COMMON VULNERABILITIES AND EXPOSURES (CVE) TREND & VULNERABILITY INTELLIGENCE ANALYSIS

Author: Pavel Pecheniuk

Date: 30.07.2025

1. Abstract

This report provides an overview of vulnerabilities trends observed during the first half of the 2025 year. The analysis is general, also it is focused on the vulnerabilities of a highest relevance for Fin Analytics and the fintech industry in general, such as those affecting authentication systems, web and mobile applications, cloud environments. In the following sections, the short approach description is given, findings are presented and elaborated on, and further probable impact on Fin Analytics along with the mitigation plans are discussed.

---

2. Methodology

In this section, an approach for analysis conduction is described. The key points are summarized below:

Data:

- CISA Known Exploited Vulnerabilities (KEV) Catalog for a period of time 01.01.2025 – 30.06.2025. The full database can be found here https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- The database utilized in the analysis is supplemented with Common Vulnerability Scoring System (CVSS) scores for corresponding vulnerabilities. Scores are fetched from the https://www.cvedetails.com/
- Assignment of severity levels is taken from NIST, MITRE, CISA and from the vendor's estimation.

Selection Criteria:

- Data related to our field of interest;
- CVSS score ≥ 7.0.

Tools:

- Microsoft Excel: data filtering and visualization.

---

3. Findings

3.1 Vulnerabilities by Severity

Let's gain an insight to the distribution of CVEs by their severity levels. Below are the Figure 1 and Figure 2, depicting this distribution.

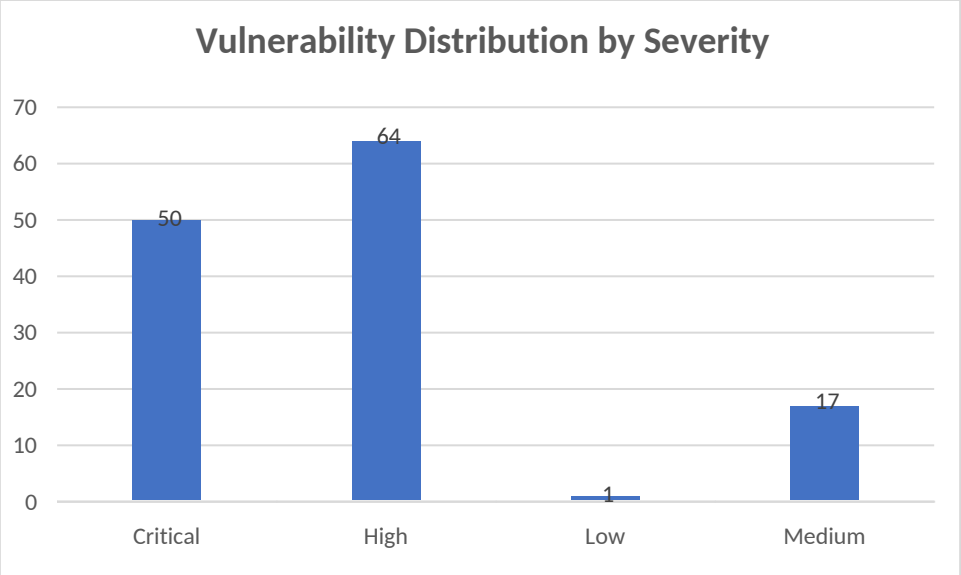

Figure 1. Number of vulnerabilities dependent on their severity levels

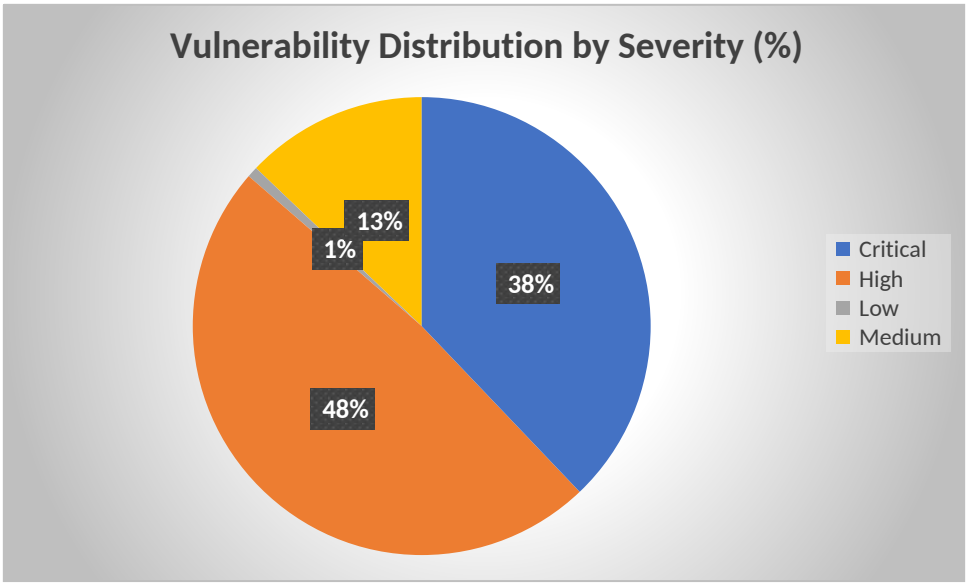

Figure 2. Percentage of vulnerabilities by severity levels

Outcome: Out of 132 vulnerabilities 86% of known exploited vulnerabilities were rated as High and Critical, which emphasizes the need for continuous threat monitoring and proactive patch management.

3.2 Categorization of Vulnerabilities

To identify the type of the most frequent known observed vulnerabilities, Common Weakness Enumeration (CWE) was evaluated across the CVE database. Top 10 most common CWEs for the first half of 2025 are summarized below in the Table 1.

CWE ID	Description	Frequency (%)
CWE-78	OS Command Injection	9.1
CWE-22	Path Traversal	7.6
CWE-416	Use-After-Free	6.8
CWE-502	Deserialization of Untrusted Data	6.1
CWE-79 CWE-122 CWE-787	Cross-Site Scripting (XSS) Heap-Based Buffer Overflow Out-of-Bounds Write	3.8
CWE-94 CWE-125	Code Injection Out-of-Bounds Read	3.1
CWE-287	Improper Authentication	2.3

Table 1. CWE of current vulnerabilities

Outcome: The four main exploitable vulnerabilities are command injection, path traversal, use-after-free and deserialization of untrusted data, together summing up approximately to 30% of all known exploitable vulnerabilities currently.

3.3 Potential Affect on Fin Analytics

Having the core threatening vulnerabilities, let's now concentrate on their impact on Fin Analytics. **That is a fintech cloud start-up** which offers financial analytics and investment portfolio management tools for clients. Additionally, it provides to the clients a possibility to maintain operations via web and mobile application, operating APIs for handling sensitive financial data, such as customer data and transactions.

Now let's define which vulnerabilities pose a greatest threat to the Fin Analytics assets. Based on all we know, **the most relevant those which:**

- Threaten the confidentiality and integrity of data;
- Can be exploited remotely – common issue of API/cloud/web environment;
- Jeopardize regulatory compliance – GDPR.

Let's analyse the most frequent vulnerabilities and define the most dangerous ones. Below is the proposed rank of vulnerabilities, complemented with the description and reasoning. Explanations are done with the help of data from the CWE intelligence database operated by The MITRE Corporation (MITRE), and from The OWASP Foundation.

1. Deserialization of Untrusted Data (CWE-502)

What: Malicious modification of deserialized data or code.

Why: APIs of Fin Analytics web and mobile apps handles the data exchange and flows of financial operations. If deserialization is present, it becomes possible for penetrators to execute malicious code remotely via backdoor to manipulate client transactions, compromise company's backbone infrastructure or tamper client data, affecting clients and reputation.

2. Improper Authentication (CWE-287)

Note: MITRE discourages vulnerability mapping for this particular CWE since it is a high-level entity describing authentication failures. However, it is still meaningful to reflect authentication-related weaknesses discussed in our analysis. For precision I refer also to CWE-1390: Weak Authentication as it is recommended by MITRE here.

What: Lack of proper mechanism confirming that clamed identity is correct.

Why: Negative impact of weak authentication is significant and versatile – unauthorized access poses a threat to any possible component of the system, leading to the consequences like client data theft, financial fraud and financial damage to the company expressed in non-compliance penalties.

3. Cross-Site Scripting (XSS) (CWE-79)

What: Injection of malicious code scripts into trusted websites.

Why: XSS targets web and mobile portals, causing the massive negative consequences – an attacker can craft the scripts to disclose confidential information, hijack the session, and manipulate ongoing processes such as transactions.

4. Code Injection (CWE-94)

What: Insertion of malicious code which is then executed by vulnerable application.

Why: Code injection poses a threat to a web app, a mobile app, APIs or backend system processing dynamic user's input. Mentioned components are exploited to compromise integrity and availability of the data. Among the possible negative outcomes is manipulation of client's financial data.

---

4. Recommended Mitigation Strategies

Now when the most relevant vulnerabilities to Fin Analytics are identified, let's establish some general and detailed guidelines, aiming to mitigate risk and prevent security breaches.

Technical Measures for Web Vulnerabilities:

Deserialization of Untrusted Data:

- Reduction of deserialization's impact - safe execution of deserialized code;
- Usage of safe data formats like JSON instead of serialization;
- Employment of authentication mechanisms on serialized objects.

XSS:

- Input and output sanitization with Django framework and specialized libraries for user input;
- Implementation of measures for blocking inline scripts such as Content Security Policy (CSP) headers;

Code Injection:

- Validation of external input before processing;
- Avoidance of dynamic code execution functions like eval;

Technical Measures for Authentication Vulnerability:

Improper Authentication:

- Enforcement of Multi-Factor Authentication (MFA) for accounts of all privilege levels;
- Deployment of centralized identity management;
- Hardening password policies;
- Introduction of role-based access control (RBAC) and least privilege;
- Conduction of system logs audit with focus on authentication-related events (logins, privilege escalations).

General Mitigation Plans:

- Continuous conduction of vulnerability analysis by means of vulnerability scanning and penetration testing;
- Application of immediate patch management for the most critical vulnerabilities;
- Reviewing vulnerabilities databases like CISA KEV/NIST intelligence to stay up-to-date with threats posed by evolving vulnerabilities;
- Hardening key components of infrastructure even if they are not explicitly endangered: maintenance of secure cloud configuration and encryption of data, monitoring of APIs' operation and patching it if necessary, update of Web Application Firewall (WAF), monitoring of system behaviour by means of log analysis for the events of interest.

---

5. Summary

Performed analysis of common vulnerabilities trends and impact of specific vulnerabilities on the fintech start-up named Fin Analytics for a first half of 2025 year allows to conclude the following:

- The majority of observed known vulnerabilities for a given period of time labelled as High or Critical, indicating necessity for continuous threat monitoring.
- Currently, the most common vulnerabilities are command injection, path traversal, use-after-free and deserialization of untrusted data.
- Deserialization of untrusted data, improper authentication, cross-site scripting and code injection represent the most noticeable threat to the company's infrastructure, emphasizing the need to apply security measures to strengthen the web infrastructure and access control.
- It is necessary to stay consistent and up-to-date while managing overall security posture via continuous analysis of existing vulnerabilities and exploration of threat intelligence databases.