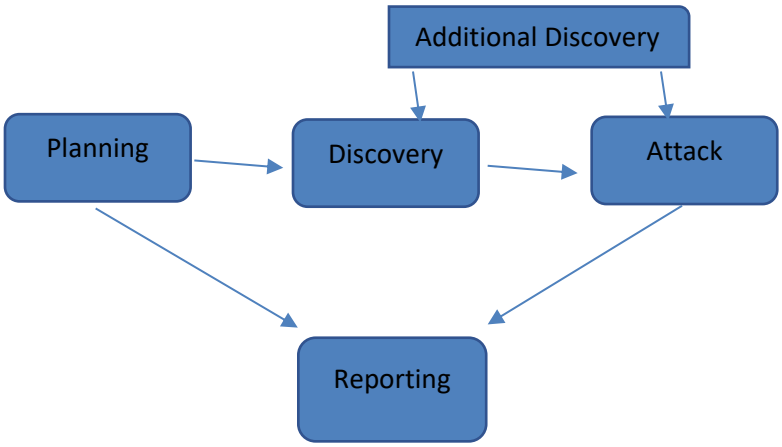
Name	Title	Contact Information (Email)
Oral4		
• Fred	Head, Backend Engineering	fred@123.design
Penetration Testers		
• Dominion • Daniel	Tester	dominionagonor@gmail.com

Assessment Overview

From Novemberr 25th to November 28th, 2024, Finders Keepers engaged the tester to evaluate the security posture of their applications compared to current industry best practices. All testing was based on the NIST SP800-115 technical guide to information security testing and assessment, the OWASP testing guide(v4), and customised testing frameworks.

Phases of penetration testing activities include the following:

- Planning: Customer goals are gathered, and rules of engagement are obtained.
- Discovery: Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack: Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting: Document all found vulnerabilities and exploits, failed attempts, and Finders Keepers strengths and weaknesses.



Assessment Components

Application Penetration Test

An application penetration test emulates the role of an attacker who attempts to breach the scope of the applications. An engineer will enumerate and scan the applications to identify potential vulnerabilities and perform common and advanced attacks, such as cross-site scripting, SQL injection, man-in-the-middle attacks, and more. The Engineer will seek to gain access to the applications through brute forcing, credential stuffing, and exfiltrating sensitive data.

Finding Severity Ratings

The following table defines severity levels and the corresponding CVSS score range, which are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. Forming a plan of action and patching it immediately is advised.
High	7.0-8.9	Exploitation is more complex but could cause elevated privileges and potentially data loss or downtime. Forming a plan of action and patching as soon as possible is advised.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. Forming a plan of action and patch after resolving high-priority issues is advised.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organisation's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information regarding items noticed during testing, robust controls, and other documentation is provided.

Risk Factors

Risk is measured by two factors: Likelihood and impact.

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, the attacker's skill level, and the client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, availability of client systems and/ or data, reputational harm, and financial loss.

Scope

Assessment	Details
API Penetration Test	Finders Keepers API Collections, which includes 95 (95) Subcollections such as Login, Device ,password,. These collections encompass various sub-collections and endpoints for functions like user login, admin tasks, file storage, totaling 95 endpoints across all collections.
Mobile Application Penetration Test	

Scope Exclusions

The tester did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering (specific to the scoped applications for the application test alone)
- Finders Keepers permitted all other attacks not specified above.

Client Allowances

Finders keepers provided the tester with the following allowance(s):

- Finders keepers Mobile Application
- Finders keepers API collection

Engagement Drawbacks

Some of the drawbacks encountered by the testers include:

- Unavailability of finders keepers APK on time at the time of testing
- The application was on a test environment.

Executive Summary

The tester evaluated Finders keepers application security posture through penetration testing from November 25th, 2024, to November 28th, 2024. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Testing Summary

The assessment evaluated the Company's application security posture (for the applications in scope). From an external perspective, the tester performed enumeration and vulnerability scanning against all Mobile application and API access point, provided by Finders keepers to evaluate overall exposure. The team also performed common authentication-based attacks, such as brute-forcing, credential stuffing, SQL injection, etc. Beyond vulnerability scanning and authentication attacks, the tester evaluated other potential risks, such as IDOR, XSS, HTML, injection, etc., consistent with the OWASP testing methodology.

For further information on findings, please review the ***Technical Findings*** section.

Tester Notes and Recommendations

Testing results of Company's applications revealed several critical and high severity rated vulnerabilities that could compromise the organisation's applications and customer accounts/data.

We recommend that Finders keepers evaluates its application deployment process to ensure that security is a part of it. Dynamic and static scanning, including penetration tests for business logic vulnerabilities, should be prioritised before applications are deployed to production.

Limitations

During our testing, we encountered an issue where we were unable to register on the API, which significantly limited our ability to create test accounts. This restriction prevented us from fully evaluating the functionality and security of account registration and related features.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and remediation status:

API Vulnerability Assessment and Penetration Test Findings

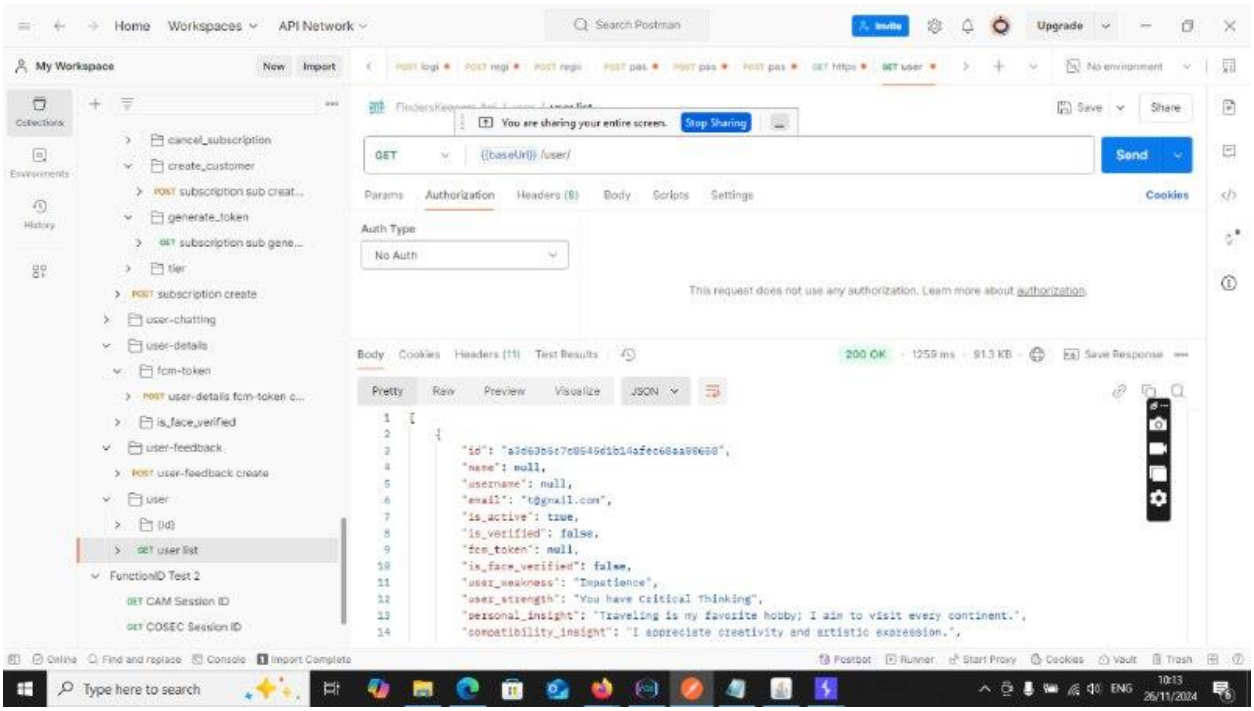
4	2	0	0	0
Critical	High	Medium	Low	Informational

Finding	Severity	Remediation Status
Penetration Static Analysis Test		
MPT-001: Broken Authorization	Critical	*Not Remediated*
MPT-002: Unauthorized Access to Customer Sensitive Data	Critical	*Not Remediated*
MPT-003: Broken Password Reset Logic	Critical	*Not Remediated*
MPT-004: Rate limiting on Password Field	Critical	*Not Remediated*
MPT-005: No Password Policy	High	*Not Remediated*
MPT-006: HSTS errors	Medium	*Not Remediated*

Technical Findings

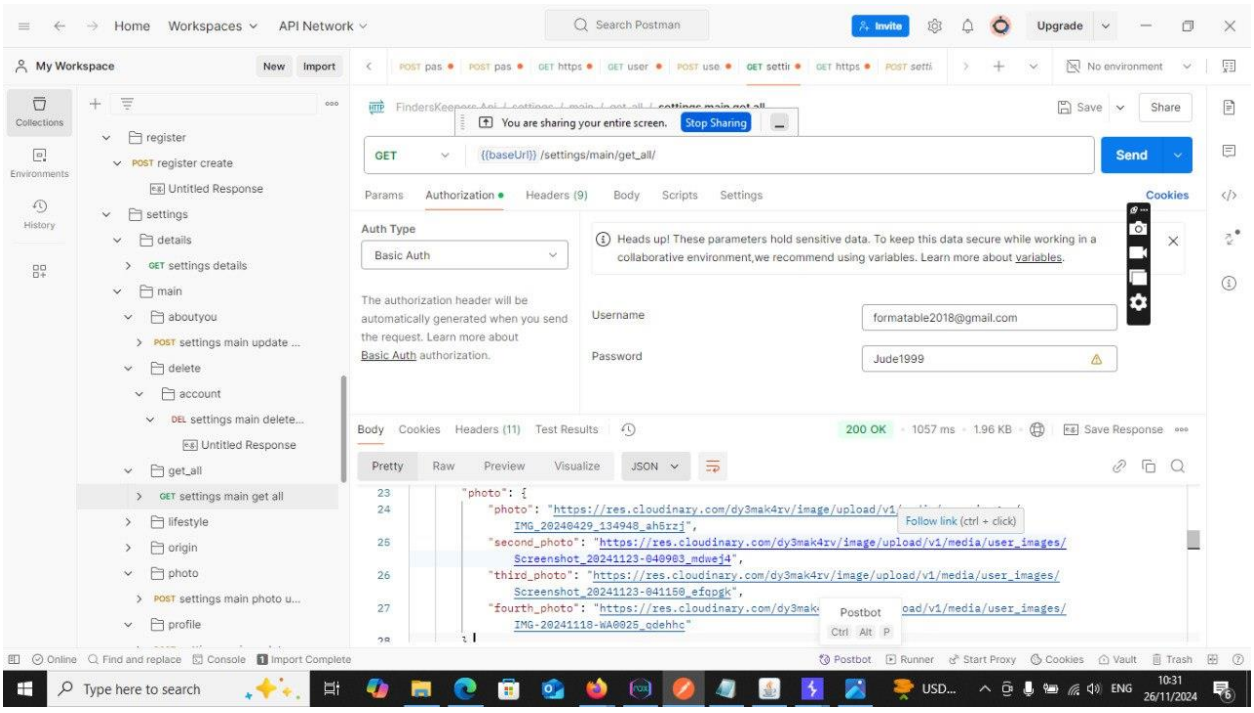
API Vulnerability Assessment and Penetration Test Findings.

MPT-001: Broken authorization	
description of Finding	Our assessment revealed that the Finders keepers mobile application's is vulnerable to broken authorization, which is a critical vulnerability. Specifically, we were able to view users details which includes id, fcm-token, email, name and personal information added by the user without any authentication or authorization. This was performed by calling the endpoint `/user/details` or `/user/{id} without supplying any credentials and the endpoint responded with the user data.
Affected Asset:	backend.finderskeepers/user/
Severity:	***Critical***
Risk:	The risk exists that the exposure of sensitive information violates data protection regulations such as GDPR, CCPA, or other relevant laws. Non-compliance can lead to substantial fines and legal actions against the organisation. The breach can significantly damage the organisation's reputation and erode trust with users. This also violates the confidentiality and integrity of information assets.
Recommendation:	We recommend implementing the following measures: ▪ Ensure that all API endpoints on the Finders Keepers application containing sensitive information require proper authentication before they can be accessed. This includes implementing login mechanisms and access controls. ▪ Apply strict authorization checks to ensure that users can only access documents and data relevant to their role or permissions.
Remediation Status:	*Not Remediated*

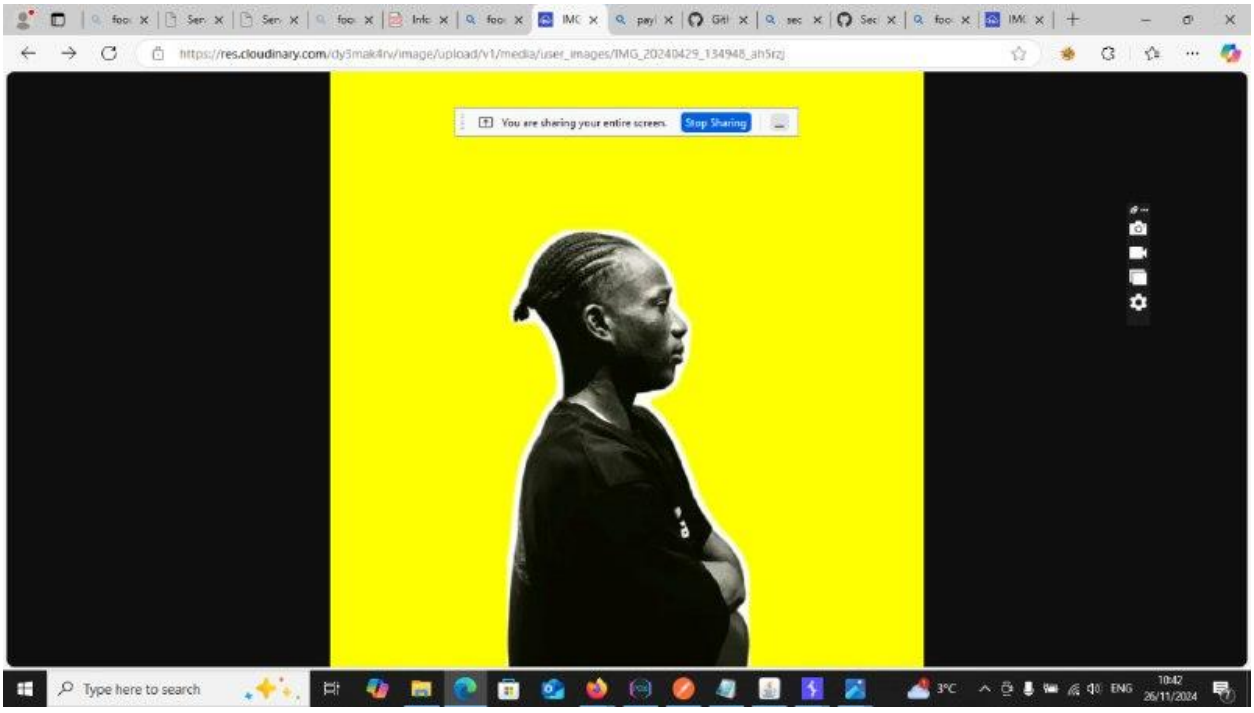


the image shows customer details being accessed

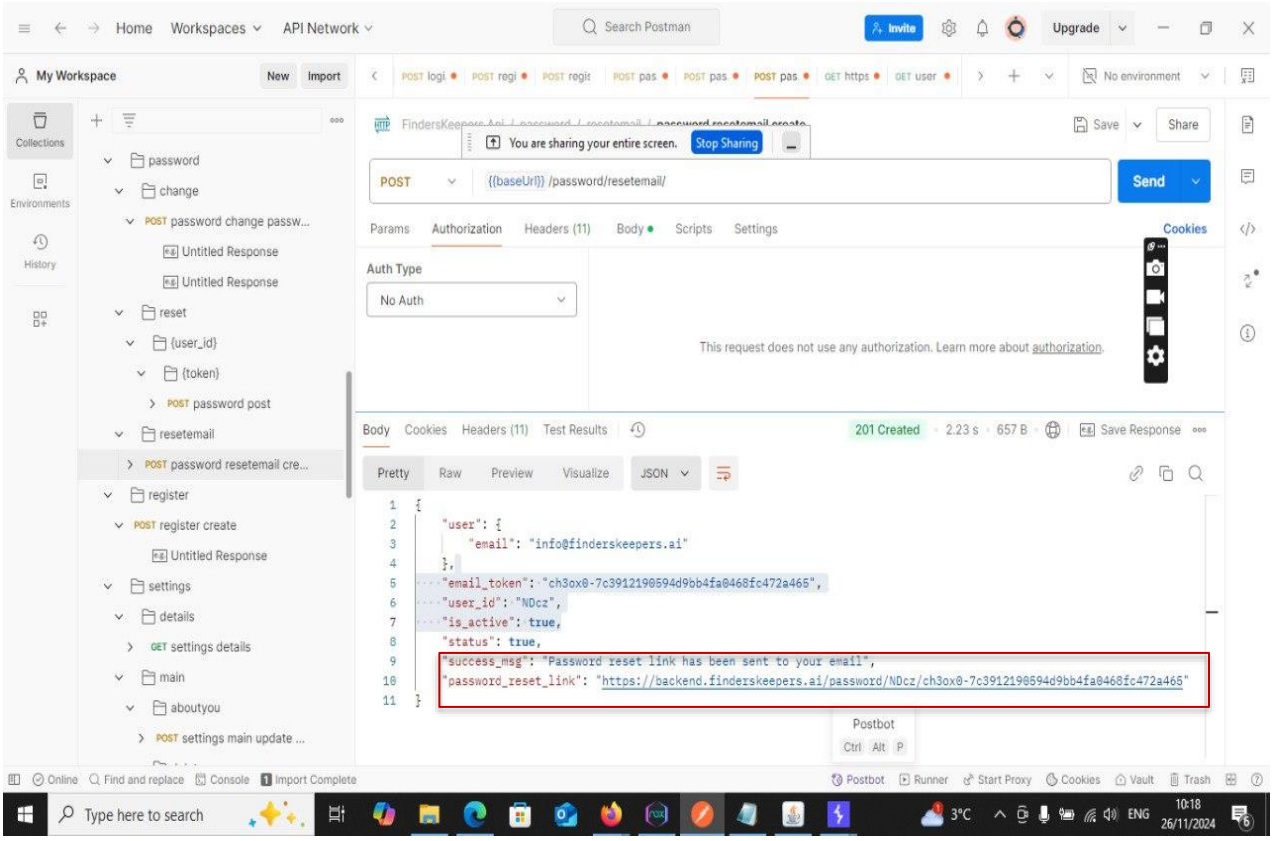
MPT-002: Unauthorized Access to Customer Sensitive Data	
description of Finding	Our assessment revealed that the Finders Keepers mobile application exposes customer pictures to unauthorized users due to insufficient access control mechanisms on one of its APIs. By intercepting and interacting with the API endpoint responsible for retrieving customer data, we were able to identify that pictures of customers are stored in a publicly accessible url that doesn't require authentication to access. This indicates that the application does not adequately store users data efficiently.
Affected Asset:	backend.finderskeepers/setting/main/get_all/
Severity:	***Critical***
Risk:	This vulnerability poses significant risks to the application, its users, and its reputation, including: - **Privacy Breach:** Exposing customer pictures compromises user privacy and may lead to unauthorized use of personal data. - **Reputation Damage:** Public exposure of this issue could erode user trust in the application and the brand. - **Potential for Abuse:** Malicious actors could exploit the exposed pictures for identity theft, phishing campaigns, or other harmful activities. - **Regulatory Non-Compliance:** This issue could result in violations of privacy laws such as GDPR, CCPA, or NDPR, potentially leading to legal action and financial penalties.
Recommendation:	To address this vulnerability and prevent unauthorized access to sensitive data, the following steps should be implemented: • **Implement Proper Access Controls:** Ensure that all sensitive customer data, including pictures, is protected by strong access control mechanisms. Access should be restricted to authorized users only, and the application should validate that the user requesting the data has the appropriate permissions. • **Store Customer Data Securely:** Store customer pictures and other sensitive information in secure locations with proper encryption both at rest and in transit. Avoid storing such data in publicly accessible URLs or directories
Remediation Status:	*Not Remediated*



this picture shows user photo links and by linking each link it gives you access to the picture submitted by the user.

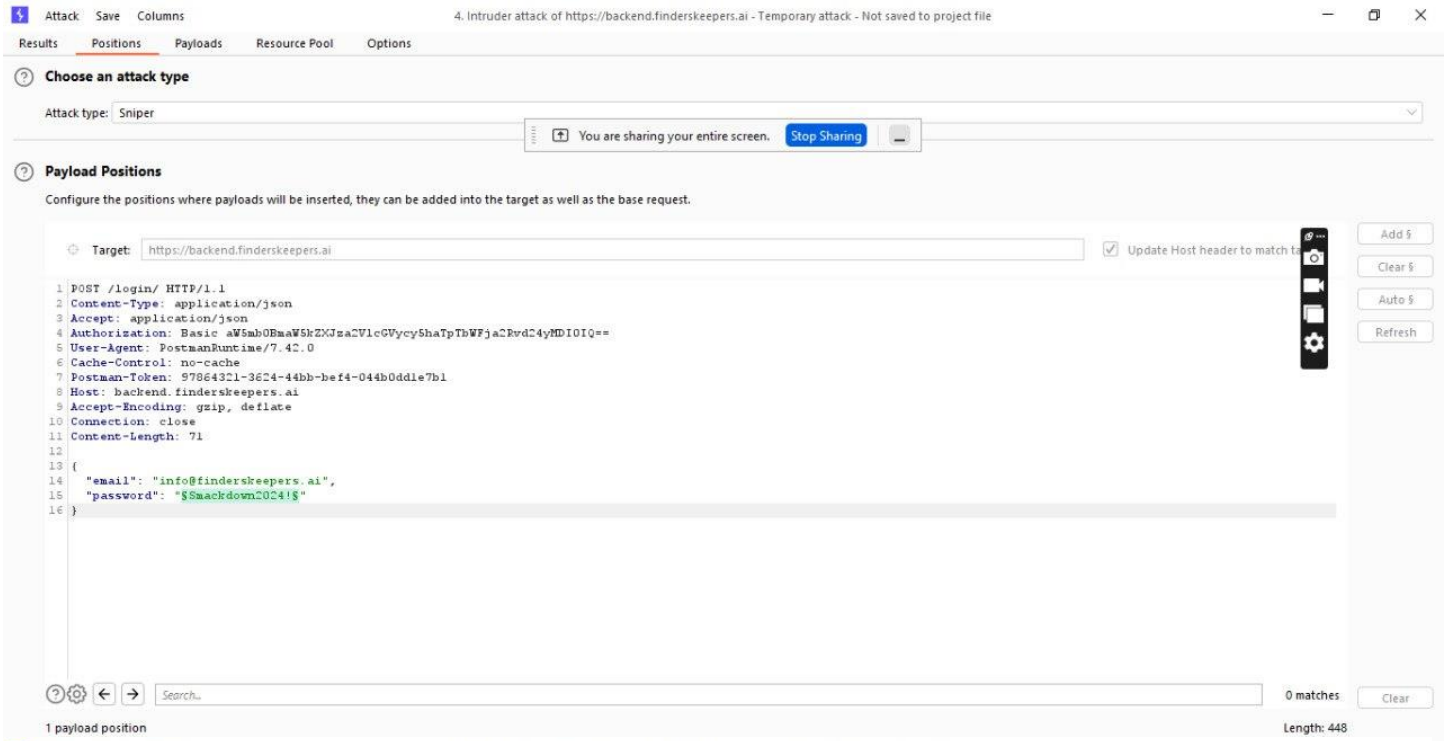


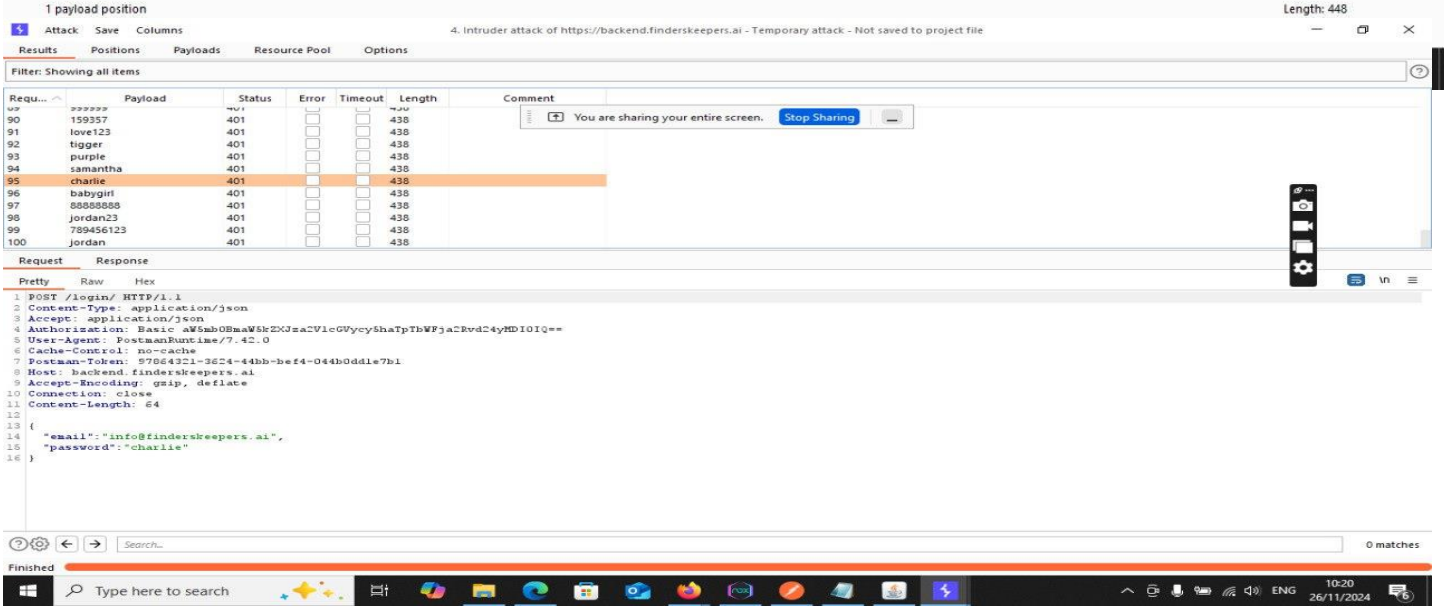
MPT-003: Broken Password Reset Logic	
description of Finding	Our assessment identified a vulnerability in the Finders Keepers mobile application's password reset logic. Specifically, we found that the password/reset_mail endpoint requires users to enter their email address, after which a reset link is sent to the provided email. However, we discovered that the reset link is included directly in the endpoint's response. This means that by intercepting the server response within the mobile application, an attacker could potentially reset a user's password without their knowledge.
Affected Asset:	backend.finderskeepers/setting/main/get_all/
Severity:	***Critical***
Risk:	This vulnerability poses a significant security risk, as it allows an attacker to potentially reset a user's password without their knowledge or consent. By intercepting the response from the password reset endpoint, an attacker could gain unauthorized access to a user's account, compromising sensitive information and functionality. This could lead to unauthorized account access, identity theft, data loss, or further exploitation of the compromised account. Given that this issue bypasses the need for user interaction, it is critical to address it to prevent unauthorized access to user accounts and protect the integrity of the application.
Recommendation:	• **Do Not Include Reset Links in API Responses**: The password reset link should not be included directly in the response from the password/reset_mail endpoint. Instead, the application should send the reset link via a secure email to the user, ensuring it cannot be intercepted through the server response. • **Enhance Authentication for Sensitive Operations**: Implement additional security measures such as multi-factor authentication (MFA) for password reset processes, ensuring that even if an attacker intercepts the reset link, they cannot complete the reset without further verification.
Remediation Status:	*Not Remediated*

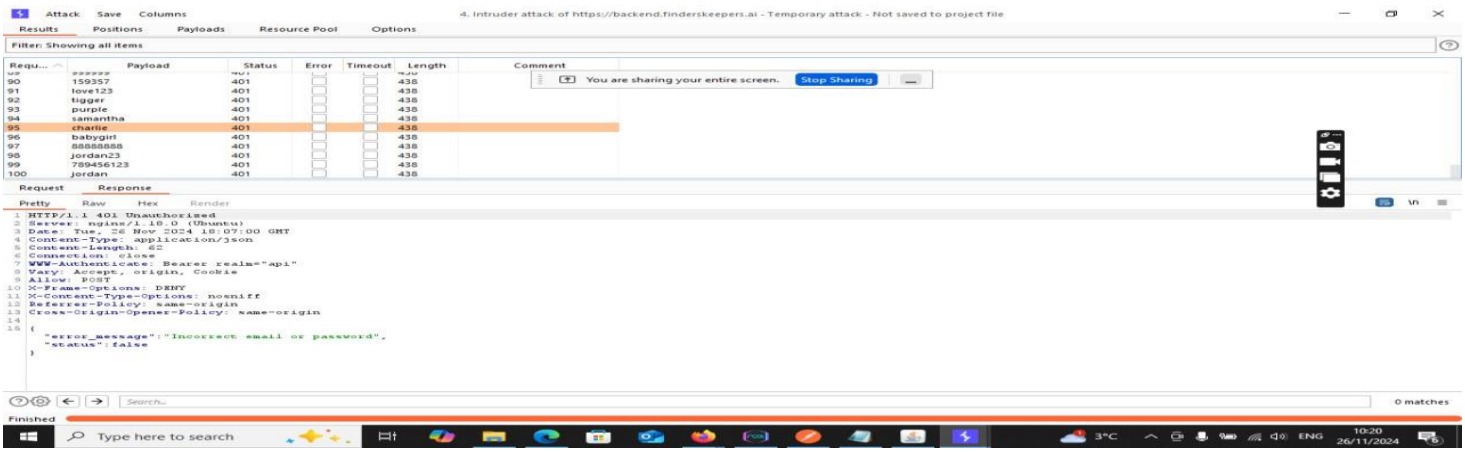


The reset link is shown in the response and can be stolen by attackers to change a user password

MPT-001: Rate limiting on password field	
Description of Finding	Our observation revealed that the Finders keepers mobile application does not implement adequate rate-limiting mechanisms on the password field. This absence of rate limiting allows repeated login attempts without restriction, significantly increasing the risk of brute force or credential-stuffing attacks. The repeated attempts enabled by this vulnerability could lead to unauthorized access to user accounts, exposing sensitive information or compromising user data. An attacker could leverage automated tools to exploit this weakness and bypass authentication mechanisms. To mitigate this risk, the application should enforce rate-limiting controls on the password field by restricting the number of failed login attempts allowed within a specific time frame. *Rate limiting is an essential mechanism to reduce the risk of unauthorized access through automated or malicious login attempts, necessitating immediate implementation within the application*
Affected Asset:	hhtps://backend.finderskeepers.ai
Severity:	***Critical***
Risk:	Without proper rate limiting, the password field becomes vulnerable to automated attacks that can compromise user accounts. An attacker can exploit this weakness to gain unauthorized access, potentially leading to data breaches, account hijacking, or the exposure of sensitive user information.
Recommendation:	Management should: • **Restricting Repeated Attempts:** Set a threshold for the number of failed login attempts within a given timeframe. • **Temporary Account Lockouts:** Temporarily lock the account after exceeding the allowed attempts. • **Error Message Uniformity**: Avoid revealing whether a username or password is incorrect, as this can aid attackers. • Additionally, the implementation of multi-factor authentication (MFA) can further enhance account security by adding another layer of protection beyond the password.
Remediation Status:	*Not Remediated*



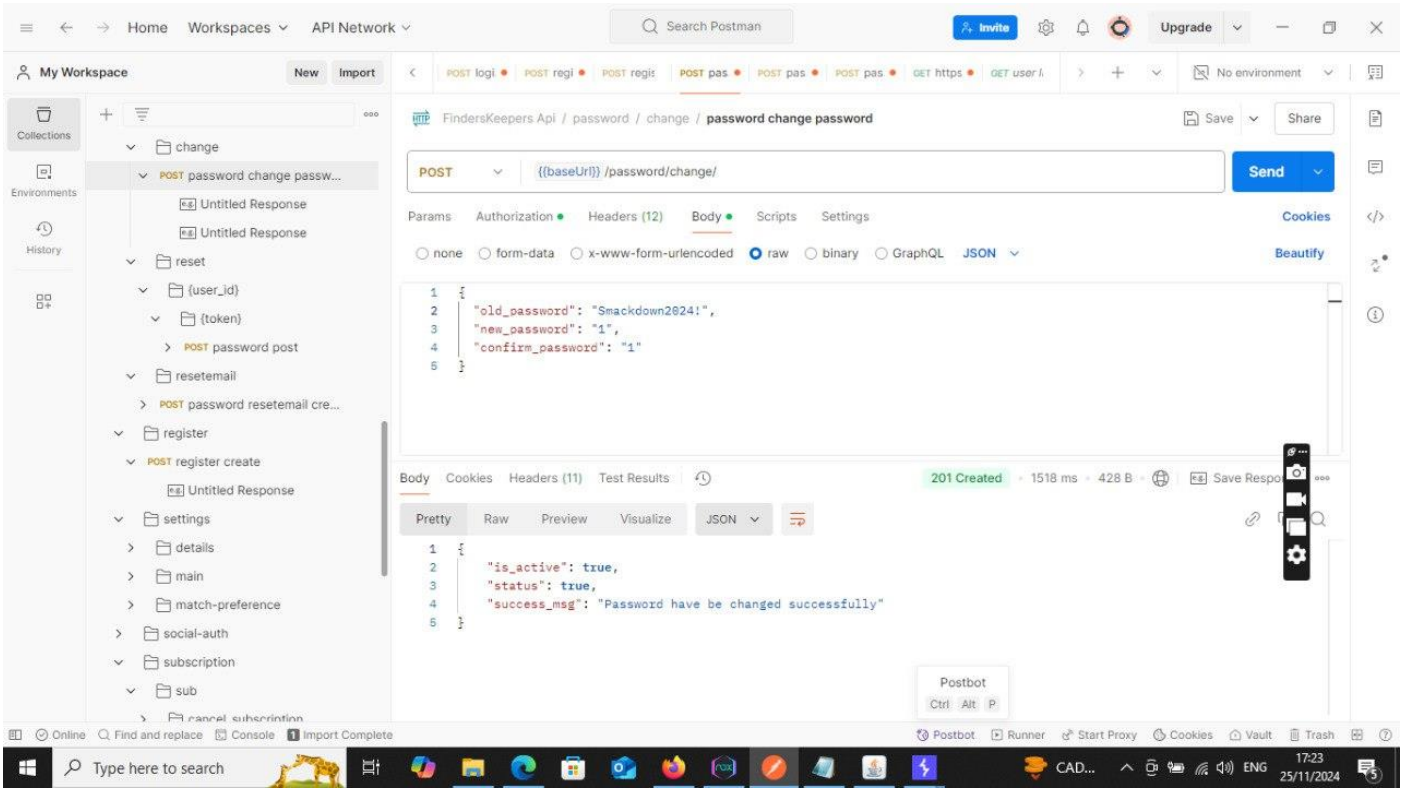




The image shows that 100 different possible passwords were tries.

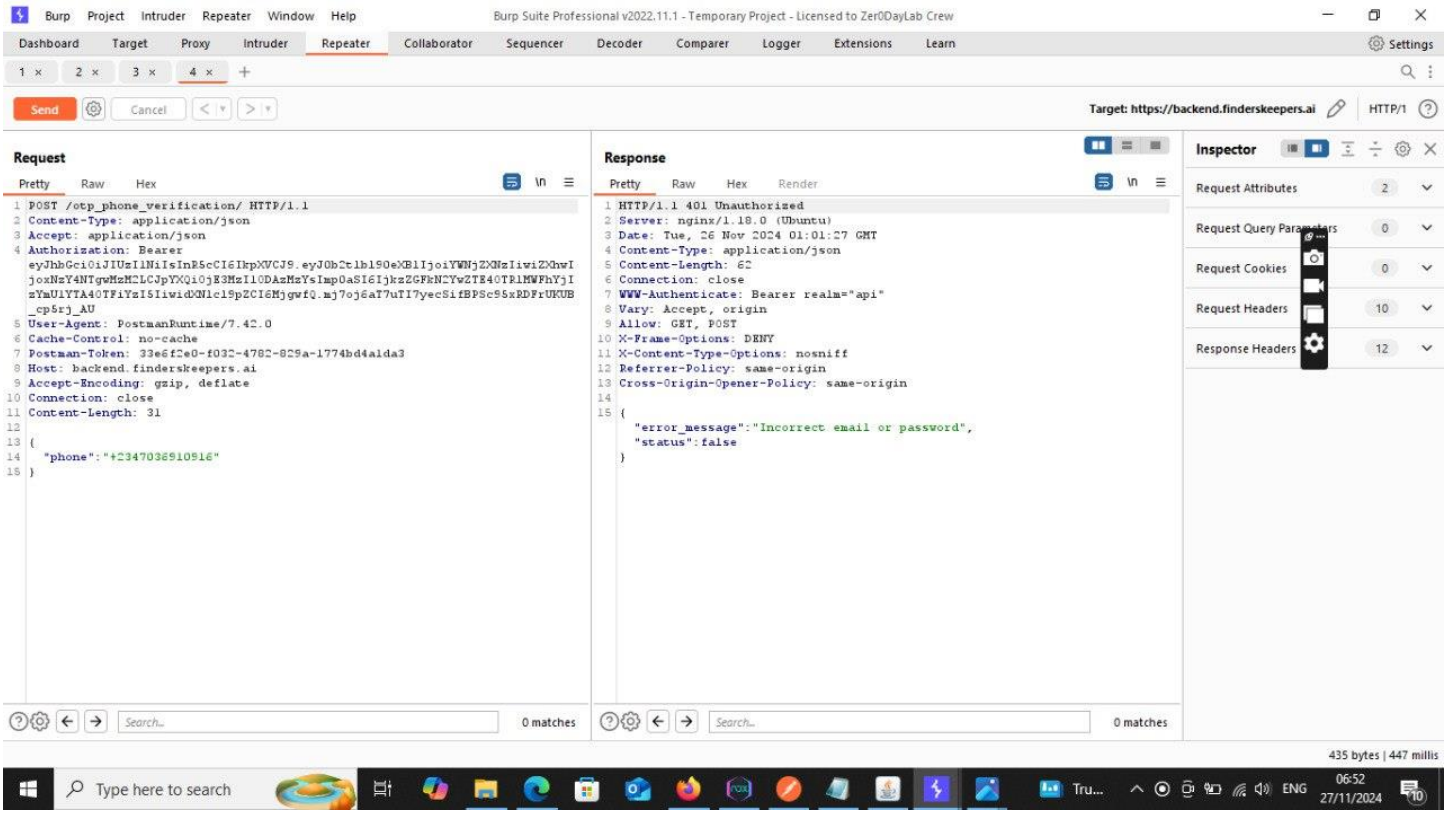
MPT-005: No Password Policy	
Description of Finding	Our assessment revealed that the Finders Keepers mobile application does not enforce a password polic during account creation or password updates. Users can set weak, short, or commonly used passwords (e "123456", "password",”1”), significantly increasing the likelihood of account compromise through brute for credential stuffing, or dictionary attacks.
Affected Asset:	backend.finderskeepers/password/change
Severity:	***HIGH***
Risk:	The lack of a password policy introduces several critical risks, including: - **Increased Vulnerability to Attacks:** Weak or common passwords can be easily guessed or cracked u automated tools. - **Account Takeover:** Compromised accounts can be exploited for fraud, data breaches, or other malicio activities. - **Regulatory Non-Compliance**: Failure to implement basic password security controls may violate data protection standards like GDPR, CCPA, or PCI-DSS. - **Loss of User Trust:** Users may lose confidence in the application’s ability to safeguard their informatio
Recommendation:	To address this issue, the application should implement and enforce a strong password policy with the fo requirements: 1. **Password Strength Requirements:** - Minimum length of 8–12 characters. - Include a combination of uppercase and lowercase letters, numbers, and special characters. - Prohibit commonly used passwords (e.g., "123456", "password"). - Use a blocklist to prevent passwords exposed in known data breaches. 2. **Validation and Feedback:** - Provide real-time feedback on password strength during password creation. - Ensure that all password validations are performed server-side to prevent bypasses.

	3.**Multi-Factor Authentication (MFA):** - Complement the password policy by implementing MFA to provide an additional layer of security. Additional Safeguards: - Regularly audit the password policy to ensure compliance with evolving security standards. - Perform periodic penetration testing to identify potential weaknesses in the authentication workflow.
Remediation Status:	*Not Remediated*



We set the password as “1” and it was changed.

MPT-006: Missing Security Headers	
Description of Finding	We noted that several important security headers are missing from the HTTP responses of the oral4 web application. These headers are crucial for enhancing the security posture of web application providing addi layers of protection against common web vulnerabilities. Specifically, we identified that the following security headers are not present in the HTTP response • X-Frame-Options: Prevents the page from being displayed in an iframe, protecting again clickjacking att • X-XSS-Protection: Enables the browser's built -in filter against reflected XSS attacks. • X-Content-Type-Options: Prevents MIME type sniffing, which can lead to security vulnerabilities. • Strict-Transport-Security (HSTS): Tells browsers to only access the site using HTTPS, preventing down attacks. • Content-Security-Policy (CSP): Specifies which dynamic resources are allowed to load, mitigating XSS a data injection attacks
Affected Asset:	backend.finderskeepers/password/change
Severity:	***MEDIUM***
Risk:	The risk exists that the affected applications may be vulnerable to various web -based attacks due to the al of critical security headers. Without these protective measures, the application cou susceptible to clickjack cross -ite scripting (XS), MIME type sniffing, and man -in-the-middle attacks. This vulnerability potentially e user data, compromises the integrity of the platform, and lead to unauthorised access or content manipula ultimately risking the security of both use the organization.
Recommendation:	Management should ensure the following security headers are implemented in all HTTP responses: • X-Frame-Options: Set to DENY or SAMEORIGIN to prevent clickjacking • X-XSS-Protection: Set to "1; mode=block" to enable browser's XSS protection. • X-Content-Type-Options: Set to "nosniff" to prevent MIME type sniffing • Strict-Transport-Security: Implement with appropriate max-age to enforce HTTPS • Content-Security-Policy: Develop and implement a suitable policy to prevent XSS and other injection att
Remediation Status:	*Not Remediated*


Burp Project Intruder Repeater Window Help
Burp Suite Professional v2022.11.1 - Temporary Project - Licensed to Zer0DayLab Crew
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Settings
1 × 2 × 3 × 4 ×
Send Cancel
Target: https://backend.finderskeepers.ai
HTTP/1
Request
Pretty Raw Hex
1 POST /otp_phone_verification/ HTTP/1.1
2 Content-Type: application/json
3 Accept: application/json
4 Authorization: Bearer
eyJhbGci0iJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwI
joxNzY4NTgwMzM2LCJpYXQi0jE3MzI1ODAzMzYsImp0aSI6IjkzZGFkNzYwZTE40TRlMWFhYjI
zYmU1YTA40TFiYzI5IiwidXNlcl9pZCI6MjgwfQ.mj7oj6aT7uTI7yecSifBPSc95xRDFrUKUB
_cp5rj_AU
5 User-Agent: PostmanRuntime/7.42.0
6 Cache-Control: no-cache
7 Postman-Token: 33e6f2e0-f032-4782-8295a-1774bd4alda3
8 Host: backend.finderskeepers.ai
9 Accept-Encoding: gzip, deflate
10 Connection: close
11 Content-Length: 31
12
13 {
14 "phone":"+2347036910916"
15 }
Search...
0 matches
Response
Pretty Raw Hex Render
1 HTTP/1.1 401 Unauthorized
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Tue, 26 Nov 2024 01:01:27 GMT
4 Content-Type: application/json
5 Content-Length: 62
6 Connection: close
7 WWW-Authenticate: Bearer realm="api"
8 Vary: Accept, origin
9 Allow: GET, POST
10 X-Frame-Options: DENY
11 X-Content-Type-Options: nosniff
12 Referrer-Policy: same-origin
13 Cross-Origin-Opener-Policy: same-origin
14
15 {
"error_message":"Incorrect email or password",
"status":false
}
Search...
0 matches
Inspector
Request Attributes 2
Request Query Parameters 0
Request Cookies 0
Request Headers 10
Response Headers 12
435 bytes | 447 millis
Type here to search
ENG
06:52
27/11/2024