0-DAY NO CVE HIGH PUBLIC

vsftpd cmds_denied misses RFC aliases

vsftpd 3.0.5 cmds_denied is an exact token match. Denying RMD still allows XRMD to rmdir. Labbed. Authenticated FTP privilege leftover.

NAME vsftpd cmds_denied misses RFC aliases	TYPE 0-day analysis
CVE n/a	CVE RISK high
DISCLOSURE STATUS public	VENDOR vsftpd (Chris Evans)
AFFECTED Official vsftpd 3.0.5 (SHA-256 26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3). cmds_denied set. Default unset.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, vsftpd, cmds_denied, privilege-leftover, cwe-863, remote

TL;DR

Lab files and proof-of-concept here.

01 RMD was 550. XRMD was 250.

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

This is **vsftpd 3.0.5**, Chris Evans, the official tarball (`vsftpd-3.0.5.tar.gz` , SHA-256 `26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3`). Not a GitHub mirror. Not a Debian or Fedora fork. `vsftpver.h` says `VSF_VERSION "3.0.5"` . GPL-2.0-only WITH vsftpd-openssl-exception. 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. Authenticated FTP. Privilege leftover.

The operator wrote `cmds_denied=RMD` . That is the virtual-user recipe: share a guest uid, take rmdir away in the FTP policy. `RMD labdir` came back `550 Permission denied.` `XRMD labdir` came back `250 Remove directory operation successful.` `CWD labdir` was `550 Failed to change directory.` The directory was gone.

RFC 775 named the aliases in 1980. vsftpd still dispatches them. `cmds_denied` never heard of them.

I did not rmdir host `/etc` . The oracle is a directory the session uid can already Unix-rmdir, inside the anonymous incoming tree, after FTP policy said no. `rmdir(2)` said yes.

02 Why vsftpd, after Pure-FTPd

I had just finished Pure-FTPd 1.0.54. Fifteen hunts. Default core unpublished Crit/High: none. The public tickets sitting on that tag were Medium availability, RFC conformance, a TLS 1.3 session-reuse WONTFIX, and a LIST/SIZE metadata gap that RETR already blocked. Nothing at this bar.

Then I wanted the other famous C FTP daemon. The one that named itself Very Secure. Two-process: unprivileged protocol child, privileged parent that only binds

PASV and PORT-connects to the control peer. Seccomp on the dropped child since 3.0.0. `isolate` and `CLONE_NEWNET`. A 2021 last GA, August, and then silence. That is vsftpd.

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-as-designed is out of bar. Unauthenticated banner / FEAT / SYST is recon. Distro packaging is a different tree. Firmware `vsftpd.conf` that turns `chroot_local_user=YES` is some other product's conf, not this C.

Default compiled core FTP on this pin came back empty. The leftover that survived is config-gated. This one is `cmds_denied`. Default unset. The man page names it for virtual-user lock-down. Common advice is `cmds_denied=DELE,RMD`. That string does not mention `XRMD`.

Same pin, two other leftovers I am not mixing into this post: brace `deny_file` missing children, and FTPS data accepting any resumed ticket. This write-up is the denied verb and the alias that still ran.

03 Pinning official 3.0.5

Pin the official tarball. Not a GitHub mirror. Not a distro fork.

CVE-2025-14242 is why. Red Hat introduced a STAT/LIST range parser in their `ls.c`. That parser integer-overflows. CNA: Red Hat. Medium 6.5, authenticated STAT DoS. Debian's tracker marks upstream `<not-affected>`: the vulnerable code is a Red Hat-specific patch. Fedora has a fix commit of their own. If you clone a GitHub "vsftpd-3.0.5" or unpack a Fedora src.rpm, you are not hunting the daemon Chris Evans ships from security.appspot.com. You are hunting a distro delta.

SHA-256 of `vsftpd-3.0.5.tar.gz`:

PLAIN TEXT Copy

```
26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3
```

The lab image greps `VSF_VERSION "3.0.5"` in `vsftpver.h` before `make`. Banner in the run is `220 (vsFTPd 3.0.5)`. `vsftpd -v` in the image is the same string.

Stock `builddefs.h` has `#undef VSF_BUILD_SSL`. Distros typically `#define` it, and PAM is already `#define VSF_BUILD_PAM` upstream. This leftover does not need SSL. The lab leaves dummyinc headers in place and links `-lcrypt`. It is the official tree, not a package.

Compiled defaults that bound the hunt: `listen=1`, `anonymous_enable=1`, `local_enable=0`, `write_enable=0`, `ssl_enable=0`, `http_enable=0`, `seccomp_sandbox=1`, `isolate=1`, `isolate_network=1`, `pasv_promiscuous=0`, `port_promiscuous=0`, `allow_writeable_chroot=0`, `chroot_local_user=0`. `tunable_cmds_denied` is NULL. Stock 3.0.5 without that string is not this leftover. KEEP is the operator who sets it.

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. vsftpd's list is short, and most of the 2025 noise is not this tarball.

- CVE-2025-14242 Red Hat-introduced STAT integer overflow. Medium 6.5. Debian not-affected. Pin `ls.c` has no `[` range parser. Not upstream.
- CVE-2021-3618 ALPACA. Generic TLS+FTP application-layer confusion. 3.0.3 kills HTTP verbs (Jann Horn). 3.0.4 rejects HTTP verbs pre-login, closes after ten unknown pre-login commands, rejects non-`ftp` ALPN, adds `ssl_sni_hostname`. 3.0.5 only fixes ALPN select for FileZilla-3.55.0. Unmatched HELO/EHLO/PUT still 530 then the ten-error close. Not a new unmatched HTTP leftover.

- CVE-2021-30047 connection-limit DoS on 3.0.3. Availability. Below this bar.

- CVE-2015-1419 distro `deny_file` realpath/basename. Public. Official 3.0.5 Changelog has no such fix. Do not re-file a public distro CVE against this tarball. The brace-child leftover on the same pin is a different walk, and it is not this post.

- CVE-2011-2523 malicious 2.3.4 tarball backdoor, a few days in 2011. Not this SHA-256.

- CVE-2011-0762 `vsf_filename_passes_filter` glob CPU DoS before 2.3.3. `VSFTP_MATCHITERS_MAX` 1000 is on this pin. Not a deny bypass.

- Firmware `vsftpd.conf` CVEs: TOTOLink / Linksys / TRENDnet / H3C / D-Link turning `chroot_local_user=YES` or opening anonymous upload in a vendor conf. Operator tunables. Other products.

3.0.x hardening already in the tarball: seccomp (3.0.0), kill on HTTP verbs and SSL reuse-fail (3.0.3), pre-login HTTP-kill and TLS 1.2 floor (3.0.4), FileZilla ALPN (3.0.5). Sample `vsftpd.conf` already documents ASCII SIZE as predicted and safe.

Default core FTP unpublished Crit/High: none.

Documented tunables I am not filing: `pasv_promiscuous` / `port_promiscuous` (FXP), `allow_writeable_chroot`, `http_enable`, `hide_file` still RETR-able, `one_process_model` (anonymous-only). The man page says what they do.

05 Surfaces that returned nothing

Nineteen hunts. FTP parser, PORT/PASV/FXP, SSL/TLS, seccomp/ptrace, privdrop/caps, chroot/path, PAM/virtual users, ls/STAT filter, post-login xfer, parseconf/`user_config_dir`, standalone listener, str/tunables/secbuf,

Debian/Fedora patch delta, hide/deny/`cmds_denied`, logging/banner/HTTP, sysdeputil/sysutil, isolate/process model, ASCII/REST/SIZE, features/OPTS/hash.

Most of that is vsftpd doing what vsftpd said it would do. The named High guards still hold.

PORT / PASV / FXP. Compiled defaults `pasv_promiscuous=0`, `port_promiscuous=0`. PORT/EPRT refuse a target whose IP is not the control peer, and refuse ports `< 1024`. PASV accept must match the control IP. The privileged parent ignores a hostile child's sockaddr and connects `p_remote_addr` from `getpeername`. Classic FTP bounce to `:25` / RFC1918 / metadata is closed unless someone turns promiscuous on. That is documented FXP. Sample conf does not.

`user_config_dir` after PAM. Overlay is off by default. When an admin sets it, `handle_per_user_config` runs only after a successful PASS, concatenates `user_config_dir + '/' + USER`, refuses any USER that contains `/`, and the parser requires a regular file owned by the daemon uid. Unauthenticated USER never names a config path. No include. No stack buffer. SIGHUP reloads the listener on the saved main filename. The 1.2.1pre2 "Paranoia: ignore user_config_dir if username has a / in it" comment is still the only basename guard, and it is still there.

ASCII SIZE. Sample conf says other servers DoS on `SIZE /big/file` in ASCII, and that vsftpd predicted it. `handle_size` `stat`s, requires `S_ISREG`, reports raw `st_size`. No open, no read, no `ascii.c`. REST in TYPE A is refused before open. REST is an offset on the later command's own fd. Not a second leftover.

SITE CHMOD `& 0777`. `str_octal_to_uint` then `str_chmod` with a comment that the implementation only allows `0 - 0777`. Anonymous never gets SITE. `chmod_enable=NO` kills it without `cmds_denied`. `cmds_denied=CHMOD` never matches: the FTP command is `SITE`. That is a footgun against a wrong token, not this alias leftover.

Pre-login table. USER, PASS, QUIT, FEAT, OPTS, and if SSL is on, AUTH / PBSZ / PROT. Everything else is `530 Please login with USER and PASS.` Ten unknowns

close. SIZE, RETR, STOR, PORT, PASV, SITE, CWD, RMD, XRMD are not in that table. `cmds_denied` is a post-login gate. Pre-login ignores it. No unauth rmdir.

HTTP-kill. GET / POST / HEAD / OPTIONS / CONNECT `vsf_cmdio_write_exit` both pre-login and post-login. 3.0.4 ALPACA work. Default `http_enable=0`. When FTP is on, GET is killed, not served. Dual-protocol is `ftp_enable=NO && http_enable=YES`, anonymous-only, and that path still hits `deny_file` via `handle_retr`. Documented, default off.

Parser leftovers of the 4096-byte command line, pipelining, and CRLF class are closed. `str_netfd_alloc` peeks to first LF and consumes only that. AUTH TLS leftover of a ClientHello `die()`s. Seccomp on x86_64: no `execve` / `ptrace` / `mount` / `clone` for the dropped child. `open` is allowed; FS boundary is chroot. Writeable-chroot deny default on.

Then I opened `process_post_login` next to the dispatch ORs.

06 `cmds_denied` is an exact token. Dispatch is not.

`process_post_login` in `postlogin.c` is the only gate. After `vsf_cmdio_get_cmd_and_arg` (which `str_upper`s the verb in `ftpcmdio.c`), it walks comma tokens and `str_equal`s the uppercased command against each one. Denial is exact. There is no alias table. There is no second check inside `handle_rmd`.

C Copy

```c
    if (tunable_cmds_denied)
    {
      static struct mystr s_src_str;
      static struct mystr s_rhs_str;
      str_alloc_text(&s_src_str, tunable_cmds_denied);
      while (1)
      {
        str_split_char(&s_src_str, &s_rhs_str, ',');
        if (str_isempty(&s_src_str))
        {
```

```
          break;
        }
        else if (str_equal(&s_src_str, &p_sess->ftp_cmd_str))
        {
          cmd_ok = 0;
          break;
        }
        str_copy(&s_src_str, &s_rhs_str);
      }
    }
    if (!cmd_ok)
    {
      vsf_cmdio_write(p_sess, FTP_NOPERM, "Permission denied.");
    }
```

`cmds_allowed`, if set, is the other walk: empty remainder means `cmd_ok=0`. Fail-closed. Unlisted verbs die, including `XRMD`. The man page calls that "a powerful method of really locking down an FTP server." Example: `cmds_allowed=PASV,RETR,QUIT`. That is the allowlist. `cmds_denied` is the denylist people actually paste.

Then the same function ORs RFC 775 aliases (and one SMC-router joke) onto the modern handlers:

C Copy

```
    else if (str_equal_text(&p_sess->ftp_cmd_str, "CWD") ||
             str_equal_text(&p_sess->ftp_cmd_str, "XCWD"))
    {
      handle_cwd(p_sess);
    }
    else if (tunable_pasv_enable &&
             !p_sess->epsv_all &&
             (str_equal_text(&p_sess->ftp_cmd_str, "PASV") ||
              str_equal_text(&p_sess->ftp_cmd_str, "P@SW")))
    {
      handle_pasv(p_sess, 0);
    }
    else if (tunable_pasv_enable &&
             str_equal_text(&p_sess->ftp_cmd_str, "EPSV"))
    {
```

```
    handle_pasv(p_sess, 1);
  }
  else if (tunable_write_enable &&
           (tunable_anon_mkdir_write_enable || !p_sess->is_anonymous) &&
           (str_equal_text(&p_sess->ftp_cmd_str, "MKD") ||
            str_equal_text(&p_sess->ftp_cmd_str, "XMKD")))
  {
    handle_mkd(p_sess);
  }
  else if (tunable_write_enable &&
           (tunable_anon_other_write_enable || !p_sess->is_anonymous) &&
           (str_equal_text(&p_sess->ftp_cmd_str, "RMD") ||
            str_equal_text(&p_sess->ftp_cmd_str, "XRMD")))
  {
    handle_rmd(p_sess);
  }
```

`PWD` or `XPWD`. `CDUP` or `XCUP`. `STOR` is its own token. `STOU` and `APPE` are other tokens that still call `handle_upload_common` / `handle_stou`. `HELP` even advertises `XCWD XMKD XPWD XRMD` on the wire.

`handle_rmd` does not ask `cmds_denied` again. Tilde resolve, `vsf_access_check_file` (the `deny_file` glob, a different leftover), `str_rmdir`. If Unix says yes, `250 Remove directory operation successful.`

`SITE` vs `site` is not a bypass. Verbs are uppercased. `cmds_denied=CHMOD` never matches because the command is `SITE`. Kill chmod with `chmod_enable=NO` or `cmds_denied=SITE`. The deny **list** is not uppercased: `cmds_denied=site` is a silent no-op against `SITE`. Comma split does not trim: `cmds_denied=DELE, RMD` makes the second token `RMD`. Those are footguns against a mistyped list. The leftover against a correctly-cased `RMD` is `XRMD`.

`P@SW` is in the Changelog: "Recognize P@SW as PASV; works around an SMC router bug." `cmds_denied=PASV` does not deny `P@SW`. `EPSV` is a separate token and a separate `handle_pasv(..., 1)`.

That is the split. Same uid. Same Unix mode. Same `cmds_denied`. `RMD` hits `str_equal`. `XRMD` walks off the end of the list and lands in the OR.

07 The lab

Official 3.0.5 built in Docker from the pinned tarball. Loopback `127.0.0.1:18720`. Compose project `vsftpd-cmds-denied-alias`. Witness `VSFTPD-CMDS-DENIED-ALIAS-WITNESS`. Image greps `VSF_VERSION "3.0.5"`, greps `cmds_denied` and `XRMD` in `postlogin.c`, then `make`.

`lab/Dockerfile` is the pin:

DOCKERFILE Copy

```
# Pin: official vsftpd 3.0.5 SHA-256 26b602ae454b0ba6d99ef44a09b6b9e0dfa7f672281
COPY vsftpd-3.0.5.tar.gz /src/vsftpd-3.0.5.tar.gz

RUN echo "26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3  vsft
        | sha256sum -c - \
    && tar -xzf vsftpd-3.0.5.tar.gz \
    && test -f vsftpd-3.0.5/vsftpver.h \
    && grep -q 'VSF_VERSION "3.0.5"' vsftpd-3.0.5/vsftpver.h \
    && grep -n 'cmds_denied' vsftpd-3.0.5/postlogin.c \
    && grep -n 'XRMD' vsftpd-3.0.5/postlogin.c
```

Official Makefile uses `-Werror`. gcc 12 on bookworm fails the pin without stripping it. SSL/PAM/libcap stay on dummyinc so this is not a distro package.

`lab/docker-compose.yml` binds FTP to loopback:

YAML Copy

```
    ports:
      - "127.0.0.1:18720:21"
```

`lab/vsftpd.conf`: `anonymous_enable=YES`, `write_enable=YES`, `anon_mkdir_write_enable=YES`, `anon_other_write_enable=YES`, `cmds_denied=RMD`, `allow_writeable_chroot=NO`. Chroot is `/var/ftp`, owned `root:root` mode `755`. Writable tree is `/var/ftp/incoming`, `ftp:ftp` mode `0775`. That is the documented

anonymous-upload layout. `seccomp_sandbox=NO` and `isolate=NO` are lab plumbing so PID 1 in a slim container does not fight SIGCHLD. They are not the leftover.

Control, then the alias, then the closed neighbor:

01 Banner `220 (vsFTPd 3.0.5)`. `USER ftp` / `PASS x@x` → `230`.

02 `CWD incoming` → `250`.

03 `MKD labdir` → `257 "/incoming/labdir" created`.

04 `RMD labdir` → `550 Permission denied.`

05 `XRMD labdir` → `250 Remove directory operation successful.`

06 `CWD labdir` → `550 Failed to change directory.` Directory gone.

07 `MKD labctrl` → `257`. `RMD labctrl` → `550`. The denylist still matches `RMD` after the alias has fired. `XRMD labctrl` → `250` again.

Replay:

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/vsftpd-cmds-denied-alias
cd vsftpd-cmds-denied-alias/lab
./run.sh
```

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

PLAIN TEXT Copy

```
SUCCESS vsftpd-cmds-denied-alias rmd=550 xrmd=250 dir-gone=yes VSFTPD-CMDS-DENIED
```

XMKD, XCWD, STOU, APPE, EPSV, `P@SW` are the same `str_equal` gate. The lab drives `RMD` / `XRMD`. The others are inferred from the ORs in `postlogin.c`.

08 Wrong turns, in the order I made them

chmod 777 on `/var/ftp`. Anonymous vsftpd chroots to the ftp home. `allow_writeable_chroot` defaults off, and the lab leaves it off. If the jail itself is writable by the session uid, login dies. I wanted a directory the uid could rmdir. The jail is `755` root. The incoming child is `0775` ftp. That is the layout the man page already described. The leftover is FTP policy, not a writeable-chroot bypass.

`network_mode: host` on Docker Desktop. On Linux, host network is the host. On macOS, host network is the Linux VM. Publishing `127.0.0.1:18720` from the VM is not publishing it on the Mac loopback the PoC dials. The compose file publishes ports. The client talks to `127.0.0.1:18720`. That is the loopback bind, not "FTP on the LAN."

Makefile `-Werror` on bookworm gcc. 3.0.5 last GA'd in 2021. gcc 12 is less impressed. The image `sed`s `-Werror` out and still greps the version string and the XRMD dispatch before install. Dummyinc stays. This is not `apt install vsftpd`.

Treating `cmds_denied` as fail-closed. It is not. Empty remainder of the denylist walk means "not found, allow." Empty remainder of the **allowlist** walk means deny. I had to put the two loops next to each other before the leftover was "the denylist is a string compare" instead of "vsftpd command policy."

`cmds_denied=CHMOD` as this bug. SITE subcommands are not FTP commands. `cmds_denied=SITE` blocks CHMOD, UMASK, and HELP together. The documented way to kill chmod is `chmod_enable=NO`. Different token, different leftover class, not an alias.

`SITE` vs `site`. Uppercased. Not a bypass. Lowercase in the conf list is a silent no-op. Do not confuse a mistyped denylist with RFC 775.

Scoring it Medium because you already have Unix rmdir. Virtual-user recipes share `guest_username`. Unix DAC cannot separate those accounts. `cmds_denied` is the extra control the man page sold them. Running the denied verb through an alias is integrity High against that control. Availability of the directory is a side effect of `XRMD`, not the class.

Inflating rmdir to RCE. The session uid can `rmdir` a policy-denied directory it can Unix-rmdir. If that directory is later a deploy target, a queue, or someone else's incoming, that chain is not in the lab. Class is privilege leftover.

A pasteable `XRMD` against someone else's `:21`. Theatre. The missing alias is the useful part. If you own the daemon, run the proof of concept against loopback.

09 What an attacker can do

Needs an authenticated FTP login (anonymous or local) and `cmds_denied` set. Default unset is not this leftover. Write-family aliases also need `write_enable` (and the anon_write knobs for anonymous). Victim path must be Unix-reachable by the session uid. Virtual-user setups that share a guest uid cannot fall back to DAC. No unauth. No shell. No chroot escape. No root.

PRIVILEGE	METHOD / SINK
Integrity: rmdir a directory FTP policy denied	`XRMD dir` after `cmds_denied=RMD` (or `DELE,RMD`). Gate is exact `str_equal` on `ftp_cmd_str`. Dispatch ORs `RMD` / `XRMD` to `handle_rmd` → `str_rmdir`

PRIVILEGE	METHOD / SINK
Integrity: mkdir after `cmds_denied=MKD`	`XMKD dir`. Same exact-token gate. Dispatch ORs `MKD` / `XMKD` to `handle_mkd` → `str_mkdir`
Integrity: chdir after `cmds_denied=CWD`	`XCWD path`. Dispatch ORs `CWD` / `XCWD` to `handle_cwd` → `str_chdir`. Then `RETR` if `download_enable`
Integrity: chdir-up after `cmds_denied=CDUP`	`XCUP`. Dispatch ORs `CDUP` / `XCUP` to `handle_cdup`
Integrity: create/append after `cmds_denied=STOR`	`STOU` / `APPE`. Separate tokens. `handle_stou` / `handle_appe` → `handle_upload_common`. Needs `write_enable`
Integrity: open a data listener after `cmds_denied=PASV`	`EPSV` (separate token, `handle_pasv(..., 1)`) or `P@SW` (Changelog SMC-router alias ORed with PASV)
Recon: cwd string after `cmds_denied=PWD`	`XPWD`. Dispatch ORs `PWD` / `XPWD` to `handle_pwd`
Negative: `RMD` **still denied**	`str_equal` against `cmds_denied=RMD`
Negative: unauthenticated `XRMD`	pre-login table is USER/PASS/QUIT/FEAT/OPTS/AUTH
Negative: `cmds_allowed=PASV,RETR,QUIT`	allowlist fail-closed. Unlisted `XRMD` is `cmd_ok=0`
Negative: `write_enable=NO`	write family gated before the OR

PRIVILEGE	METHOD / SINK
Negative: `chmod_enable=NO`	SITE CHMOD
Negative: `SITE` **vs** `site`	`str_upper` in `ftpcmdio.c`
Negative: PORT/PASV bounce	`port_promiscuous` / `pasv_promiscuous` default off
Negative: `user_config_dir` **from USER**	load after PASS, reject `/` in USER, root-owned regular file
Negative: ASCII SIZE / REST TYPE A	raw `st_size`; ASCII resume refused
Negative: SITE CHMOD setuid	`& 0777`
Negative: HTTP GET as RETR	HTTP-kill; `http_enable` default off
Negative: chroot escape / host `/etc`	jail aliases the jail. Unix DAC still applies
Negative: root / shell / RCE inside vsftpd	none claimed

`DELE` has no RFC 775 alias in this dispatch. `cmds_denied=DELE,RMD` still kills `DELE` and still misses `XRMD`. That is the AskUbuntu-shaped recipe.

Unix DAC still applies when users do not share a uid. A local account that cannot `rmdir` the path cannot `XRMD` it either. The High is the shared-guest virtual-user case the man page pointed `cmds_denied` at.

10 The fix

Deny the aliases with the modern verb. `cmds_denied=RMD,XRMD` (and `MKD,XMKD`, `CWD,XCWD`, `CDUP,XCUP`, `PWD,XPWD`, `PASV,P@SW,EPSV`, `STOR,STOU,APPE` as needed). Or stop using a denylist: `cmds_allowed` is fail-closed for every unlisted token, which is what the man page already called the lock-down.

A real fix in `postlogin.c` is to treat the RFC 775 names (and `P@SW`) as the same command for both lists. One token in the conf should mean one handler. Until then, the denylist is a string compare against a dialect FTP clients and RFC 775 still speak.

I am not going to print an `XRMD` you can paste at someone else's `:21`. The missing alias is the useful part. If you own the daemon, run the proof of concept against loopback.

PLAIN TEXT Copy

```
SUCCESS vsftpd-cmds-denied-alias rmd=550 xrmd=250 dir-gone=yes VSFTPD-CMDS-DENIED
```

11 References

- Proof of concept: abraxas/vsftpd-cmds-denied-alias · vsftpd-cmds-denied-alias-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py · vsftpd.conf
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-863 · CVSS 7.1 High (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N`)
- Tree: official vsftpd 3.0.5 tarball `vsftpd-3.0.5.tar.gz` (SHA-256 `26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3`) · `postlogin.c` (`process_post_login` `cmds_denied` walk, then `RMD` / `XRMD`, `MKD` / `XMKD`, `CWD` / `XCWD`, `PASV` / `P@SW`, `EPSV`, `STOU`, `APPE`) · `ftpcmdio.c`

`str_upper` · `prelogin.c` USER/PASS/QUIT/FEAT/OPTS/AUTH · `vsftpd.conf.5` `cmds_denied` / `cmds_allowed`

- Nearby public (not this leftover): CVE-2025-14242 RH STAT DoS · CVE-2021-3618 ALPACA · CVE-2021-30047 · CVE-2015-1419 · CVE-2011-2523 · CVE-2011-0762

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**

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

© 2026 Abraxas Labs