%title: systemd-homed @ ChicagoLUG %author: Jim Campbell (jcampbell@gnome.org) %date: 2020-01-25

-> # A LOOK AT SYSTEMD-HOMED <- ^

-> by Jim Campbell (jcampbell@gnome.org) <- -> at <- -> ## ChicagoLUG - January 25, 2020 <- -> ## Pumping Station: One <-

-> https://jimcampbell.org <-

-> # Overview <-

1. Current Home directory configuration issues
2. Other (non-config)issues with Home directories
3. What problems systemd-homed aims to solve
4. Components of systemd-homed
5. Using it / not using it

-> # Current Home Directory Configuration <- ^

-> ## State and configuration is scattered throughtout the filesystem <- ^

Some user data is stored in `/etc/passwd`:

```
jimsaccount:x:1000:1000:Jim Campbell:/home/jimsaccount:/bin/bash
```

-> # Current Home Directory Configuration (cont.) <-

-> ... more user configuration data is stored in `/etc/samba`: <-

```
[global] workgroup = SAMBA security = user passdb backend = tdbsam printing = cups printcap name = cups load printers = yes [homes] comment = Home Directories valid users = %S, %D%w%S browseable = No read only = No inherit acls = Yes
```

-> # Current Home Directory Configuration (cont.) <-

-> Public SSH keys are stored in `/home/jimsaccount/.ssh/`: <-

```
[jimsaccount@ohokay ~]$ ls -1 .ssh/ id_ed25519 id_ed25519.pub id_rsa id_rsa.pub known_hosts
```

-> # Current Home Directory Configuration (cont.) <-

-> User resource restrictions are stored in `pam_limits` <- -> (i.e., `/etc/security/limits.conf` & `/etc/security/limits.d/foo.conf`): <-

```
# /etc/security/limits.conf
#
# This file sets the resource limits for the users logged in via PAM.
# It does not affect resource limits of the system services.
#
# Also note that configuration files in
```

/etc/security/limits.d directory,

which are read in alphabetical order, override the settings in this

file in case the domain is the same or more specific.

That means for example that setting a limit for wildcard domain here

can be overriden with a wildcard setting in a config file in the

subdirectory, but a user specific setting here can be overriden only

with a user specific setting in the subdirectory.

#

Each line describes a limit for a user in the form:

```

---

-> # Other Issues with Home Directories <- ^

1. Not encrypted on suspend
2. Not portable - You can't easily take your home dir setup from one machine to another
3. Not extensible - Can't easily store add'l metadata

---

-> # What systemd-homed aims for <- ^

1. Consolidate user home directory metadata
2. Encryption on suspend
3. Home directories can ultimately be portable (for those who want that)
4. Extensible home directory metadata
5. Yubikeys as first-class citizens

---

-> # Components of systemd-homed <- ^

- Metadata stored in `~/.identity` file ^
- Several systemd-homed components to manage old and new style home dirs: ^
  - `systemd-homed.service`: Manages home dirs & embeds JSON records in the home dir images. ^
  - `pam-systemd`: Processes the JSON records & works w/ systemd-logind to set appropriate session configs ^
  - `systemd-logind.service`: Also parses the JSON records & sets-up the session ^
  - `nss-systemd`: Synthesizes classic NSS records from JSON for backwards compatibility ^
  - `systemd-userdb.service`: Translates old-skool NSS records to JSON records.
    - Provides VARLINK API. ^
  - `VARLINK API` to query and enumerate old style records & convert them to new style ^
```

    - `homectl` command-line application

-> # More components of systemd-homed <- ^

```
* Code to handle different home dir storage mechanisms:
    * Plain directory / btrfs subvolume
    * Encrypted `fscrypt` directories
    * cifs home directories
    * luks home directories
    ^
```

- Mounting of home dirs will be done as: ^
    - bind mounts (for plain, subvolume or fscrypt),
    - cifs mount for cifs network mount
    - mounting of a block device which contains the LUKS2 image ^

* "The directories become inaccessible under their regular path the instant they are deactivated"

^

More information is available in the systemd-homed documentation.

-> # A look at the ~/.identity file <- ^

- It's JSON. ^
    - A lot of programming languages can parse it
    - Popular on the web
        - Can be linked to other services (eventually) ^
- Can hold extra stuff that isn't available in /etc/passwd: ^
    - Biometric / Yubikey info
    - Picture, email address, preferred location or time zone
    - Resource management settings (CPU/IO weights, resource limits, etc.)
    - Runtime parameters such as env variables (e.g., nodev, noexec, etc.)
    - Info about where to mount the home dir from ^
- It (can) contain different sections: ^
    - regular, privileged, perMachine, binding, status, signature & secret

---

-> # An initial look at the ~/.identity file (cont.) <- ^

The various sections

- `regular`: Fields that apply unconditionally in all context. Not security sensitive. ^
- `privileged`: Security sensitive fields. Similar to `/etc/shadow` ^
- `perMachine`: "If you're on this machine (UUID), you should only get 1GB of RAM" ^
- `binding`: Can include details on special UID or path assignments on that particular host.
    - I don't fully groc how this is different from `perMachine`.

-> # An initial look at the ~/.identity file (cont.) <- ^

More of the sections

- `status`: Augmented during runtime & never persisted to disk. Can include current resource usage (for example: currently used disk space of the user) ^
- `signature`: Contains one or more cryptographic signatures of a reduced version of the user record. ^
- `secret`: Contains secret user credentials, such as password or PIN information. This data is never persisted, and never returned when user records are inquired by a client, privileged or not. ^

Full details are here.

-> ## An Example Record <-

-> # LET'S TAKE A BREAK TO LOOK AT SOME JSON <-

-> # Using it / not using it <- ^

- Largely targeted towards laptops and (a bit toward?) workstation / desktop users ^
- Not as helpful for servers ^
- Handling of SSH logins seems ... awkward. ^
- Being mathematically locked-out of a home dir is a feature, not a bug
- Enterprise workstations? How can you SSH-in to help a user w/ a borked homedir? Unclear.

-> ## More information about systemd-homed <-

- systemd-homed documentation
- JSON User Records
- JSON Group Records
- 2019 All Systems Go introductory talk on systemd-homed ^

And ^

- MDP Presentation Software

-> # THANK YOU <-