Path injection in SecureDrop Client read_gzip_header_filename()

High **iiatyy** published **GHSA-2jrc-x8fq-prvc** 4 days ago

Package

Client (SecureDrop)

Affected versions	Patched versions
=<0.17.2	0.17.3

Description

Summary

A malicious SecureDrop Server could obtain code execution on the SecureDrop Client virtual machine (`sd-app`). SecureDrop Server itself has multiple layers of built-in hardening, and is a dedicated physical machine exposed on the internet only via Tor hidden services for the Source and Journalist interfaces, and optionally via remote SSH access over another Tor hidden service. A newsroom's SecureDrop Workstation communicates only with its own dedicated SecureDrop Server.

Despite the exploitability requirements, given the high potential impact, the vulnerability is classified as "High" with a CVSSv3 score of 7.5.

Thank you to cookiejack15 for reporting this through the SecureDrop bug bounty program; we've awarded them $2,500 for the discovery.

This vulnerability is similar to CVE-2025-24888 in terms of impact but through a different code path. As the SecureDrop Client is being phased out for the new SecureDrop Inbox, a more robust solution has been implemented in the new codebase.

Score breakdown

Factor Name	Factor Value	Description
Attack Vector (AV)	Network (N)	The attack can only be performed by the specific SecureDrop Server for which this client is configured.
Attack Complexity (AC)	High (H)	Given the requirement to compromise a major component of the system, the server, complexity is High.
Privileges Required (PR)	None (N)	No further privileges on the target client are necessary.
User Interaction (UI)	Required (R)	A journalist must download a file provided by a source, however it is expected that journalists will regularly do this.
Scope (S)	Unchanged (U)	The vulnerability lies in software running in the `sd-app` Qubes virtual machine, and affects mainly that component.
Confidentiality (C)	High (L)	Confidentiality impact is high, because the affected component contains the decrypted content of source submissions, unless manually deleted.
Integrity (I)	High (L)	Integrity impact is high, since it is trivial to gain persistence and the affected virtual machine is one of the main components of the system.
Availability (A)	High (N)	Availability impact is high, since once code execution is gained, data and code required for the main application to function could be deleted.

Technical details

The SecureDrop Client runs in a dedicated Qubes virtual machine, named `sd-app`, as part of the SecureDrop Workstation. The private OpenPGP key used to decrypt submissions and replies is stored in a separate virtual machine and never accessed directly.

The vulnerability lies in the code responsible for extracting the filename from compressed submissions. The filename is obtained from the gzip archive header used to write the decrypted file on disk. Note these filenames are generated and sanitized server-side, so a remote attacker who has not achieved server compromise, such as one posing as a source, could not craft the gzip archive needed for this attack.

While the filename is checked to guard against path traversal, it incorrectly permitted absolute paths. In this case, `check_path_traversal()` would permit an absolute path, allowing the attacker to specify any path within `/home/user/.securedrop_client/`. Code execution could be gained by saving a malicious SQLite database to `/home/user/.securedrop_client/svs.sqlite`.

We are not aware of any exploitation in the wild. This attack requires a previously compromised SecureDrop Server.

Severity

High 7.5 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	High
Privileges required	None
User interaction	Required
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H

CVE ID

CVE-2026-35465

Weaknesses

▶ CWE-73

Credits

 **cookiejack15** Reporter