LATACORA

Fleet Device Management Application Security Assessment

CFD

June 2024

Executive Summary

In June of 2024, Latacora performed an application penetration assessment of Fleet's desktop and server applications.

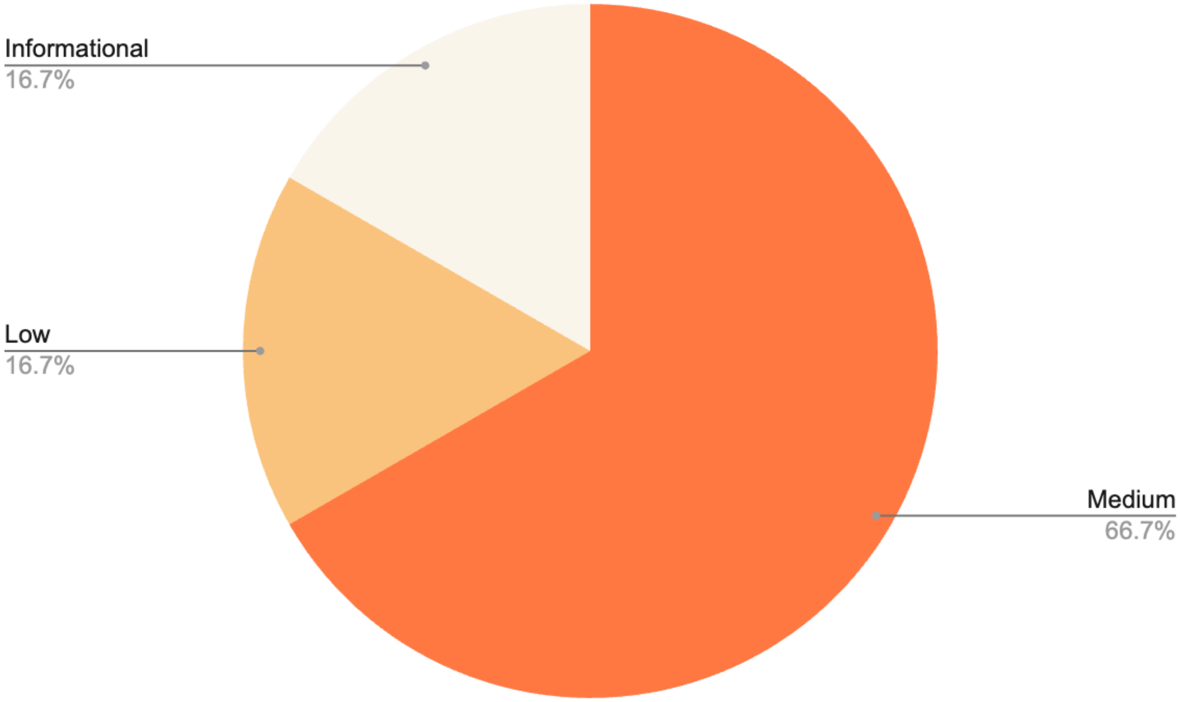
An application penetration test captures a point-in-time assessment of vulnerabilities, misconfigurations, and gaps in applications that could allow an attacker to compromise the security, availability, processing integrity, confidentiality, and privacy (SAPCP) of sensitive data and application resources. An application penetration test simulates the capabilities of a real adversary, but accelerates testing by using information provided by the target company.

This **customer facing document** relates the methodology our team used to conduct the assessment, key findings, and a summary of the coverage and findings achieved. The purpose of this report is to capture the assessment work done and the outcome of that assessment.

About Latacora and the Application Security Testing Team

Latacora is a boutique security firm that provides application, network, cloud, and corporate security services to startups. The principal members of Latacora were founders of Matasano Security and executive management at NCC Group, North America's largest software security firm, and Rackspace Managed Security.

Discovered Findings



Testing identified 4 Medium severity, 1 Low severity, and 1 Informational findings.

Coverage Obtained

Latacora reviewed all application targets from a source code review perspective and performed dynamic manual testing of the applications in their respective production environments.

Methodology

Logistics

Testers were given access to Fleet source code repositories as well as access to a production environment running an up-to-date version (4.50.1) of the platform, with credentials sufficient to replicate all the levels of privilege involved in the defined testing scope.

Scope

The scope of this assessment included the following web and client applications:

- https://sectest.cloud.fleetdm.com/ (Main web application)
- The corresponding client-side components, fleetctl and the Orbit installer generated by fleetctl, for Mac and Windows.

Discovery

At the start of an assessment, we identify and classify risks within the business context of the application.

In an application security assessment of a single coherent software target, the goals of Discovery include:

1. **Enumerate pre-authentication attack surface**: we seek to separate the portion of the application attack surface that doesn't require credentials from the portion that does. Pre-auth attack surface is especially exposed to "OWASP Top 10"-style web attacks.
2. **Identify administrative functionality**: we locate in the code any functionality limited to administrators and determine the code paths used to control access to that code. Administrative functionality is especially sensitive to authorization and privilege escalation attacks, and sometimes, with a threat model that includes trusted administrators, less sensitive to certain other classes of attacks.
3. **Map authorization controls**: we build an understanding of the levels of access different users have, and the application-layer controls that enforce those levels of access. A major focus of application security testing is statically and dynamically exercising these boundaries.
4. **Catalog dependencies**: modern web applications are almost invariably built on collections of third-party libraries and Fleet is not an exception. Dependencies must be analyzed for known vulnerabilities. Additionally, dependencies of security significance – those including memory-unsafe C

code, or which enforce security boundaries – receive direct inspection as a task in the assessment.

5. **Classify data**: the sensitivity of data stored in financial services applications is blended. All data accesses need to be assured for integrity and database safety. Sensitive nonpublic personal information (NPI) needs further assurances, such as at-rest data protection and additional authorization controls.

Because we had access to source code, we were able to straightforwardly enumerate all exposed endpoints (frontend and API), break down which portions of the backend they implicated, and evaluate how they were authenticated.

Testing Goals

Application Security Testing Goals	
Authentication	◦ Pre-authorization attack surface is cataloged and verified ◦ Password form fields are of type password ◦ Credentials are delivered over TLS ◦ Credentials are not inadvertently disclosed to a third party, on client or server ◦ Strong password complexity guidance and validation ◦ Brute force password countermeasures ◦ Password reset hygiene ◦ Email address verification
Authorization	◦ Privilege escalation ◦ Co-tenant segregation ◦ Privilege change requires additional authentication ◦ Mass assignment ◦ Verb tampering ◦ Consistency of authorization across applications ◦ Information leakage/disclosure ◦ Rate limiting ◦ File handling
Session Management	◦ Consistent, logically centralized session management ◦ Cryptographically opaque client-visible session IDs. ◦ Proper session invalidation ◦ Session timeout ◦ Session fixation ◦ Session equivalent cookies cataloged, audited and validated ◦ SSO controls are consistent with application controls
Data Storage	◦ SQL query safety and ORM usage ◦ User inputs are typecast, sanitized and/or parameterized ◦ Encryption ◦ Non-SQL or non-database data storage (such as file systems)
Server Configuration and Deployment	◦ Exposed service ports ◦ Transport encryption, TLS and cipher suites ◦ Certificate expiry, validation and pinning ◦ HSTS ◦ Fingerprinting software versions ◦ File upload and post-processing ◦ Application extensions

	◦ Header splitting ◦ Web application firewall ◦ Host headers ◦ Server side request forgery
Application Logic	◦ Format of currency values ◦ Handling of payments and refunds ◦ Validation of awards, credits, or discounts ◦ Application can be used to validate stolen credit cards ◦ Altering prices or other payment amounts ◦ Canceling orders after fulfillment
Cryptography	◦ Encrypted data is authenticated, using an allowlist of allowed AEAD constructions ◦ Ciphers are selected from an allowlist of allowed ciphers ◦ Exposed encrypted tokens and capability grants are not reusable ◦ Exposed encrypted tokens are time stamped and checked for expiry
Application Hygiene	◦ Exposed stack traces ◦ Exposed server side artifacts such as path names ◦ Unused code paths ◦ Debugging code or other information ◦ Exposed server side source code ◦ Redirect handling
Administrative Functionality	◦ Administrative functions are out of band ◦ Administrative resources require 2FA ◦ Cross-site request forgery ◦ Reflected and stored Cross-site scripting (XSS) ◦ Audit trails and logging ◦ User impersonation features hygiene

Risk Ratings

When a risk is identified, Latacora calculates an impact or risk rating score as part of documenting the finding. Impact can be broken down into factors aligned with the traditional security areas of concern: security, availability, processing integrity, confidentiality, and privacy. The goal is to estimate the magnitude of the impact on the system if the vulnerability were to be exploited. The following areas of concern (abbreviated SAPCP) are:

- **Security** - which addresses the resistance of the application and connected systems to hostile attacks, such as DDoS attacks, injection attacks, account takeover attempts, server-side forgery, and direct object references.
- **Availability** - which addresses service interruption and the critical nature of down time, as well as the adequate provisioning of resources.
- **Processing Integrity** - which addresses complete, correct, accurate, and timely processing of data
- **Confidentiality** - which addresses storage and handling of data, as well as how much and what kind of data should be disclosed or retained.
- **Privacy** - which addresses the requirement for notice to data subjects about data collection, choice and consent to data collection and access to collected data, and the system's use, retention, disclosure, and disposal of personal information according to the applicable regulations.

Determining Severity

The severity of each risk is determined by analyzing likelihood along with impact. This table describes the severity scale used throughout Latacora's assessments:

Classification	Description
INFORMATIONAL	Not exploitable. Potentially no action required.
LOW	Limited impact on SAPCP. High awareness/easily detected. Difficult to exploit or not directly exploitable. May be useful for reconnaissance. Potentially no action required.
MEDIUM	Moderate impact on SAPCP. High awareness/possibly detected. Difficult to exploit. Some logging, some monitoring.
HIGH	Significant impact on SAPCP. Low awareness/difficult to detect.
CRITICAL	Serious impact on SAPCP. No awareness/no detection. Suggest immediate remediation.

Summary of Findings

Reference	Description	Severity
LT-APP24-1	Hosts can Access Any Software	Medium
~~**LT-APP24-2**~~	~~Deployment Link Pointing to Vacant Domain~~	~~Medium~~
LT-APP24-3	Observers can Access ABM Keys	Medium
LT-APP24-4	Lack of Name Validation in Device Update	Informational
LT-APP24-5	Observers can Access Any Software	Medium
LT-APP24-6	MDM Status Leaked to Unauthenticated Users	Low

In addition to these net-new findings, Latacora's reviewers were aware of several vulnerabilities previously reported, which have either been risk-accepted or are still pending mitigation. Examples include formula injection in CSV export and SSRF.

Typically, this Latacora report would include re-testing and re-reporting of all such open findings, so that this report would be comprehensive if read by itself. Since Fleet publicly discloses these issues on https://fleetdm.com/handbook/business-operations/security-audits, for brevity they are not reiterated in this year's report from Latacora. Readers seeking a comprehensive understanding of Fleet's security should read this document in context of the URL above.