Reinventing Home Directories
All Systems Go! 2019

September 2019
$HOME
$HOME

~
/etc/passwd
Problems

- Needs writable /etc
- Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or "mismatching" encryption)
- No modern authentication mechanisms
- Not extensible; plenty "Sidecar" databases (/etc/shadow, accounts-daemon, samba, SSH, pam, limits)
- No resource management

Reinventing Home Directories
Problems

- Needs writable /etc
Problems

- Needs writable /etc + Mixes State and Configuration
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or “mismatching” encryption)
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or “mismatching” encryption)
- No modern authentication mechanisms
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or “mismatching” encryption)
- No modern authentication mechanisms
- Not extensible
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or “mismatching” encryption)
- No modern authentication mechanisms
- Not extensible; plenty “Sidecar” databases (/etc/shadow, accounts-daemon, samba, SSH, pam_limits)
Problems

- Needs writable /etc + Mixes State and Configuration
- UID assignments need to be propagated between systems
- No encryption (Or “mismatching” encryption)
- No modern authentication mechanisms
- Not extensible; plenty “Sidecar” databases (/etc/shadow, accounts-daemon, samba, SSH, pam_limits)
- No resource management
Focus
Focus

Human ("real", "regular" users)
Focus

Human (“real”, “regular” users)

particularly Laptop users
Goals
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)

UID assignments as local artifact
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)

UID assignments as local artifact

Unification of User Password and Encryption Key
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)

UID assignments as local artifact

Unification of User Password and Encryption Key

Extensible User Records
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)

UID assignments as local artifact

Unification of User Password and Encryption Key

Extensible User Records

Lock LUKS on System Suspend
Goals

Migratable Home Directories (all the way to the point of “home-on-a-stick”)

Self Contained Home Directories (i.e. mere existence of a file /home/foobar.home synthesizes a user foobar)

UID assignments as local artifact

Unification of User Password and Encryption Key

Extensible User Records

Lock LUKS on System Suspend

Yubikeys, from day #1
Complications
Complications

SSH Logins
Complications

SSH Logins

Disk Space Assignments
Complications

SSH Logins

Disk Space Assignments

UID Assignments (chown() on login)
Complications

SSH Logins

Disk Space Assignments

UID Assignments (chown() on login)

LUKS Locking
Two new Concepts
Concept A:

JSON Users Records

(Superset of NSS records: struct passwd + struct group)

Queriable via Varlink interface

Convertible forth and back (lossy though, necessarily)
Concept A:

JSON User Records
Concept A:

JSON User Records

(Superset of NSS records: struct passwd + struct group)
Concept A:

JSON User Records

(Superset of NSS records: struct passwd + struct group)

Queriable via Varlink interface
Concept A:

JSON User Records

(Superset of NSS records: `struct passwd + struct group`)

Queriable via Varlink interface

Convertible forth and back (lossy though, necessarily)
Example:

```json
{
    "userName" : "grobie",
    "disposition" : "regular",
    "lastChangeUsec" : 1565950024279735,
    "niceLevel" : 5,
    "memberOf" : [
        "wheel"
    ],
    "binding" : {
        "15e19cf24e004b949ddaac60c74aa165" : {
            "fileSystemType" : "ext4",
            "fileSystemUUID" : "758e88c8-5851-4a2a-b88f-e7474279c111",
            "gid" : 60232,
            "homeDirectory" : "/home/grobie",
            "imagePath" : "/home/grobie.home",
            "luksCipher" : "aes",
            "luksCipherMode" : "xts-plain64",
            "luksUUID" : "e63581ba-79fb-4226-b9de-1888393f7573",
            "luksVolumeKeySize" : 32,
            "partitionUUID" : "41f9ce04-c827-4b74-a981-c669f93eb4dc",
            "storage" : "luks",
            "uid" : 60232
        }
    },
    ...
}
```

Reinventing Home Directories
Example (continued):

```
...
"privileged" : {
    "hashedPassword" : [
        "\$6\$WHBKvAFFT9jKPA4k\$0PY4D4TczKN/j0nJzy54DDu00agCcvxybrwMbe1SVdm.Bbr.z0mBdATp.Qrv9ow/
    ],
},
"signature" : [
  {
    "data" : "LU/HeVrPZSzi3MJ0PVHwD5m/xf51XDYCrSpbDRNBdtF4fDVhrN0t2I20qH/1yXiBidXlV0pt5
    "key" : "-----BEGIN PUBLIC KEY-----\nMCowBQYDK2VwAyEA/QT6kQW0AMhDJf56jBmszEQQpJHq5
  }
],
}``
Concept B
Concept B

Encrypted LUKS Home Directories in Loopback Files
Concept B

Encrypted LUKS Home Directories in Loopback Files
(/home/foobar.home with JSON records in ~/.identity)
Concept B
Encrypted LUKS Home Directories in Loopback Files
(/home/foobar.home with JSON records in ~/.identity)
Managed by systemd-homed.service
Concept B

Encrypted LUKS Home Directories in Loopback Files
(/home/foobar.home with JSON records in ~/.identity)

Managed by systemd-homed.service

Concept A may be used without Concept B though
Integration with the OS
Integration with the OS

`pam_systemd` enforces per-process settings of JSON record (nice level, environment, resource limits, ...)

`systemd-logind.service` enforces per-user settings of JSON record (CPU, IO, memory limits, ...)

Reinventing Home Directories
Integration with the OS

`pam_systemd` enforces per-process settings of JSON record (nice level, environment, resource limits, …)

`systemd-logind.service` enforces per-user settings of JSON record (CPU, IO, memory limits, …)
Encrypted Home Directories in Loopback Files
Encrypted Home Directories in Loopback Files
Encrypted Home Directories on Block devices
Encrypted Home Directories in Loopback Files

Encrypted Home Directories on Block devices

Think: USB sticks with migratable home directories
systemd-homed.service has multiple backends:

Plain directories
systemd-homed.service has multiple backends:
Plain directories, btrfs subvolumes
systemd-homed.service has multiple backends:
Plain directories, btrfs subvolumes, fscrypt
systemd-homed.service has multiple backends:
Plain directories, btrfs subvolumes, fscrypt, CIFS,
systemd-homed.service has multiple backends:
Plain directories, btrfs subvolumes, fscrypt, CIFS, LUKS
systemd-homed.service has multiple backends:
Plain directories, btrfs subvolumes, fscrypt, CIFS, LUKS
Quota, resize, password, encryption, ... all supported
Why LUKS:
Why LUKS:
Fully secure
Why LUKS:
  Fully secure
  Fully featured
Why LUKS:
- Fully secure
- Fully featured
- Industry Standard
LUKS Format

Reinventing Home Directories
LUKS Format

Loopback File →

Reinventing Home Directories
LUKS Format

Loopback File $\rightarrow$ GPT Partition Table $\rightarrow$

All labels: username (Yes, you can just use simple tools to access these home directories)
LUKS Format

Loopback File $\rightarrow$ GPT Partition Table $\rightarrow$ LUKS Volume with embedded Identity $\rightarrow$ All labels: user name (Yes, you can just use simple tools to access these home directories)

Reinventing Home Directories
LUKS Format

Loopback File → GPT Partition Table → LUKS Volume with embedded Identity → ext4/xfs/btrfs File System →

Reinventing Home Directories
LUKS Format

Loopback File → GPT Partition Table → LUKS Volume with embedded Identity → ext4/xfs/btrfs File System → Subdirectory
LUKS Format

Loopback File → GPT Partition Table → LUKS Volume with embedded Identity → ext4/xfs/btrfs File System → Subdirectory → .identity
LUKS Format

Loopback File → GPT Partition Table → LUKS Volume with embedded Identity → ext4/xfs/btrfs File System → Subdirectory → .identity

All labels: user name
LUKS Format

Loopback File → GPT Partition Table → LUKS Volume with embedded Identity → ext4/xfs/btrfs File System → Subdirectory → .identity

All labels: user name

(Yes, you can just use simple tools to access these home directories)
Activation

Superficial Validation →
Activation

Superficial Validation → LUKS Volume Activation/Identity Validation →

Reinventing Home Directories
Activation

Superficial Validation $\rightarrow$ LUKS Volume Activation/Identity Validation $\rightarrow$ `fsck` $\rightarrow$
Activation

Superficial Validation $\rightarrow$ LUKS Volume Activation/Identity Validation $\rightarrow$ fsck $\rightarrow$ chown $\rightarrow$
Activation

Superficial Validation $\rightarrow$ LUKS Volume Activation/Identity Validation $\rightarrow$ fsck $\rightarrow$ chown $\rightarrow$ mount $\rightarrow$
Activation
Superficial Validation $\rightarrow$ LUKS Volume Activation/Identity Validation $\rightarrow$ \texttt{fsck} $\rightarrow$ \texttt{chown} $\rightarrow$ mount $\rightarrow$ 2nd Identity Validation
Signing User Records
Signing User Records
Not an option, but implied default
Signing User Records
Not an option, but implied default
Control who can log in where
Signing User Records
Not an option, but implied default
Control who can log in where
Automatic update propagation
PKCS#11 support
PKCS#11 support
i.e. authenticate and decrypt with Yubikey
PKCS#11 support

i.e. authenticate and decrypt with Yubikey

(done properly: use any token that can store PK key pairs and decrypt, and decrypt LUKS key on the card)
User interface
User interface

homectl create foobar
User interface

`homectl create foobar`

`homectl activate foobar`
User interface

`homectl create foobar`
`homectl activate foobar`
`homectl resize foobar 3G`
User interface

homectl create foobar
homectl activate foobar
homectl resize foobar 3G
homectl passwd foobar
User interface
User interface

userdbctl user ...
User interface

userdbctl user ...

userdbctl group ...
Services
Services

systemd-homed.service
Services

systemd-homed.service

systemd-userdbd.service
Why Varlink?
Why Varlink? https://varlink.org/
Why Varlink? https://varlink.org/

Early Boot + JSON + Streamable
What about LDAP?
What about LDAP?

Integration with 3rd party JSON record providers: you get resource management for free
What about LDAP?

Integration with 3rd party JSON record providers: you get resource management for free

Simple: no C API, nothing: just bind a socket in
/run/systemd/userdb/
What about LDAP?

Integration with 3rd party JSON record providers: you get resource management for free

Simple: no C API, nothing: just bind a socket in
/run/systemd/userdb/

NSS for free
What about LDAP?

Integration with 3rd party JSON record providers: you get resource management for free

Simple: no C API, nothing: just bind a socket in /run/systemd/userdb/  
NSS for free

All services linked there are asked in parallel, first successful reply wins
Where?
Where?

https://github.com/poettering/systemd/tree/homed

When?
Where?

https://github.com/poettering/systemd/tree/homed

When?

Hopefully 244, maybe 245
Summary
Summary

Secure, Next Generation Home Directories
Summary
Secure, Next Generation Home Directories
Resource Management
Summary

Secure, Next Generation Home Directories

Resource Management

Extensible
Summary
Secure, Next Generation Home Directories
Resource Management
Extensible
Lock on Suspend
Summary

Secure, Next Generation Home Directories

Resource Management

Extensible

Lock on Suspend

Yubikeys 1st class citizens
That’s all, folks!