Ronin IP

Offensive Security & Risk Assessment

Reference	RONIN-2026-001
Date	February 2026
Target	mzansi-innovation-hub.co.za
Type	Black-box / Source-assisted

Security Assessment Report

1. Executive Summary

Overall Risk Rating: CRITICAL

Summary: The assessment identified multiple critical and high-severity security weaknesses. These issues collectively enable authentication bypass risk, cross-origin request abuse, credential leakage, and exposure of internal administrative services. No exploit payloads were required to demonstrate compromise potential. Public exposure combined with leaked secrets constitutes demonstrable access risk under standard security assessment criteria.

Impacted Security Goals: Confidentiality, Integrity, Access Control

2. Scope & Methodology

Network Surface

- Open Ports: 80, 443, 7547
- IPs: 3.33.130.190, 15.197.148.33

Source Code

- mih_api_hub (FastAPI backend)
- mih_ui (Flutter frontend)
- Docker infrastructure

3. Detailed Findings

MIH-001

CRITICAL

Public Exposure of Internal Infrastructure Services

Multiple internal services are bound to 0.0.0.0 and mapped directly tohost ports, making them publicly reachable when deployed on an internet-facing host.

EVIDENCE (DOCKER-COMPOSE.YML):

```
MySQL exposed on 3306:3306 MinIO API/Console exposed on 9000/9001 Gitea exposed on 3000 SuperTokens core exposed on 3567 Ollama AI service exposed on 11434 Portainer exposed on 9443
```

RISKS

- Unauthorized administrative access
- Data exfiltration
- Service takeover
- Lateral movement

REMEDIATION

- Remove public port mappings for internal services
- Bind services to localhost (127.0.0.1) only
- Place admin services behind VPN

MIH-002

CRITICAL

Hardcoded SuperTokens API Key in Backend

An administrative SuperTokens API key is hardcoded directly in the backend source code.

EVIDENCE (MIH_API_HUB/MAIN.PY):

```
Line ~58: api_key="leatucczyixqwkqqdrhayiwzeofkltds"
```

RISKS

- Authentication system compromise
- Session forgery
- Account takeover

REMEDIATION

- Rotate the compromised key immediately
- Move secrets to environment variables

MIH-003

CRITICAL

Backend Auth Secret Distributed in Frontend

The same SuperTokens API key used by the backend is embedded in frontend code, distributing it to every client.

EVIDENCE (MIH_AUTHENTICATION_SERVICES.DART):

```
"Authorization": "leatucczyixqwkqqdrhayiwzeofkltds"
```

RISKS

- Public disclosure of admin credentials
- Unrestricted API abuse
- Permanent loss of trust boundary

REMEDIATION

- Remove all backend secrets from frontend code
- Implement proper backend-only auth flows
- Rotate all exposed keys

MIH-004

CRITICAL

CORS Misconfiguration

The API allows all origins while permitting credentials, violating browser security models.

EVIDENCE (MIH_API_HUB/MAIN.PY):

```
allow_origins = ["*"] allow_credentials = True
```

RISKS

- Cross-Site Request Forgery (CSRF)
- Authenticated cross-origin abuse
- Silent data exfiltration

REMEDIATION

- Remove wildcard origin
- Explicitly whitelist trusted domains

MIH-005

HIGH

Unsafe File Upload and Path Construction

User-controlled values are used to construct storage paths without validation, and file types are determined using unsafe filename parsing.

EVIDENCE (MIH_API_HUB/ROUTERS/FILESTORAGE.PY):

```
Issues: User input used directly in object paths. File type inferred from
filename only.
```

RISKS

- Path traversal & Cross-tenant access
- Malicious file uploads (Stored XSS/RCE)

REMEDIATION

- Validate all path components strictly
- Use MIME type detection based on content

MIH-006

HIGH

Containers Running as Root

All application containers run as root, increasing the impact of container compromise.

EVIDENCE (DOCKERFILE):

```
Observation: No USER directive defined in Dockerfiles.
```

RISKS

- Host escape amplification
- Privilege escalation
- Expanded blast radius

REMEDIATION

- Define non-root users in Dockerfiles
- Apply least-privilege container execution

MIH-007

MEDIUM

Missing Web Security Headers

The frontend web server serves files without enforcing standard browser security headers.

EVIDENCE (MIH_WEB_SERVER.PY):

```
Missing: Content-Security-Policy, X-Frame-Options, HSTS, X-Content-Type-Options
```

RISKS

- Increased XSS risk
- Clickjacking
- MIME-type confusion

REMEDIATION

- Add standard HTTP security headers
- Implement a strict CSP

4. Evidence of Access Risk

The combination of publicly exposed administrative services, leaked authentication secrets, and permissive CORS configuration provides sufficient evidence of effective access risk.

- Admin secrets are publicly disclosed
- Authentication boundary is broken by design
- Internal services are reachable from the internet

5. Overall Recommendations

- Immediately rotate all exposed secrets
- Restrict network exposure of internal services
- Fix CORS configuration
- Implement proper secrets management
- Harden container execution
- Add web security headers

Ronin IP
roninip.co.za

Generated: February 04, 2026
Confidential & Proprietary

Disclosure Notice: This report documents security conditions observed during authorized testing. No data was modified, accessed, or exfiltrated.