Vulnerability Management

`2021.1`

Policy Statements

MiCT policy requires that:

(a) All product systems must be scanned for vulnerability at least quarterly and with each major change.

(b) All vulnerability findings must be reported and tracked to resolution. Records of findings must be retained for at least seven years.

Controls and Procedures

Vulnerability Scanning and Infrastructure Security Testing

The scanning and identification of system vulnerabilities is performed by

1. Automated security agent installed on all Linux servers.
    - This includes physical and virtual servers hosted on premise as well as EC2 instances in AWS.
    - The agent automatically reports to a centralized management server/dashboard with details of the server instance and any vulnerability finding.
    - This assessment is performed on an ongoing basis.
2. Additionally, periodic security scans of MiCT on-premise systems are done using a combination of open-source and commercial vulnerability testing tools, including:
    - OpenVAS
    - Nmap
    - OWASP ZAP
    - Burp Suite Pro
3. Penetration testing is performed regularly as part of the MiCT vulnerability management policy.
    - External penetration testing is performed continuously through a public vulnerability disclosure / bug bounty program.
    - Additional external penetration testing is performed at least annually by either a certified penetration tester on MiCT security team or an independent third party.
    - White-box internal penetration testing is performed at least quarterly by the security team.
4. MiCT developed an internal vulnerability management tool/database used to track all system entities and associated vulnerabilities.

5. Findings from a vulnerability scan or penetration testing are analyzed by the security team, together with IT and Engineering as needed, and reported following the process as defined in the next section. A written report may be generated in addition to creating the findings in Github.
6. All security testing reports and findings records are retained for 7 years.

Security Findings Reporting, Tracking and Remediation

We follow a simple vulnerability tracking process using Github. The records of findings are retained for seven years.

Reporting a finding

- Upon identification of a vulnerability (including vulnerability in software, system, or process), a Github Issue of (issueType = **Finding**) is created on the SECURITY Project.
- Populate the following custom fields as part of the Github issue when applicable:
  - **Source of Finding** (dropdown list)
  - **In Production** (yes/no/na selection)
  - **Application/Repo Name** (text/tag)
  - **Version Number** (text/tag)
- The **Summary** of the Finding should be in this format: "{[sev]} {short description}" (e.g. "[High] Outdated package on ECS AMI image").
- The **Description** of the Finding should include further details, without any confidential information, and a link to the source.
- The **Priority** of the Finding should match its severity level.

Priority/Severity Ratings and Service Level Agreements In an effort to quickly remediate security vulnerabilities the following timelines have been put in place to drive resolution.

Sev Rating	Priority Level	SLA	Definition	Examples
P0	Highest	3 days	Vulnerabilities that cause a privilege escalation on the platform from unprivileged to admin, allows remote code execution, financial theft, unauthorized access to/extraction of sensitive data, etc.	Vulnerabilities that result in Remote Code Execution such as Vertical Authentication bypass, SSRF, XXE, SQL Injection, User authentication bypass
P1	High	7 days	Vulnerabilities that affect the security of the platform including the processes it supports.	Lateral authentication bypass, Stored XSS, some CSRF depending on impact.
P2	Medium	30 days	Vulnerabilities that affect multiple users, and require little or no user interaction to trigger.	Reflective XSS, Direct object reference, URL Redirect, some CSRF depending on impact.

Sev Rating	Priority Level	SLA	Definition	Examples
P3	Low	Best Effort	Issues that affect singular users and require interaction or significant prerequisites (MitM) to trigger.	Common flaws, Debug information, Mixed Content.

In the case a sev rating / priority level is updated after a vulnerability finding was originally created, the SLA is updated as follow:

- **severity upgrade:** reset SLA from time of escalation
- **severity downgrade:** SLA time remains the same from time of creation/identification of finding

Resolving a finding

- The Finding should be assigned to the owner responsible for the system or software package.
- All findings should be addressed according to the established SLA.
- No software should be deployed to production with unresolved HIGH or MEDIUM findings, unless an Exception is in place (see below).
- A finding may be resolved by
    1. providing a valid fix/mitigation
    2. determining as a false positive
    3. documenting an approved exception

Closing a finding

- The assignee should provide a valid resolution (see above) and add a comment to the finding.
- The finding should be re-assigned to the Reporter or a member of the security team for validation.
- Upon validation, the finding can be marked as Done (closed) by the Reporter.
- **Before the finding can be marked as closed by the reporter, the fix must be deployed to dev and have a targeted release date for deploying to production noted on the ticket.**

Exceptions

- An Exception may be requested when a viable or direct fix to a vulnerability is not available. For example, a version of the package that contains the fix is not supported on the particular operating system in use.
- An alternative solution (a.k.a. compensating control) must be in place to address the original vulnerability such that the risk is mitigated. The compensating control may be technical or a process or a combination of both.
- An Exception must be opened in the form of a Github issue (issueType = **Exception**) on the SECURITY project.
- The Exception Github issue must reference the original Finding by adding an Issue Link to the Finding Github issue.
- Each Exception must be reviewed and approved by the Security team and the impacted asset owner.
- All Exceptions are reviewed every six months to re-assess its validity.