Security Vulnerabilities Addressed in Fluent Bit v4.1 and Backported to v4.0

Summary of security issues reported and remediated in Fluent Bit v4.2, v4.1.1, and v4.0.13, including path traversal, stack buffer overflow, and authentication bypass fixes.

Written by Fluent Bit Team

October 28, 2025

Scope: Fluent Bit Security issues reported in GitHub Security Advisory (restricted access)

On October 23, 2025, we were notified of a few security issues present in Fluent Bit. Those security issues have been remediated in the latest versions currently available of Fluent Bit v4.2, v4.1 (4.1.1) as well as v4.0 (4.0.13). Below please find a summary of the issues and remediation.

---

1. Executive Summary

Purpose: Summarize recently disclosed vulnerabilities affecting pipelines running 3.x.

Key Findings:

- **Off-by-one read in percent-decoder (CWE 193)** – Not a security issue. Logic bug has been remediated.
- **Tag key prefix match (CWE 187)** – Potential for data corruption but not expanded access. Exploitability requires access to the endpoint of the service.
- **Path traversal in Out_File (CWE 35)** – Potential for disruptive access to the filesystem beyond permitted bounds.
- **Stack buffer overflow in in_docker (CWE 121)** – Potential for denial of service or arbitrary code execution. Requires the ability to invoke Fluent Bit with the Docker safety issue.
- **Missing authentication in in_forward (CWE 306)** – Potential to send traffic to Fluent Bit or other destinations without authentication.

Overall Risk Rating: External exposure possible

---

2. Summary of New and Updated CVEs

ID	Description	Exploit Status	Remediation Status
CWE 193	The issue cannot be exploited for memory corruption, data leakage, or arbitrary code execution. It may only lead to incorrect string decoding.	None reported	Bug fix - not a security issue (no CVE applicable). Remediated in version 4.1.1 and 4.0.13
CWE 187	A remote attacker with access to input endpoints (e.g. HTTP, Splunk, Elasticsearch) could manipulate tag prefixes to inject data into unintended backends. This affects data integrity, not confidentiality or memory safety.	None reported	Security issue - CVE applicable (data integrity impact). Remediated in version 4.1.1 and 4.0.13
CWE 35	An attacker able to control the tag value could cause Fluent Bit to write outside the configured directory path. Affects file integrity and host filesystem isolation.	None reported	Security issue - CVE applicable. Remediated in version 4.1.1 and 4.0.13
CWE 121	A long container name from the Docker API could trigger a stack overflow. The issue is locally exploitable and may lead to denial of service (crash) or potentially arbitrary code execution.	None reported	Security issue - CVE applicable. Remediated in version 4.1.1 and 4.0.13

ID	Description	Exploit Status	Remediation Status
CWE 306	Remote attackers could send data without credentials, bypassing authentication controls. This compromises the authenticity of ingested logs and can allow injection of forged data.	None reported	Security issue - CVE applicable. Remediated in version 4.1.1 and 4.0.13

3. Detailed Analysis

CWE 193 - Off-by-one read in percent-decoder

- **Impact:** The issue cannot be exploited for memory corruption, data leakage, or arbitrary code execution. It may only lead to incorrect string decoding.
- **Resolution:** Invalid escape sequences are now preserved as raw strings and are not decoded - bug fixed. Not a security issue (no CVE applicable).
- **References:** PR #10961

CWE 187 - Tag key prefix match

- **Impact:** A remote attacker with access to input endpoints (e.g. HTTP, Splunk, and Elasticsearch) could manipulate tag prefixes to inject data into unintended backends. This affects data integrity, not confidentiality or memory safety.
- **Resolution:** The tag_key matching logic has been reimplemented using the record accessor API, enforcing strict and complete key matching. Security issue identified; CVE applicable (data integrity impact).
- **References:** PR #10967

CWE 35 – Path traversal in `Out_File`

- **Impact:** An attacker able to control the tag value (provided via input plugins such as HTTP or forward) could cause Fluent Bit to write outside the configured directory path. Affects file integrity and host filesystem isolation.
- **Resolution:** File name sanitization and path canonicalization have been implemented to prevent traversal outside the allowed path. Security issue identified; CVE applicable.
- **References:** PR #10969

CWE 121 - Stack buffer overflow in `in_docker`

- **Impact:** A long container name when Fluent Bit is invoked from the Docker API could trigger a stack overflow. The issue is locally exploitable and may lead to denial of service (crash) or, potentially, arbitrary code execution.
- **Resolution:** Buffer handling has been hardened and binding copies are enforced. Security issue identified; CVE applicable.
- **References:** PR #10972

CWE 306 - Missing authentication in `in_forward`

- **Impact:** Remote attackers could send data to an in-forward endpoint without valid credentials, bypassing authentication controls. This may compromise the authenticity of ingested logs and can allow injection of forged data.
- **Resolution:** Authentication logic has been re-implemented and verified in multiple follow-up fixes. Security issue identified; CVE applicable.
- **References:** PR #10973, PR #11026, PR #11028

4. Recommended Next Steps

Immediate:

- Update to Fluent Bit v4.1 or v4.2