The Surreal Horror of PAM

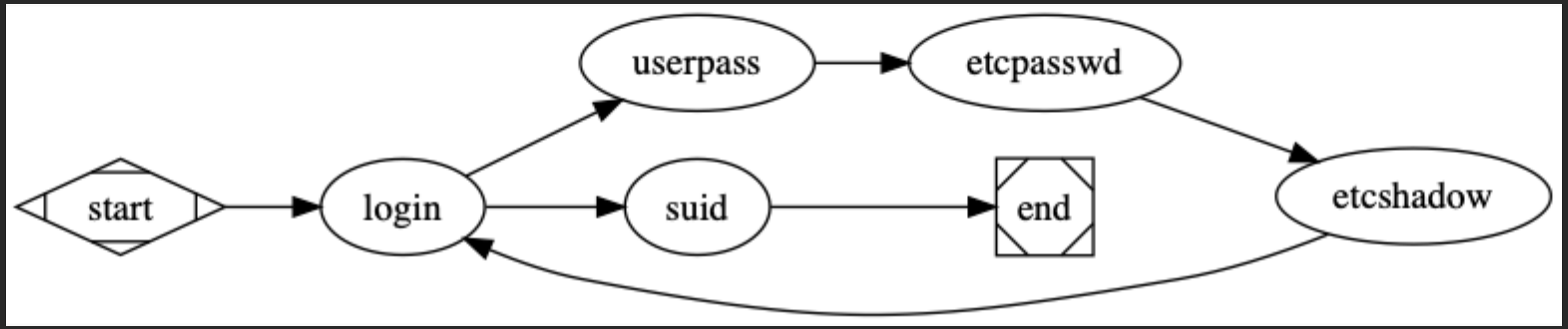
AKA: The Tragedy of Authentication

Xe

Tailscale Virtual Onsite Fall 2021

The UNIX authn/authz Flow

- Start at a login program
- Get username and password
- Check it against /etc/passwd
- Check it against /etc/shadow
- Login calls setuid(2)
- Login shell created and end



Pros/Cons for UNIX Auth

- It is exceedingly simple
- It just relies on text files
- It has been around since the dawn of UNIX time

- It works on only one machine at a time
- We live in the age of hyperconverged cloud federated femtoservices

Workarounds

- Put those files on a network filesystem
  - None of the network filesystems let you have a TLS-validated session to mount them
  - Every network filesystem assumes the network is trusted
- Put them on a CD or something
  - Expensive to change passwords

SSH...
OpenSSH
KEEPING YOUR COMMUNIQUÉS SECRET
TOP SECRET

NO-STICK COOKING SPRAY
PAM
BAKING
made with FLOUR
PERFECT RELEASE
BON APPÉTIT
Highly
Recommend

What is PAM?

PAM handles authn/authz

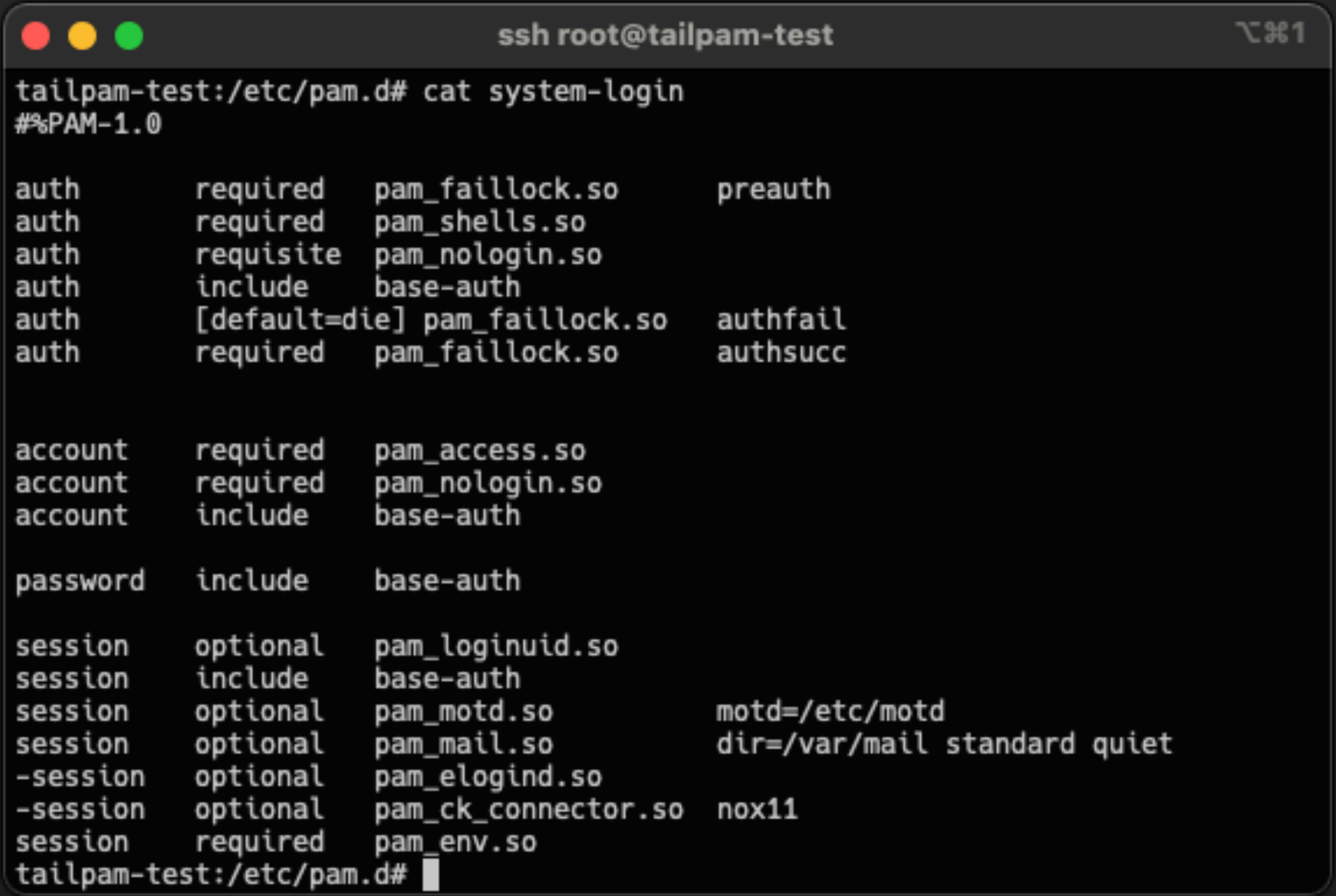
Authentication and Authorization

Who Made PAM?

sun
microsystems

<Mara> How does PAM work?

The Core of PAM

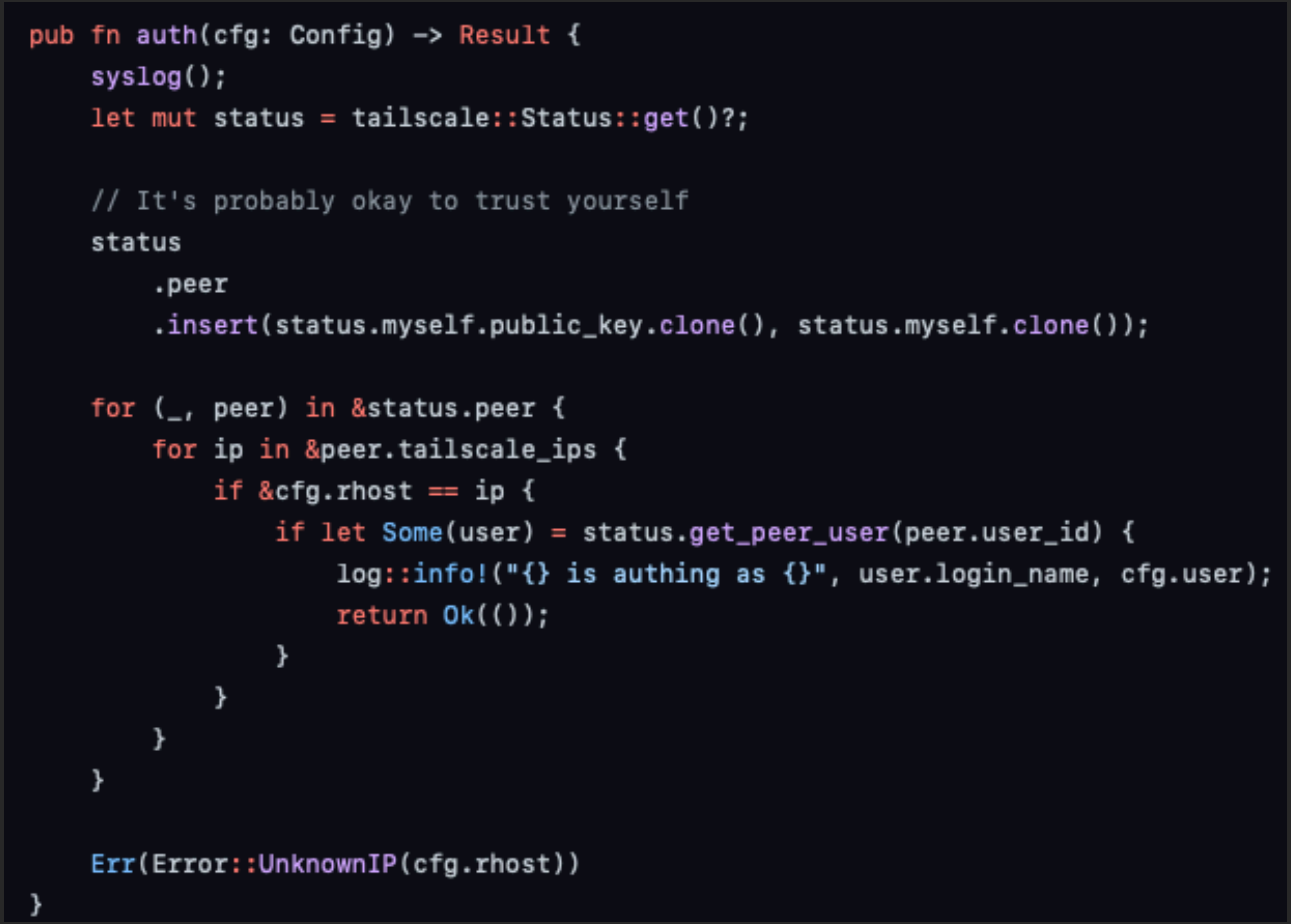


ssh root@tailpam-test ⌥⌘1

```
tailpam-test:/etc/pam.d# cat system-login
#%PAM-1.0

auth       required   pam_faillock.so      preauth
auth       required   pam_shells.so
auth       requisite  pam_nologin.so
auth       include    base-auth
auth       [default=die] pam_faillock.so   authfail
auth       required   pam_faillock.so      authsucc


account    required   pam_access.so
account    required   pam_nologin.so
account    include    base-auth

password   include    base-auth

session    optional   pam_loginuid.so
session    include    base-auth
session    optional   pam_motd.so          motd=/etc/motd
session    optional   pam_mail.so          dir=/var/mail standard quiet
-session   optional   pam_elogind.so
-session   optional   pam_ck_connector.so  nox11
session    required   pam_env.so
tailpam-test:/etc/pam.d# 
```

<Mara> How is this relevant to Tailscalars?

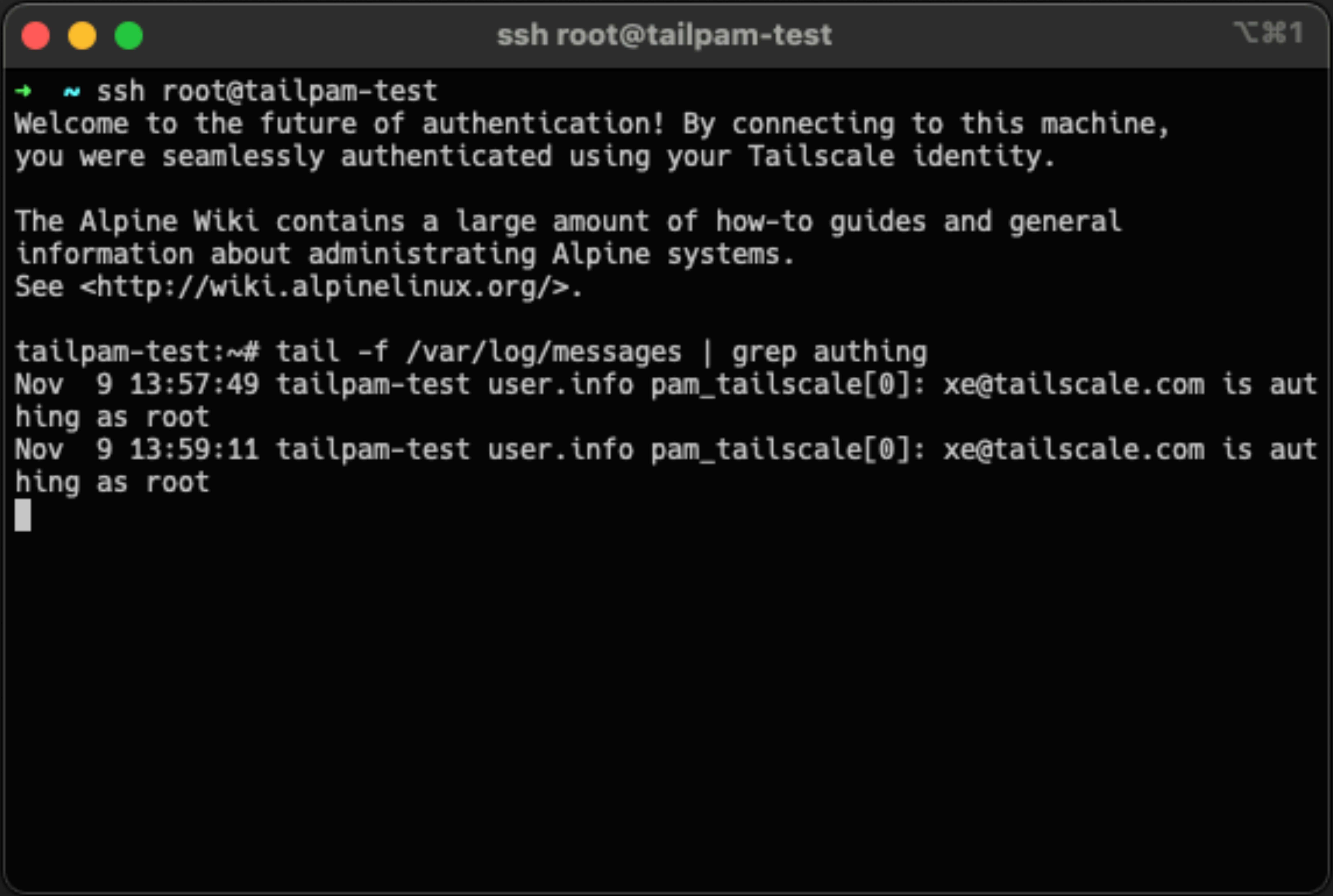
Le cœur de le module



```rust
pub fn auth(cfg: Config) -> Result {
    syslog();
    let mut status = tailscale::Status::get()?;

    // It's probably okay to trust yourself
    status
        .peer
        .insert(status.myself.public_key.clone(), status.myself.clone());

    for (_, peer) in &status.peer {
        for ip in &peer.tailscale_ips {
            if &cfg.rhost == ip {
                if let Some(user) = status.get_peer_user(peer.user_id) {
                    log::info!("{} is authing as {}", user.login_name, cfg.user);
                    return Ok(());
                }
            }
        }
    }

    Err(Error::UnknownIP(cfg.rhost))
}
```

Live Demo

```
ssh root@tailpam-test
```



```
➜  ~ ssh root@tailpam-test
Welcome to the future of authentication! By connecting to this machine,
you were seamlessly authenticated using your Tailscale identity.

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

tailpam-test:~# tail -f /var/log/messages | grep authing
Nov  9 13:57:49 tailpam-test user.info pam_tailscale[0]: xe@tailscale.com is aut
hing as root
Nov  9 13:59:11 tailpam-test user.info pam_tailscale[0]: xe@tailscale.com is aut
hing as root
```

<Mara> How do you debug PAM?

<Mara> What about bsd_auth or factotum?