Vulnerability Assessment Report

for Cloud Infrastructure

Table of Contents

[1. Summary 3](#_Toc94358811)

[2. Methodology 3](#_Toc94358812)

[3. Assessment Results 4](#_Toc94358813)

[4. Severity, Risk-score & Prioritization 4](#_Toc94358814)

[5. Recommendations 4](#_Toc94358815)

# Summary

The purpose of this vulnerability/security assessment is to identify security issues in the AWS infrastructure that hosts a web application. A total of 21 issues were identified. Following tables are a break-down of vulnerabilities based on severity and priority.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Severity of  Vulnerability** | **Count** |  | **Priority** | **Count** |
| Critical | 5 |  | P0 | 6 |
| High | 6 |  | P1 | 3 |
| Medium | 4 |  | P2 | 10 |
| Low | 6 |  | P3 | 2 |

The vulnerabilities found on AWS infrastructure consisted of Misconfigurations, Weak passwords, and Usage of outdated software components.

It is recommended to take proper action based on the suggested remediation.

# Methodology

Methodology implemented for the assessment included both SAST & DAST approaches, details of which are provided below.

**AWS Infrastructure**

* **SAST**

1. Review Terraform template and Ansible Playbook manually.
2. Leverage a SAST tool - Checkov(<https://github.com/bridgecrewio/checkov>) to analyze Terraform template.

* **DAST**

1. Spin-up the infrastructure using Terraform in Test environment. Review the created infrastructure manually using Management Console.
2. Use a CSPM tool(Prowler - <https://github.com/prowler-cloud/prowler>) to audit the infrastructure.

**Web application**

* **SAST –** Leverage a tool (snyk - <https://github.com/snyk/snyk>) to scan the source code of application - *myap*p.

**Note:**

* + 1. Furthur research is needed to weed out false positives from the above listed vulnerabilities.
    2. Priorities/Remediations are assigned purely based on the report by SAST tool(snyk.io).

# Assessment Results

Following spreadsheet gives granular details about the vulnerabilities detected during the assessment.



Note: There are two tabs in the spreadsheet - one for AWS infrastructure and the other for the web application.

# Severity, Risk-score & Prioritization

Severity – Quantifies the security impact when an exploit happens.

Risk-score – Measures the likelihood of exploit happening and the frequency of possible exploit.

Prioritization – Answers the question - what vulnerability needs to be handled first?

Here’s the info that summarizes the procedure involved in the process of prioritization.

|  |  |
| --- | --- |
| Severity | Severity score |
| Critical | 4 |
| High | 3 |
| Medium | 2 |
| Low(Default) | 1 |

**Risk score** = (likelihood of exploit \* frequency of exploit)  
**Vulnerability score** = (Severity-score + Risk-score)/2

|  |  |
| --- | --- |
| Vulnerability  score | Priority |
| >=3.5 | P0 |
| >=2.5 | P1 |
| >=1.5 | P2 |
| Default | P3 |

# Recommendations

Remediations for the vulnerabilities are suggested in the spread sheet attached in Section3. Refer to the “Remediations” column for each entry in the spreadsheet.