

Executive report

maekuss/DamnVulnerableRemixApp

October 2026

Confidential. Prepared for Hacktron.

HACKTRON

Contents

1. Engagement details 3
2. Executive summary 3
3. Scope 4
4. Methodology 4
5. Severity rubric 5
6. Overall risk rating 5
7. Findings summary 6
8. Key findings 6
9. Conclusion 7
10. Limitations and disclaimer 7

1. Engagement details

Client	Hacktron
Engagement	maekuss/DamnVulnerableRemixApp
Testing window	19 October to 23 October 2026
Report issued	27 October 2026
Prepared by	Hacktron AI
Classification	Confidential

2. Executive summary

Hacktron assessed Hacktron's maekuss/DamnVulnerableRemixApp repository on the main branch using source-code analysis, local testing and validation against the deployed application. The engagement identified 8 findings: 1 Critical, 5 High, 2 Medium.

The assessment identified eight distinct issues in account recovery, role enforcement, expense authorization and receipt handling. The most serious finding permits unauthenticated takeover of an administrator account. Additional paths allow employees to acquire administrator privileges, read or delete other users' expenses, and access broader exports.

Dynamic validation confirmed six findings and partially confirmed two. SSRF validation established a controlled internal callback; receipt-import validation established acceptance and storage of an unvalidated file type. Cloud credential retrieval and server-side code execution were not established.

Engagement type	White-box penetration test
Methodology	Source-code analysis, local reproduction and testing against the deployed application
Code in scope	https://github.com/maekuss/DamnVulnerableRemixApp
Deployed target	https://expenseflow.demo.example

3. Scope

Testing was limited to the repositories and deployed targets listed below.

Repository	maekuss/DamnVulnerableRemixApp
Branch	main
URL	https://github.com/maekuss/DamnVulnerableRemixApp
Deployed target	https://expenseflow.demo.example

Testing	Authenticated
Access obtained	Customer-provided browser access instructions. Credentials are not included.
Accounts used	Employee A (Alex), Employee B (Blair), Sales manager, Administrator
Access check	Verified on 19 October 2026, before testing started
Assessment	19 October to 23 October 2026

4. Methodology

Hacktron conducted a white-box penetration test using source-code analysis, local testing and validation against the deployed application.

Static analysis

1. Map applications, entry points and sensitive operations.
2. Trace attacker-controlled input through the code and review access controls, data handling and business logic.
3. Check exploit paths, preconditions and existing controls.
4. Validate findings, consolidate duplicates and assign severity and remediation recommendations.

Local testing

Local testing used the reviewed code and focused reproduction scripts to investigate findings, check exploit conditions and develop proof-of-concept steps.

Dynamic testing

Source review and local reproduction guided tests of selected attack paths against the application listed in Scope. Hacktron used proof-of-concept requests and application workflows to test whether the suspected vulnerabilities could be reproduced.

Application responses and observable behaviour were examined to determine whether each tested issue was reproduced and what impact was demonstrated. The findings record the outcome, supporting evidence and any prerequisites needed for reproduction.

Authenticated testing used the accounts and roles listed in Scope.

Validation and evidence	Code analysis and local testing: Code review and local reproduction investigate the vulnerable path and exploit prerequisites. Dynamic testing: Live Validation records the observed result, supporting evidence and impact demonstrated.

5. Severity rubric

Severity reflects impact on the application and the conditions needed for exploitation. These ratings are Hacktron's own and are not derived from CVSS. External controls and business context unavailable during testing are excluded.

Severity	Definition	Fix window
Critical ■	Compromises the application, its data, or its users directly, with no meaningful precondition beyond reaching the entry point.	Within 7 days
High ■	Compromises data or an access control on a path that requires an account, a role, or a plausible precondition.	Within 30 days
Medium ■	Requires user interaction, a specific configuration, or a narrower set of conditions, or the impact is limited in scope.	Within 90 days
Low ■	Limited impact on its own. Usually needs to be chained with another issue to cause harm.	Next release cycle
Info ■	No direct security impact. A hardening opportunity or an observation recorded for completeness.	Best effort

6. Overall risk rating

Critical

The overall Critical rating is driven by the unauthenticated account-takeover finding HCK-001. The assessment also identified five High and two Medium findings. Independent privilege-escalation paths and cross-user expense operations increase the scope of potential compromise.

Prioritize account recovery and server-side authorization first. The internal-fetch and file-import findings retain explicit limits on demonstrated impact; broader exploitation paths are not assumed.

7. Findings summary

Critical	High	Medium	Low	Info
1	5	2	0	0

Recommended fix windows. Critical: Within 7 days. High: Within 30 days. Medium: Within 90 days. Low: Next release cycle.

Vulnerability distribution	Count
Broken access control	4
Account recovery	1
Mass assignment	1
Server-side request forgery	1
Unrestricted file upload	1

8. Key findings

Severity	ID	Title
Critical	HCK-001	Predictable password-reset tokens permit account takeover
High	HCK-002	Profile mass assignment allows an employee to become an administrator
High	HCK-003	Administrative user management lacks role enforcement
High	HCK-004	Expense details and deletion do not enforce record ownership
High	HCK-005	Receipt import can issue server-side requests to internal services
High	HCK-006	Managers can approve expenses outside their department
Medium	HCK-007	Receipt import stores unvalidated file types
Medium	HCK-008	CSV export discloses expenses belonging to other employees

9. Conclusion

Weaknesses in account recovery and server-side authorization can expose administrative access and financial records. Remediation should first remove deterministic recovery tokens, restrict protected profile fields and enforce authorization at every administrative and expense endpoint.

All eight findings remain Open. A follow-up retest should verify the fixes and their negative controls before closure.

Next steps

1. Within 7 days: replace predictable reset tokens and invalidate outstanding tokens.
2. Within 30 days: enforce profile-field allowlists, administrator checks, expense ownership and manager department checks; constrain receipt-fetch destinations.
3. Within 90 days: validate receipt formats and align export access with interactive permissions.
4. Submit fixes with regression evidence, then retest the original request, cross-role controls and relevant variants.

10. Limitations and disclaimer

Testing covers the source code and deployed application during the dates listed in Scope. Later changes were not reviewed.

Results cover the paths and controls examined during testing. Additional vulnerabilities may remain.

Unless listed in Scope, testing excluded transport and TLS configuration, dependency and supply-chain analysis (SCA), secret scanning and credential hygiene, cloud and infrastructure configuration, denial-of-service and load testing, social engineering and physical security, and third-party services outside the repositories listed in Scope.

The application owner is responsible for remediation. Recommended fixes have not been retested.

Prepared for Hacktron. This report is confidential and should be shared only with authorized recipients.