0-DAY NO CVE HIGH PUBLIC

vsftpd FTPS data accepts any resumed ticket

vsftpd 3.0.5 ssl_accept binds FTPS data to SSL_session_reused() only. A same-IP peer who AUTH TLS (no password) can resume that ticket on the victim PASV and steal the RETR. Labbed. Distinct from the 3.0.3 kill-on-fresh-handshake.

NAME vsftpd FTPS data accepts any resumed ticket	TYPE 0-day analysis
CVE n/a	CVE RISK high
DISCLOSURE STATUS public	VENDOR vsftpd (Chris Evans)
AFFECTED Official vsftpd 3.0.5 compiled with VSF_BUILD_SSL (distro common; stock tarball undef), ssl_enable=YES, standalone listen=YES, require_ssl_reuse=YES (default).	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, vsftpd, ftps, tls-session-reuse, privilege-leftover, cwe-345, cwe-287, remote

TL;DR

Lab files and proof-of-concept here.

01 The man page promised a master secret. The C asked if any ticket decrypted.

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/vsftpd-ssl-data-any-ticket** (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**, the official tarball from security.appspot.com (SHA-256 `26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3`). `vsftpver.h` says `VSF_VERSION "3.0.5"`. No CVE yet. CWE-345 / CWE-287. 7.4 High (`CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N`). Remote. Privilege leftover. Distro FTPS compile. Same-IP PASV.

The man page for `require_ssl_reuse` is not shy. Default YES since 2.1.0. "If set to yes, all SSL data connections are required to exhibit SSL session reuse (which proves that they know the same master secret as the control channel)." Chris Evans wrote that in 2009, pointed at the blog post, and cited `SSL_session_reused` as the relevant OpenSSL API.

I opened `ssl_accept`. It calls `SSL_session_reused()` and stops there. Reused means the ClientHello carried a ticket this process can decrypt. It does not mean Alice's ticket. Mallory `AUTH TLS`s on his own control, holds that ticket, never sends USER or PASS, and resumes it on Alice's PASV. The data channel has no FTP login. The default bind is a boolean.

That is the leftover. Distinct from the 3.0.3 kill-on-fresh-handshake that Tim Kosse filed. The lab negative for that kill is still `522 session reuse required`. This leftover is the other direction: a *different* control's ticket, resumed, accepted.

I did not steal a password. I did not get a shell. The oracle is a 35-byte witness on Mallory's TLS while Alice's control still prints 150 then 226.

02 Why vsftpd

I wanted a C FTP daemon that still ships as "Very Secure," still has a two-process model with a privileged parent that only binds PASV, still treats OpenSSL as an optional compile, and still has a man page that claims a cryptographic bind the C does not implement. vsftpd is honest about a lot of things. `pasv_promiscuous` is documented FXP. `allow_writeable_chroot` is a documented bypass. `ssl_enable` defaults NO and the stock tarball does not even compile the SSL object in. Distros compile it in anyway, operators set `ssl_enable=YES`, and then the defaults force TLS 1.2+ for local logins and data and require session reuse.

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. Firmware `vsftpd.conf` CVEs are operator tunables of other products.

Default compiled core FTP on this pin came back empty. The leftover that survived is distro FTPS: `VSF_BUILD_SSL` plus `ssl_enable=YES` plus standalone `listen=YES`. That is the common package, not the stock `#undef`.

Same pin, two other leftovers I am not mixing into this post: `cmds_denied` misses RFC aliases, and brace `deny_file` misses children. This write-up is the FTPS ticket bind.

03 Pinning the official tarball, not a GitHub mirror

vsftpd does not live on GitHub. Chris Evans ships it from security.appspot.com/vsftpd.html. GitHub is full of mirrors (`mamh-mixed/vsftpd-3.0.5` and friends). Distros carry their own patch series. I pinned the official 3.0.5 tarball, SHA-256 `26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3`, and grepped `VSF_VERSION "3.0.5"` before `make`. August 2021 GA. GPL-2.0-only WITH vsftpd-openssl-exception.

Debian and Fedora trees are other pins. CVE-2025-14242 is a Red Hat Is parser that is not in this tarball. Debian later capped `SSL_CTX_set_timeout` at 7 days for RFC 8446. That is a ticket-lifetime patch, not a session bind, and it is not this tree.

Stock `builddefs.h`:

```c
#undef VSF_BUILD_TCPWRAPPERS
#define VSF_BUILD_PAM
#undef VSF_BUILD_SSL
```

`README.ssl` is explicit: SSL is off at compile time and at runtime until you flip both. Distro packages typically `#define VSF_BUILD_SSL` and ship a PEM path. Runtime `ssl_enable` still defaults NO. A leftover that needs `ssl_enable=YES` is config-gated. KEEP only if it is High when SSL is on. Data-channel impersonation is High when SSL is on.

Standalone `listen=YES` is the compiled default (`tunables.c`) and the sample `EXAMPLE/INTERNET_SITE_NOINETD/vsftpd.conf`. inetd-per-connection does not share ticket keys this way. This leftover is the standalone listener.

Intake is scarybeasts@gmail.com. Unpaid. No SECURITY.md. No bounty.

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. 3.0.5 already closed or publicly described the High classes this hunt was looking for:

- CVE-2021-3618 ALPACA. TLS application-layer protocol confusion, HTTP on an FTP port. 3.0.4 already rejects HTTP verbs pre-login (`GET` / `POST` / `HEAD` / `OPTIONS` / `CONNECT`), closes after 10 unknown pre-login commands, rejects a non-`ftp` ALPN when ALPN is offered, and adds optional

`ssl_sni_hostname`. 3.0.5 then fixed the ALPN *select* so FileZilla 3.55's `ftp` slice is chosen. No new unmatched HTTP/SMTP verb leftover that skips auth. Skip ALPACA.

- Tim Kosse, 3.0.3, kill-on-reuse-**fail**. Changelog: "Kill the FTP session if we see session re-use failure." `ftpdataio.c` `vsf_ftpdataio_post_mark_connect` calls `vsf_cmdio_write_exit` with `522 SSL connection failed: session reuse required` when `require_ssl_reuse` and the data handshake returns 0. That is the same-control inject-then-resume from ftps_connection_stealing_exploit.cpp. The lab negative is still 522. The KEEP below is a different bind miss.

- 3.0.4 disabled TLS prior to 1.2 by default. `tlsv1` / `tlsv1_1` default 0; `tlsv1_2` / `tlsv1_3` default 1.

- CVE-2025-14242 RH STAT integer overflow. Medium 6.5. Vulnerable code is a Fedora/RHEL patch. Debian `<not-affected>`. Pin `ls.c` has no `[` range parser. Not this tree.

- CVE-2021-30047 connection-limit DoS. Availability.

- CVE-2015-1419 distro realpath. Public. Not in official 3.0.5. Do not re-file.

- CVE-2011-2523 malicious 2.3.4 backdoor tarball. Not this SHA-256.

- CVE-2011-0762 filter recursion. `VSFTP_MATCHITERS_MAX` 1000 is in tree.

- Documented tunables: `pasv_promiscuous` / `port_promiscuous`, `require_ssl_reuse=NO`, `ssl_enable` itself, `allow_writeable_chroot`, `http_enable`.

- Firmware `vsftpd.conf` CVEs (`chroot_local_user=YES` on TOTOLink / Linksys / TRENDnet). Operator config of other products.

Two public tickets sit next to this leftover and are not it.

Kosse's 2015 write-up already said the 3.0.3 kill **requires ticket lifetime bound to the control connection**, and that "a server that uses a single long-lived TLS context cannot use this solution." The other solution he named was "verify that the session of the control connection is the same as the session of the data connection." vsftpd 3.0.3 implemented the kill. 3.0.5 still has one `SSL_CTX` in the standalone listener, `SSL_CTX_set_timeout(p_ctx, INT_MAX)`, and no `SSL_SESSION` compare. That sentence from 2015 is this leftover.

ALPACA is "HTTP client talks to FTP server." This leftover is "FTPS client resumes the wrong ticket on a PASV." Different protocol confusion.

Those stay on the skip list.

05 Nineteen surfaces, one FTPS leftover

I split the tree. 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.

Pre-login table is USER/PASS/QUIT/FEAT/OPTS/AUTH/PBSZ/PROT. HTTP verbs kill the session. Ten unknown pre-login commands close. Command line is 4096 bytes, `str_netfd_alloc` peeks to the first LF and consumes only that. PORT/EPRT reject a foreign IP and ports `< 1024` unless `port_promiscuous`. PASV accept must match the control IP unless `pasv_promiscuous`. Parent `GET_DATA_SOCK` connects `p_remote_addr`. `user_config_dir` loads after PASS, ignores USER with `/`, wants a regular file owned by the daemon uid. ASCII SIZE `stat`s `S_ISREG` and reports raw `st_size`; the sample conf already documents this as safe. Seccomp on x86_64 has

no `execve` / `ptrace` / `mount` / `clone` for the dropped child. Writeable chroot deny default on. SITE CHMOD `& 0777`.

Default core FTP unpublished Crit/High: none.

SSL is where the temperature changed. AUTH TLS does not keep a userland command leftover. CCC is not implemented. PROT C is blocked for local users by `force_local_data_ssl` (default YES). sslslave is the unprivileged pre-login child, not the privileged parent. SNI does not switch vhosts. `ssl_verify_callback` returns 1 unless `validate_cert`. Peer-cert SHA-256 matching runs only when the control handshake stored a digest; `require_cert` / `validate_cert` default NO, so typical clients present nothing.

Then I opened `ssl_accept` next to the man page.

06 Data SSL has no USER. The bind is a boolean.

`ssl_accept` (`ssl.c:474-525`) documents the protocol gap and then fails to close it:

C Copy

```c
int
ssl_accept(struct vsf_session* p_sess, int fd)
{
  /* SECURITY: data SSL connections don't have any auth on them as part of the
   * protocol. If a client sends an unfortunately optional client cert then
   * we can check for a match between the control and data connections.
   */
  SSL* p_ssl;
  int reused;
  /* ... SSL_accept via get_ssl ... */
  reused = SSL_session_reused(p_ssl);
  if (tunable_require_ssl_reuse && !reused)
  {
    str_alloc_text(&debug_str, "No SSL session reuse on data channel.");
    vsf_log_line(p_sess, kVSFLogEntryDebug, &debug_str);
    ssl_data_close(p_sess);
    return 0;
```

```
  }
  if (str_getlen(&p_sess->control_cert_digest) > 0)
  {
    /* SHA-256 compare of presented certs */
  }
  return 1;
}
```

There is no `SSL_get_session` on the control SSL. There is no `SSL_SESSION` compare. There is no TLS 1.3 ticket appdata pin. The cert digest path is the "unfortunately optional" second bind, and it is empty for every client that did not present a cert.

`sslslave.c:52-64` calls the same `ssl_accept`. After SSL login the pre-login child becomes `ssl_slave` and closes the privileged parent fd. The parent is not on the SSL pair. A leftover in `ssl_accept` is a leftover in both the one-process and two-process paths.

Standalone `ssl_init` runs in the listener before `fork` (`main.c:147-150`):

C Copy

```c
  /* Initialize the SSL system here if needed - saves the overhead of each
   * child doing this itself.
   */
  if (tunable_ssl_enable)
  {
    ssl_init(&the_session);
  }
```

That is the usual OpenSSL worker-process model. Children inherit the listener `SSL_CTX`, including the session-ticket key. `ssl_init` then:

- `SSL_CTX_set_options(p_ctx, SSL_OP_ALL)` with protocol-version bits. No `SSL_OP_NO_TICKET`.
- `SSL_CTX_set_session_id_context(p_ctx, "vsftpd", ...)`.

- If `require_ssl_reuse`: `SSL_CTX_set_timeout(p_ctx, INT_MAX)` so tickets outlive the FTP session.

`SSL_session_reused()==1` means "this ClientHello carried a ticket this process can decrypt," not "this is the victim's control session."

PASV still requires the control peer IP (`privops.c:212-218`) unless `pasv_promiscuous`:

C Copy

```c
if (!tunable_pasv_promiscuous)
{
  if (!vsf_sysutil_sockaddr_addr_equal(p_sess->p_remote_addr, p_accept_addr))
  {
    vsf_sysutil_close(remote_fd);
    vsf_sysutil_sockaddr_clear(&p_accept_addr);
    return -2;
  }
}
```

Same-IP is mandatory. Shared NAT is the usual FTPS case `require_ssl_reuse` was added to stop. AC:H is that check, not a rare lab toy. Mallory who shares Alice's source IP is who the reuse bind was supposed to reject.

Alice's RETR blocks in `get_remote_transfer_fd` (`postlogin.c:1428-1454`) until PASV `accept` returns:

C Copy

```c
if (pasv_active(p_sess))
{
  remote_fd = vsf_ftpdataio_get_pasv_fd(p_sess);
}
/* ... */
vsf_cmdio_write(p_sess, FTP_DATACONN, p_status_msg);
if (vsf_ftpdataio_post_mark_connect(p_sess) != 1)
{
```

```
    vsf_ftpdataio_dispose_transfer_fd(p_sess);
    return -1;
  }
```

`vsf_two_process_get_pasv_fd` sends `PRIV_SOCK_PASV_ACCEPT` to the parent and waits. 150 is not written until someone has connected. Mallory can win the race: Alice has issued RETR, the parent is in `accept`, Mallory's SYN is the one that returns. Then `vsf_ftpdataio_post_mark_connect` runs `ssl_accept` on *that* fd. Mallory's resumed ticket is the TLS peer. Alice's control still gets 150, then the transfer, then 226.

ALPN must be `ftp` or vsftpd FATALs. The 3.0.5 callback is a sliding search for `3 'f' 't' 'p'`. Offer `ftp`. Offer nothing and the callback is not invoked (ALPACA residual; skip). Offer `ftp-data` alone and the handshake dies.

Method, then, is short:

01 Mallory connects to the same standalone vsftpd, `AUTH TLS`, finishes TLS 1.2 with ALPN `ftp`, holds the session ticket. No USER. No PASS.

02 Alice, same source IP, logs in, `PASV`, `RETR`.

03 Mallory connects to that PASV port (IP check allows him) and resumes **his** ticket.

04 `ssl_accept` sees reused, skips the empty cert digest, returns 1.

05 `ftp_read_data` / `ftp_write_data` / sslslave `DO_SSL_WRITE` move Alice's file bytes on Mallory's TLS.

inetd-per-connection `SSL_CTX` (no listener `ssl_init` share) does not inherit Mallory's ticket keys. This KEEP is standalone `listen=YES`.

ProFTPD 1.3.9d does the other thing. `mod_tls.c` `tls_accept` requires `SSL_session_reused`, then `tls_compare_session_ids` of control vs data, then TLS 1.3 ticket appdata as a pin when session IDs do not match. vsftpd stops at the boolean.

07 What I almost filed instead

ALPACA as this leftover. 3.0.4 HTTP-kill, ALPN `ftp`, optional SNI. 3.0.5 FileZilla select. Public. The data-channel bind is not an HTTP verb.

Tim Kosse 3.0.3 as this leftover. His PoC injected a **new** session on Alice's data fd, then resumed **that** session on Alice's **same** control. Kill-on-fail closes that control. The lab still prints 522 for a fresh (non-resumed) data handshake. It does not stop Mallory resuming a ticket minted on a **different** control, or on a previous killed session whose ticket still decrypts with the listener key. Kosse already wrote that down in 2015. vsftpd implemented the kill and not the compare.

AUTH TLS leftover as a command inject. `str_netfd_alloc` MSG_PEEKs, finds the first `\n`, and `read`s exactly that many bytes. Extra octets stay in the kernel socket. `handle_auth` writes `234 Proceed with negotiation` and `SSL_accept`s on the same fd. Leftover plaintext is consumed as TLS and fails; `get_ssl` `die()`s. No userland leftover is later parsed as a post-TLS FTP command. Not CVE-2011-0411-class. Closed.

CCC / PROT C as a downgrade. vsftpd does not implement RFC 4217 CCC. `force_local_data_ssl` default YES refuses clear data for local users. Anon `force_anon_data_ssl` default NO is Medium MITM on public files, below this bar.

sslslave confusing the privileged parent. sslslave commands are GET_USER_CMD, WRITE_USER_RESP, DO_SSL_{HANDSHAKE,READ,WRITE,CLOSE}. Unknown cmd `die()`s. The parent is not on the SSL socketpair. A compromised protocol child can already issue FTP as that user. Not a raise.

`require_cert` **without** `validate_cert` **as a cert bypass.** Any cert satisfies "must present." Login still needs PASS. Digest pin only fires when both sides presented. Default is request-not-require. Below bar as a login skip; it is the *missing* digest that leaves the data bind as reused-only.

Inflating a stolen RETR to RCE. Mallory decrypts one transfer. He does not get Alice's password, an interactive shell, or a directory walk beyond that transfer. Class is privilege leftover / data-channel impersonation.

Scoring it Medium because PASV is same-IP. Same-IP is the FTPS NAT case the feature exists to stop. AC:H, not a local-only toy. Confidentiality of the file Alice named, integrity of a STOR Alice named. High.

08 The lab

Official 3.0.5 built in Docker with `#define VSF_BUILD_SSL`. Loopback `127.0.0.1:18700`. PASV `18701-18708`. Compose project `vsftpd-ssl-data-any-ticket`. Witness `VSFTPD-SSL-DATA-ANY-TICKET-WITNESS`. Image greps `VSF_VERSION "3.0.5"` and the tarball SHA-256 before `make`.

`lab/Dockerfile` is the pin:

DOCKERFILE Copy

```
ENV VSFTPD_TARBALL_SHA256=26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f27

# Official pin; staged COPY is the offline fallback.
COPY vsftpd-3.0.5.tar.gz /tmp/vsftpd-staged.tar.gz
RUN wget -q -O /tmp/vsftpd-3.0.5.tar.gz \
        https://security.appspot.com/downloads/vsftpd-3.0.5.tar.gz \
    || true \
    && sha256sum -c ... \
    && tar -xzf /tmp/vsftpd-3.0.5.tar.gz -C /build \
    && grep -q 'VSF_VERSION "3.0.5"' /build/vsftpd-3.0.5/vsftpver.h

# Stock tarball has #undef VSF_BUILD_SSL; distro FTPS builds define it.
RUN sed -i 's/^#undef VSF_BUILD_SSL/#define VSF_BUILD_SSL/' builddefs.h \
```

```
    && sed -i 's/ -Werror / /' Makefile \
    && sed -i 's/-Wall -W -Wshadow/-Wall -W -Wshadow -Wno-deprecated-declaration
    && make -j"$(nproc)" \
        LIBS="`./vsf_findlibs.sh` -lpam -lcap -lssl -lcrypto -lcrypt"
```

`lab/docker-compose.yml` binds control and passive ports to loopback:

YAML Copy

```
    ports:
      - "127.0.0.1:18700:21"
      - "127.0.0.1:18701-18708:18701-18708"
```

`lab/vsftpd.conf`: `listen=YES`, `ssl_enable=YES`, `allow_anon_ssl=YES`, `force_anon_logins_ssl=YES`, `force_anon_data_ssl=YES`, `require_ssl_reuse=YES`, `require_cert=NO`, `validate_cert=NO`, `ssl_request_cert=NO`, `ssl_tlsv12=YES`, anonymous read of `/var/ftp/secret.txt`. `seccomp_sandbox=NO` and `isolate=NO` are lab-only so the container PID-1 and the published PASV range do not fight the proof; they are not the leftover.

The PoC is two TLS clients on loopback. TLS 1.2, ALPN `ftp`, tickets on. Mallory never logs in.

01 Banner `220 (vsFTPd 3.0.5)`.

02 Mallory `AUTH TLS` → `234`. Wrap. `has_ticket=True`, session id 32 bytes. No USER. No PASS.

03 Alice `AUTH TLS`, `USER ftp`, `PASS x@x`, `PBSZ 0`, `PROT P`, `TYPE I`.

04 Alice `PASV` → `227` with `127.0.0.1` and a port in `18701-18708`.

05 Alice `RETR secret.txt` on a waiter thread. Server blocks in `get_remote_transfer_fd` until accept.

06 Mallory connects to that PASV, resumes his ticket. Data TLS `reused=True`, ALPN `ftp`. 35 bytes `VSFTPD-SSL-DATA-ANY-TICKET-WITNESS\n`.

07 Alice control: `150 Opening BINARY mode data connection for secret.txt (35 bytes).` then `226 Transfer complete.`

08 Negative: new Alice, new PASV, `RETR`, fresh (non-resumed) data handshake. Data TLS `reused=False`. Alice: `150` then `522 SSL connection failed: session reuse required`.

Last run:

PLAIN TEXT Copy

```
mallory-banner 220 (vsFTPd 3.0.5)
mallory-control tls=TLSv1.2 alpn=ftp
mallory-session has_ticket=True id_len=32
pasv ip=127.0.0.1 port=18703
mallory-data tls=TLSv1.2 reused=True alpn=ftp
alice-retr 150 Opening BINARY mode data connection for secret.txt (35 bytes). 22
mallory-bytes=35 reused=True payload='VSFTPD-SSL-DATA-ANY-TICKET-WITNESS\n'
pasv ip=127.0.0.1 port=18708
fresh-data tls=TLSv1.2 reused=False
fresh-alice replies='150 Opening BINARY mode data connection for secret.txt (35
SUCCESS vsftpd-ssl-data-any-ticket reused-ticket=yes fresh-handshake=522 VSFTPD-
```

Replay:

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/vsftpd-ssl-data-any-ticket
cd vsftpd-ssl-data-any-ticket/lab
```

```
./run.sh
```

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

09 Wrong turns, in the order I made them

ALPN as optional. First ClientHello had no ALPN. vsftpd 3.0.4 rejects a non-`ftp` advertisement with `SSL_TLSEXT_ERR_ALERT_FATAL`. No advertisement means the callback is not invoked, which is the ALPACA residual and not this leftover. The FileZilla-shaped client *does* offer ALPN. Offer `ftp`. Anything else is a self-inflicted FATAL and a wasted afternoon.

Pipelining AUTH TLS leftover into the ClientHello on purpose, then calling it a leftover. `str_netfd_alloc` peeks to LF and consumes only that line. The kernel still has the extra octets. `SSL_accept` eats them as TLS and dies. That is fail-closed, and it is also why a sloppy PoC looks like "vsftpd crashed on AUTH" until you stop writing past the CRLF.

Fighting Docker SNAT. I spent a pass convinced published ports would make Alice and Mallory look like different IPs and trip `privops.c:212-218`. Userland-proxy SNAT from the host to the published range gives both of them the same source. That is the NAT case. It helps the PASV check, which is why same-IP is AC:H and also why the lab is faithful. Different-IP PASV is still `-2` / `Security: Bad IP connecting.`

Makefile `-Werror` on OpenSSL 3. Stock CFLAGS are `-Wall -W -Wshadow -Werror`. `SSL_library_init`, `SSLv23_server_method`, `EC_KEY_new_by_curve_name`, `SSL_CTX_set_tmp_ecdh` are deprecated. The binary does not exist until you drop `-Werror` and add `-Wno-deprecated-declarations`. That is a lab compile, not a pin change. `vsftpver.h` is still 3.0.5.

arm64 `vsf_findlibs.sh`. The helper looks for PAM at `/lib/x86_64-linux-gnu/libpam.so.0` and libcap at `/usr/lib/libcap.so`. Debian bookworm arm64 puts

them under `aarch64-linux-gnu`. Link fails without an explicit `-lpam -lcap -lssl -lcrypto -lcrypt`. The leftover is not a missing library. The Dockerfile names the libs so the image is the daemon.

Pinning a GitHub mirror. Convenient. Wrong tree. Official tarball or nothing.

Treating inetd as this leftover. Per-connection `SSL_CTX` does not inherit the listener ticket key. Standalone `listen=YES` does. Compiled default. Sample default. Distro unit files that fork a listener are this leftover. xinetd-per-connection is not.

Calling 522 a failed lab. 522 is the 3.0.3 kill working. The SUCCESS line requires both: reused ticket accepted, *and* fresh handshake 522. One without the other is FAIL.

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

10 What an attacker can do

Needs a vsftpd 3.0.5 (or the same `ssl_accept`) built with `VSF_BUILD_SSL`, `ssl_enable=YES`, standalone `listen=YES` so children inherit the listener `SSL_CTX`, `require_ssl_reuse=YES` (the default). Mallory must share Alice's source IP (NAT, SNAT, same host). Mallory needs `AUTH TLS` against that listener. He does not need Alice's password. He does not need USER/PASS at all. `require_cert=YES` plus a presented cert binds SHA-256 digests and closes this path; that is not the default.

No unauth login skip. No shell. No privileged-parent ops. No chroot escape. No different-IP PASV. Cleartext PROT C still blocked for local users.

PRIVILEGE	METHOD / SINK
Confidentiality: steal that RETR / LIST	Mallory `AUTH TLS` (no PASS), hold ticket; connect to Alice's PASV from

PRIVILEGE	METHOD / SINK
	the same IP; resume Mallory's ticket. `ssl_accept` `ssl.c:494` then `ftp_write_data` / sslslave `DO_SSL_WRITE`
Integrity: inject that STOR / APPE	Same `ssl_accept` gate, Alice `STOR` / `APPE` / `STOU`, `ftp_read_data` / sslslave `DO_SSL_READ`. Alice's named file is written from Mallory's TLS plaintext, as Alice's uid / anon ftp uid, inside the existing chroot
Availability: Alice's data TLS peer is Mallory	Alice's client does not complete the data handshake (wrong peer). Server still 226s the transfer Mallory finished
Availability: kill Alice's transfer with a fresh handshake	Connect to Alice's PASV and complete a non-resumed data TLS. `ssl_accept` returns 0, `vsf_ftpdataio_post_mark_connect` `vsf_cmdio_write_exit` 522
Negative: different-IP PASV	`privops.c:212-218`

PRIVILEGE	METHOD / SINK
Negative: USER/PASS skip	Pre-login table; Mallory never sent them and never got a session uid
Negative: shell / execve	Seccomp; no `execve` in the dropped child
Negative: privileged parent on the SSL pair	sslslave closes parent fd; parent only CHOWN / PORT / PASV
Negative: `require_cert` digest bind	`control_cert_digest` vs data digest in `ssl_accept`
Negative: inetd-per-connection ticket share	No listener `ssl_init` inherit
Negative: cleartext data for local users	`force_local_data_ssl` default YES
RCE inside vsftpd	none claimed

Mallory does not get Alice's password. He does not get other users' sessions on other IPs. He does not get a cert-mapped passwordless login (the feature does not

exist). He gets the bytes of the transfer Alice started, on a TLS session he minted without logging in.

11 The fix

Compare the data `SSL_SESSION` to the control `SSL_SESSION`. Cap tickets to the control connection. Both is better. ProFTPD 1.3.9d already does the compare (`tls_compare_session_ids`) and pins TLS 1.3 tickets with appdata so a resumed boolean is not enough. Kosse's 2015 note is still the spec: either bind ticket lifetime to the control connection, or verify it is the same session. vsftpd 3.0.3 implemented the kill-on-fail. 3.0.5 still mints tickets under a listener `SSL_CTX` with `INT_MAX` timeout and checks `SSL_session_reused()` only.

Until then: do not treat `require_ssl_reuse=YES` as a same-session proof. It is a "some ticket this process can decrypt" proof. `require_cert=YES` plus a presented client cert is the digest bind that actually ties data to that control. Turning `require_ssl_reuse` off removes the 522 kill and does not add a compare. inetd-per-connection avoids ticket-key inheritance and is not how the sample standalone listener runs.

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

12 References

- Proof of concept: abraxas/vsftpd-ssl-data-any-ticket · vsftpd-ssl-data-any-ticket-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-345 · CWE-287
- Tree: official vsftpd 3.0.5 (security.appspot.com/vsftpd.html · vsftpd-3.0.5.tar.gz, SHA-256 `26b602ae454b0ba6d99ef44a09b6b9e0dfa7f67228106736df1f278c70bc91d3`) · `ssl.c` `ssl_accept` / `ssl_init` · `main.c` listener `ssl_init` · `ftpdataio.c` `vsf_ftpdataio_post_mark_connect` · `privops.c` PASV same-IP · `sslslave.c` `PRIV_SOCK_DO_SSL_HANDSHAKE` · `postlogin.c` `get_remote_transfer_fd` · `vsftpd.conf.5` `require_ssl_reuse`
- Nearby public (not this leftover): CVE-2021-3618 ALPACA · Tim Kosse FTPS connection stealing (3.0.3 kill-on-fail) · Chris Evans vsftpd 2.1.0 `require_ssl_reuse` · CVE-2025-14242 RH STAT (not in this tarball)
- Compare: ProFTPD 1.3.9d `contrib/mod_tls.c` `tls_compare_session_ids` / ticket appdata pin
- Intake: scarybeasts@gmail.com

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