0-DAY NO CVE HIGH PUBLIC

ProFTPD SFTP copy-file skips dest WRITE

ProFTPD 1.3.9d SFTP copy-file (default-on once SFTPEngine is on) skips dest Limit WRITE when the dest does not exist yet. Authenticated SFTP can plant a new file in a WRITE DenyAll tree. Labbed. Not RCE.

NAME	TYPE
ProFTPD SFTP copy-file skips dest WRITE	0-day analysis
CVE	**CVE RISK**
n/a	high
DISCLOSURE STATUS	**VENDOR**
public	ProFTPD Project
AFFECTED	**PUBLISHED**
ProFTPD 1.3.9d (f59f248) contrib mod_sftp. copy-file is in SFTP_FXP_EXT_DEFAULT. Inverse of public Issue 2095 (FTP SITE COPY source READ).	12 Oct 2026
UPDATED	**TAGS**
12 Oct 2026	0-day, proftpd, sftp, privilege-leftover, file-write, cwe-863

TL;DR

Lab files and proof-of-concept here.

01 OPEN was denied. copy-file created it anyway.

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/proftpd-sftp-copy-file-write** (loopback client). The lab stack is `lab/`: Dockerfile, docker-compose.yml, proftpd.conf, run.sh, poc.py. Authorized lab only. It talks to loopback.

This is **proftpd/proftpd** tag **v1.3.9d** commit `f59f248e26bdfa514443e42bf72a145641df635d`. `PROFTPD_VERSION_TEXT "1.3.9d"`. No CVE yet. CWE-863. 7.1 High (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N`). Remote privilege leftover. Authenticated SFTP. Contrib `mod_sftp`. Not RCE.

`<Limit WRITE> DenyAll` is supposed to mean no create and no overwrite in that tree. SFTP `OPEN` with write/create flags returns status 3, `Permission denied`. Then `EXTENDED copy-file` of a readable source onto a dest name that does not exist yet returns `OK`, and the dest contains the witness. The WRITE check runs only if `pr_fsio_stat` on the dest succeeds. A missing dest never gets that check. `pr_fs_copy_file2` creates it.

That is the leftover. Inverse of public Issue 2095, which skipped source READ on FTP `SITE COPY`. This sink skips dest WRITE on SFTP `copy-file` for a new name.

02 Why ProFTPD after FileZilla

I had just finished FileZilla Client 3.71.1. That leftover is a Windows listing of a directory named `..` that survives the client's `.` / `..` drop, then `CreateFileW` without `\\?\` strips the trailing space and walks out of the download dir. Unix and macOS are closed for it. I left the client on the table and opened a server.

ProFTPD is the large GPL FTP/FTPS/SFTP daemon Debian still ships, with a contrib `mod_sftp` that people turn on when they want SSH file transfer without standing up a second daemon. Sample `basic.conf` leaves `DefaultRoot` commented. `RootRevoke` defaults on, so after login the worker is the login UID, not 0. Core static modules are `mod_core`, `mod_xfer`, `mod_auth`, `mod_site`, and friends. Contrib

(`mod_sftp` , `mod_copy` , `mod_tls` , `mod_sql` , `mod_rewrite` , `mod_site_misc`) is off until loaded. Distro shared builds commonly ship them. `SFTPEngine on` is the checkbox. `copy-file` is in `SFTP_FXP_EXT_DEFAULT` . You do not ask for it.

I only wanted Critical or High. Remote and local. Medium stays on the floor unless it chains. DoS stays on the floor unless it is a write primitive. Anonymous FTP as designed is not a leftover. Unauth banner / FEAT / SYST is recon. Default core FTP unpublished Crit/High on this pin was none. The leftover that survived is contrib `mod_sftp` .

03 Pinning 1.3.9d, not 1.3.10rc4

GitHub tags offered v1.3.9d on 2026-08-14, commit `f59f248` , and v1.3.10rc4 on the same calendar day, commit `769b780` . 1.3.10rc5 is unreleased master NEWS. RCs wait. Debian still marks 1.3.9d. `include/version.h` says `PROFTPD_VERSION_TEXT "1.3.9d"` and `PROFTPD_VERSION_NUMBER 0x0001030908` . That is the pin.

1.3.10rc4 added contrib `mod_procfs` (Issue 2261). That module is not in 1.3.9d. A leftover that only exists because you pinned an RC is a different product.

The clone is the tagged tree. Sixteen parallel hunts: FTP parser, path canonicalization, SITE, MLSD/facts, chroot/privs, Controls, `mod_sftp` , `mod_tls` , `mod_sql` , rewrite / `.ftpaccess` / exec, auth, data/REST, wrap2 / ldap / radius / snmp, digest / json / display, ifsession / DSO. Parent verified named leftovers against `fxp_handle_ext_copy_file` , `copy_copy` , `site_misc_rmdir` / `dir_check_canon` , `core_dele` , `rewrite_map_int_utf8trans` . Three independent labs on 1.3.9d, loopback, witness-only. No exploit payloads.

This post is the SFTP `copy-file` create. The same wave kept two more High leftovers on contrib: FTP `SITE COPY` dest follows a symlink, and `RewriteMap int:utf8trans` overflows two BSS buffers. Those are other write-ups. Default core FTP stayed empty.

04 The skip list is half the hunt

Before you look for leftovers, you write down what is already public so you do not "discover" a CVE with a press release. ProFTPD's 2026 crop and the GitHub issues still sitting in this tarball:

- CVE-2026-42167 / Issue 2052. `mod_sql` `is_escaped_text()` SQLi via USER / SQLLog expansions (`%U` `%f` `%{env}` `%{note}`). Fixed 1.3.9a; extra env/note in 1.3.9b. Skip.
- CVE-2026-44331 / Issue 2057. `mod_wrap2_sql` rDNS hostname SQLi (`UseReverseDNS`). Fixed 1.3.9b. Skip.
- CVE-2026-35025 / Issue 2170. Authenticated RNFR `/proc/self/root` ACL bypass of `<Directory>` / `<Limit>` via unresolved symlink in `dir_canonical_path()`. Public. Upstream declined a core `/proc` special case. 1.3.10rc4 added contrib `mod_procfs`. Debian still marks 1.3.9d vulnerable. Skip as public. `DefaultRoot` chroot is the documented mitigation. Sample has it commented out.
- CVE-2026-53994 / CVE-2026-63090 / Issue 2190. Authenticated SFTP heap overflow in `contrib/mod_sftp/fxp.c` packet reassembly. Fixed 1.3.9c. Skip.
- CVE-2026-63091. Authenticated SCP signed integer overflow (`UINT64_MAX` into a negative `off_t`). Fixed 1.3.9c. Skip.

Public GitHub issues still present in the 1.3.9d tarball, fixed later on the 1.3.9 branch or only on 1.3.10. Skip even when the code is still here:

- Issue 2095. FTP `SITE COPY` skips `<Limit READ>` on the source. Public. `mod_copy`. Inverse of this leftover, different protocol, different check.

- Issue 2256. Cross-filesystem RNTO EXDEV fallback follows dest symlinks. Public. Reporter: no auth bypass / no LPE.
- Issue 2286. REST/RANG `2^63` into a negative `off_t`. Public. Still in 1.3.9d `parse_offset`. Backported after this tag.
- Issue 2319. Stale `authenticated_without_pass` after TLS `AllowDotLogin` / `TLSUserName` USER. Public. Backported after this tag.
- Issue 2330. `mod_radius` long PASS stack overflow. Public. Contrib.
- Issue 2175. SITE RMDIR misses `PathAllowFilter` / `PathDenyFilter`. Public. `mod_site_misc`. Not backported to 1.3.9.
- Issue 2180. SFTP `SETSTAT` / `FSETSTAT` miss `PathAllowFilter`. Public.
- Issue 831. `UserAlias` outside `<Anonymous>` treated as anonymous. Public since 2019. Master-only fix.

NEWS 1.3.9a-d already closed the rest of the parser and Controls wave: Issue 2019 toupper non-ASCII, 2020 LF-leading 1-byte OOB read, 2030 LIST/NLST `-B` signed-char heap, 2146 MLSD/MLST stack overflow, 2173 RewriteMap unescape OOB, 2188 SQL group-name concat, 2210 Controls socket ACLs allow all users, 2163 `AdminControlsACLs` `config` / `get`, 2242 SSH max packet size 0, 2265 STAT `-C` UAF, 2272 PASV `AllowForeignAddress`, 2275 / 2279 empty passwords, 2282 `.ftpaccess` policy bypass. SSH ETM/ChaChaPoly underflow, empty payload NULL deref, DSA heap OOB, RSA/DSA algo mismatch, SFTP request underflow: 1.3.9b. Terrapin is 1.3.9rc2. `RootRevoke` default on since 1.3.7rc1.

Historical `mod_copy` is CVE-2015-3306 unauth `SITE CPFR` / `CPTO` and CVE-2019-12815 `AllowAll` default. Auth is in the tree. Default-allow copy is not. The leftover here is not "copy exists." It is dest WRITE gated on `stat` success.

05 Surfaces that returned nothing

Most of the tree did what the comments claimed.

FTP parser. `pr_cmd_read` → `pr_netio_telnet_gets2` → `make_ftp_cmd` → dispatch. Issue 2019/2020/1683/2030 closed on this pin. The 1-byte extra read leftover of 2020/1683 is a NUL already overwritten in a `pcalloc`'d buffer, not a write primitive. SITE is not shell injection. ProFTPD does not exec. Unknown methods 500 before the handler. HTTP/SMTP/SSH2 banners disconnect. HOST after login is 503. Unauth cannot RETR/STOR/LIST/SITE CHMOD. CCC requires auth. That hunt came back empty for unpublished Crit/High.

Chroot and privs. `RootRevoke` defaults true. After PASS / SFTP auth the session does `PRIVS_REVOKE` then `session.disable_id_switching = TRUE`, so later `PRIVS_ROOT` is a no-op. `RootLogin` stays off. `CreateHome` stays off. Issue 1830 GID 0 supplemental groups closed in 1.3.9rc3. `AllowChrootSymlinks` default on is the documented model, not a leftover. Sample `basic.conf` does not chroot. Without a jail, the process can `stat` any path the login UID can. That is the non-jail model, not a leftover. CAP_NET_BIND_SERVICE after login does not bind an attacker-chosen privileged port. No unpublished uid-0 leftover on default login.

Controls, Issue 2210. `--enable-ctrls` defaults off. When compiled in, socket mode is 0777 so any local uid can connect, and that used to be the bug: empty ACL inverted to allow everyone. 1.3.9c switched `ctrls_init` to `pr_ctrls_init_acl`. Empty allow-list is deny. Per-action ACLs are deny-all except documented `help`. Issue 2163 `config`/`get` now call `pr_ctrls_check_acl`. Linux `SO_PEERCRED` is not spoofable as another uid. Kick / shutdown / restart / config / insmod all ACL-gate before root-process work. Unbounded Controls `msglen` is public Issue 2036 DoS. Skip. No unpublished LPE on the unix socket.

Auth, Issue 2319. Empty `AuthUserFile` password is closed (2279). Empty `mod_sql_passwd` is closed (2275). Stale `authenticated_without_pass` after a TLS `AllowDotLogin` / `TLSUserName` USER is public 2319. Skip that exact flag. `UserAlias` outside `<Anonymous>` is public 831. `AuthOrder` does not remove PAM

from the auth-only list; docs already require `AuthPAM off` for file-only. Name-collision footgun, not default. `RequireValidShell`, `RootLogin`, PAM conv, `ftpasswd` stay below bar or are documented policy. No unpublished Crit/High on FTP authentication.

SQL and wrap2. CVE-2026-42167 pre-escapes client-supplied LogFormat vars and sets `already_escaped`. The start/end-quote heuristic is still there. The public callers are not. CVE-2026-44331 escapes wrap2_sql names before `sql_lookup`. Issue 2188 group-name concat is 1.3.9c. No unpublished SQLi on this pin.

Data / REST. Issue 2272 `AllowForeignAddress` is closed. PORT/EPRT/PASV share `pr_inet_allowforeignaddress`. SFTP has no PORT/PASV data socket; `CHANNEL_OPEN` accepts only `session`. Issue 2286 REST/RANG `2^63` is still in `parse_offset` and stays skip-public. Its STOR-no-truncate effect does not seek before byte 0.

TLS. `TLSEngine` off until set. Issue 2319 is the passwordless leftover, public. Cert mapping is exact `strcmp` of CN / Email SAN / OID to USER. Data reuse is required unless CCC, `NoSessionReuseRequired`, or cache off. No unpublished High that maps a cert to a different user in a single USER.

MLSD / facts / digest / ifsession / DSO. Issue 2146 buffer is two `PATH_MAX` plus 256. Issue 2030 `-B` casts to `unsigned char`. Issue 2265 STAT `-C` is ignored. HASH of a file the UID can Unix-open but `<Limit RETR>` denies is a Medium hash oracle, not a full read. `PerUnauthenticatedUser` is off. `ModulePath` refuses a world-writable directory. JSON parse depth is 16.

Those hunts were not empty because I stopped reading. They were empty because core FTP, after the 2026 CVE crop, does not have an unpublished Crit/High on this pin. Contrib `mod_sftp` did.

06 How copy-file checks WRITE

`SFTP_FXP_EXT_DEFAULT` includes `SFTP_FXP_EXT_COPY_FILE`. The client sends `EXTENDED copy-file` with src, dst, overwrite (`fxp.c` around 7428). Handler `fxp_handle_ext_copy_file` around 4631-4844:

01 Builds a fake `SITE COPY` `cmd_rec` and dispatches **PRE_CMD** only. `mod_copy` registers `copy_copy` as **CMD**, not PRE_CMD. FTP's dest WRITE check in `copy_copy` never runs on this path, even if `mod_copy` is loaded.

02 `dir_best_path` on src and dst. Resolved. Not the 2170 lexical `/proc` class.

03 Dest WRITE plus `AllowOverwrite` run **only if** `pr_fsio_stat(dst)` succeeds (file already exists).

04 `PathAllowFilter` on src and dst.

05 Source READ is always `dir_check`'d. Unlike 2095, source READ is enforced.

06 `pr_fs_copy_file2(src, dst, 0, NULL)` creates the missing dest with no dest WRITE check.

FTP STOR / SFTP OPEN-write go through `xfer_pre_stor`, which always `dir_check`s WRITE before create. copy-file does not.

The dest branch, shortened:

```c
pr_fs_clear_cache2(dst);
res = pr_fsio_stat(dst, &st);
if (res == 0) {
  /* AllowOverwrite + dir_check(..., "WRITE", dst) only when dest exists */
```

```
    ...
}

cmd2 = pr_cmd_alloc(fxp->pool, 3, "SITE_COPY", src, dst);
if (!dir_check(fxp->pool, cmd2, "READ", src, NULL)) {
  /* source READ is always checked */
    ...
}

res = pr_fs_copy_file2(src, dst, 0, NULL);
```

`stat` failure is the create path. There is no `else { dir_check WRITE }`. A tree with `<Limit WRITE> DenyAll` and READ allowed is a drop-box inverted: you may retrieve, you may not plant. OPEN honors that. copy-file honors it only after the dest exists.

`SFTPExtensions -copyFile` turns the extension off. Docs list `copyFile` as enabled by default with every other extension except `vendorID` and `xattr`. Typical distro SFTP vhost advertises it in the version packet. The lab saw `copy-file` in the extension list without asking.

Issue 2095 is the other way around. FTP `SITE COPY` `dir_check`s G_WRITE on dest and never G_READ on src, so you copy a READ-denied file into a WRITE-allowed dir. Public. This leftover copies a READ-allowed file into a WRITE-denied dir, and only when the dest name is new. Same family of "copy did not use the same Limit as STOR/RETR." Opposite check, opposite protocol.

07 Wrong turns, in the order I made them

Pinning 1.3.10rc4 because it was the same day. Calendar is not stability. `mod_procfs` is an RC feature. The Debian-shaped product is 1.3.9d.

Treating default core FTP empty as "no High on this product." Core with sample `basic.conf` really is empty for unpublished Crit/High. Contrib SFTP is what distros load when they say the daemon also speaks SSH files. Config-gated is still in bar when the gate is `SFTPEngine on` and the extension is default-on.

Filing Issue 2095 again. Source READ skip on FTP `SITE COPY` is public. Dest WRITE skip on SFTP `copy-file` for a missing dest is not that ticket. I went back to `copy_copy`'s cmdtab to make sure I was not hallucinating a PRE_CMD that does not exist. `{ CMD, C_SITE, G_WRITE, copy_copy, ... }`. CMD. The fake SITE COPY from SFTP never reaches it.

DELE `/proc/self/root`. `core_dele` is `dir_canonical_path` plus lexical `dir_check` plus `unlink`. Same unresolved procfs-root class as public CVE-2026-35025 / Issue 2170, which is documented on RNFR. Issue 2170 already asked about DELE. Skip-public, same class. SFTP `REMOVE` uses `dir_check_canon` and is closed for this class. `mod_procfs` is not in 1.3.9d. I dropped it.

SITE RMDIR recursive follow. The SITE hunt claimed Limit on the lexical symlink, then `stat` / `opendir` walk of the target, so `SITE RMDIR` of a planted link would `rm -fr` a WRITE-denied tree. SITE RMDIR actually `dir_check_canon`s. That is `dir_check` of `dir_best_path`. An existing symlink is resolved before Limit. Lab of the SITE COPY sibling: `SITE RMDIR /upload/link` → `550`, victim intact. Core RMD of that same link stays `550`. Dropped. The SITE COPY dest leftover is a different function: `copy_copy` Limit-checks unresolved `dir_canonical_vpath`, then `pr_fs_copy_file2` follows. That is G2, not this post.

copy-data. Not implemented on 1.3.9d. Issue 2198 is a feature. 1.3.10rc3 NEWS implements it. No handler to mis-ACL.

SCP `-f` / `-t` as RCE. `handle_exec_channel` splits with `pr_str_get_word`. First word `scp` matches the handler. getopt is `"dfprtv"`. No shell. `recv_filename` rejects `/` and `..` in the protocol name. Not this bug.

Assuming OPEN and copy-file share a Limit helper. OPEN dispatches PRE_CMD of STOR. STOR always WRITE-checks. copy-file invented its own `stat`-then-maybe-WRITE. That is why the lab opens the dest with `WRITE|CREAT|EXCL` first and requires status 3 before it sends `copy-file`. If OPEN had succeeded, the run would be FAIL. The Limit would not have been the Limit.

A reverse shell. Theatre. The oracle is a witness string in a dest the session UID created under DenyAll.

08 Lab

Compose project `proftpd-sftp-copy-file-write`. Image built from the 1.3.9d tree with `--with-modules=mod_sftp`. `SFTPEngine on`. `copy-file` left in the default extensions. Auth is `mod_auth_file`. Loopback `127.0.0.1:18710` → container `2222`.

Two trees. `readable/src.txt` AllowAll, contains `PROFTPD-SFTP-COPY-FILE-WRITE-WITNESS`. `nowrite/` is:

PLAIN TEXT Copy

```
<Directory /nowrite>
  <Limit WRITE STOR APPE MKD RMD DELE RNFR RNTO>
    DenyAll
  </Limit>
  <Limit READ RETR>
    AllowAll
  </Limit>
</Directory>
```

Same block on `/home/labuser/nowrite`. Dest name `/nowrite/PROFTPD-SFTP-COPY-FILE-WRITE-WITNESS` does not exist at start.

The PoC is a raw SFTP v3 client on a paramiko SSH channel. Stock Paramiko SFTP does not send `copy-file`. The version packet listed `copy-file` among the default extensions. Then:

01 OPEN-read of `/readable/src.txt`. Witness present. 37 bytes.

02 OPEN-write of the dest with `WRITE|CREAT|EXCL`. Status **3**, `Permission denied`. That is `xfer_pre_stor` plus `<Limit WRITE>`.

03 `EXTENDED copy-file` src → dest, overwrite false. Status **0**, `OK`.

04 OPEN-read of the dest. Same 37 bytes. Witness present. Container `cat` of the dest file agrees.

05 `copy-file` again with overwrite true onto the dest that now exists. Status **3**, `Permission denied`.

Last line:

PLAIN TEXT Copy

```
SUCCESS PROFTPD-SFTP-COPY-FILE-WRITE dest-exists=yes open-write=denied copy-file=
```

Replay:

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/proftpd-sftp-copy-file-write
cd proftpd-sftp-copy-file-write/lab
./run.sh
```

Bind it to loopback. The loopback client is `lab/run.sh` with a banner on it.

09 What an attacker can do

Authenticated SFTP, `SFTPEngine on`, `copy-file` still in the default extensions. Typical distro SFTP vhost. The account already has a readable file. The interesting

dest is a tree the operator wrapped in `<Limit WRITE> DenyAll` while leaving READ on: incoming vs secret split, or a drop-box inverted. Unix DAC still applies. The session is the login UID after `RootRevoke`. No shell. Not RCE. Not unauth.

This is the privilege set. Mirror of the pack G1 table.

METHOD
SFTP `EXTENDED copy-file` of a readable src to a dest name that does not exist yet, under `<Di`
Same dest already exists
SFTP OPEN / STOR / SCP upload of the same dest
Unauthenticated copy-file

Lab-proven vs inferred. Dest create, dest READ of the witness, OPEN-write deny, and existing-dest overwrite deny are lab-proven. Unauth and Unix DAC remain inferred from the auth / chroot model. STOR/SCP closed is the same `xfer_pre_stor` path OPEN already hit; the lab sent OPEN, not a second FTP STOR.

What this leftover does not grant: overwrite of a file that is already there, create as uid 0, chroot escape, a listing extra, another user's home, a command channel, RCE. Confidentiality of dest contents you did not plant is not this bug. If dest READ is also allowed, you get your own planted bytes back. CVSS stays `C:N` `I:H`. Bump C only if a different dest READ policy is in play and you are counting confirmation as confidentiality.

The operator who set `<Limit WRITE> DenyAll` already told ProFTPD that STOR, APPE, MKD, and OPEN-write are forbidden in that tree. copy-file is a second create path that did not get the memo until the dest existed.

10 The fix

`dir_check` WRITE on the dest even when `pr_fsio_stat` fails. Create is a write. STOR already knows that. copy-file should too.

Until then, `SFTPExtensions -copyFile` turns the extension off. That is the documented switch. Default is on once `SFTPEngine` is on.

I am not going to print an SFTP `copy-file` you can paste at someone else's `:22`. The missing dest WRITE is the useful part. If you own the daemon, run the proof of concept against loopback.

11 Contact

Abraxas Labs - abraxaslabs.tech - github.com/abraxas - @abraxas_null - abraxas.null@proton.me

12 References

- Proof of concept: abraxas/proftpd-sftp-copy-file-write · proftpd-sftp-copy-file-write-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · proftpd.conf · run.sh · poc.py
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-863 · CVSS 7.1
- Tree: proftpd/proftpd v1.3.9d (`f59f248e26bdfa514443e42bf72a145641df635d`) · `fxp_handle_ext_copy_file` · `SFTP_FXP_EXT_DEFAULT` · `copy_copy` cmdtab · `xfer_pre_stor` · `SFTPExtensions`
- Inverse public bug: Issue 2095 FTP SITE COPY source READ

Abraxas Labs

Independent security research by abraxas.
Analysis of 0-days, n-days, malware, and the leftovers that still matter.

GitHub **@abraxas_null** **abraxas.null@proton.me** **abraxaslabs.tech**

Home Research Open Source Reading Room About GitHub @abraxas_null Console

Independent research. Private analysis by arrangement. No ads. abraxas.null@proton.me · For defensive awareness and authorized testing only.

© 2026 Abraxas Labs