redhat.

# User Session Recording

Thorsten Scherf
Red Hat

SLAC 2019, Berlin - Germany

# Why?

# There is a demand from customers

Customers have been telling us for a long time that they need:

- to comply with government regulations
- to track what contractors do on their systems
- to know who broke their server and how

# And a dream

What people and governments want:

- Record everything users do
- Store that somewhere safe
- Let us find who did that thing
- Show us how they did it

# There is commercial supply

A great number of commercial offerings:

- From application-level proxies on dedicated hardware
- To user-space processes on the target system
- Recording keystrokes, display, commands, apps, URLs, etc.
- Integrated with identity management, and access control
- With central storage, searching, and playback

# But not good enough

Still people are not satisfied:

- Expensive
- Sometimes very expensive
- Can’t fix it yourself
- Can’t improve it yourself

# What can be better?

Customers want:

- Free (as in Beer)
- Open-Source
- Support

# Wait, we already have those solutions...

Nope, not really:

- script(1) plus duct tape
  - popular, but not security-oriented, needs lots of DIY
- sudo(8) I/O logging
  - security-oriented, has searching, but not centralized
- TTY audit with auditd(8)
  - security-oriented, can be centralized, but only for input
- asciinema / tmate
  - mostly for sharing session and not security-oriented

# So what do we really need?

Hottest features requested:

- Record what the user enters, sees on the screen, executes, accesses
- Get it off the machine ASAP, and store centrally and securely
- Search, analyze, and correlate with other events
- Playback in real time, or later
- Control centrally

# What do we have?
# &
# How does it work?

# Packages we ship in RHEL and Fedora

We provide:

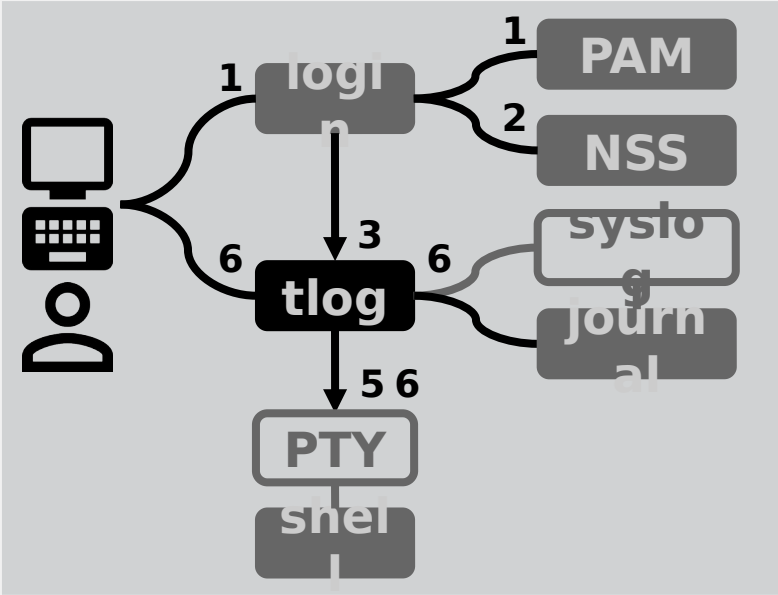
- tlog
  - A shim between the terminal and the shell
  - Converts what passes in between to searchable JSON
  - Logs to a file, syslog or journal
  - Plays back recordings on a terminal
- Cockpit-session-recording (Fedora 31)
  - JavaScript based player
  - Configuration through SSSD

# How tlog works?

Console login example

Starting a console session:

1. User authenticates to **login** via **PAM**
2. **NSS** tells **login**: **tlog** is the shell
3. **login** starts **tlog**
4. Env/config tell **tlog** the actual shell
5. **tlog** starts the actual shell in a **PTY**
6. **tlog** logs everything passing between its **terminal** and the **PTY**, via **syslog(3)** or **sd-journal(3)**

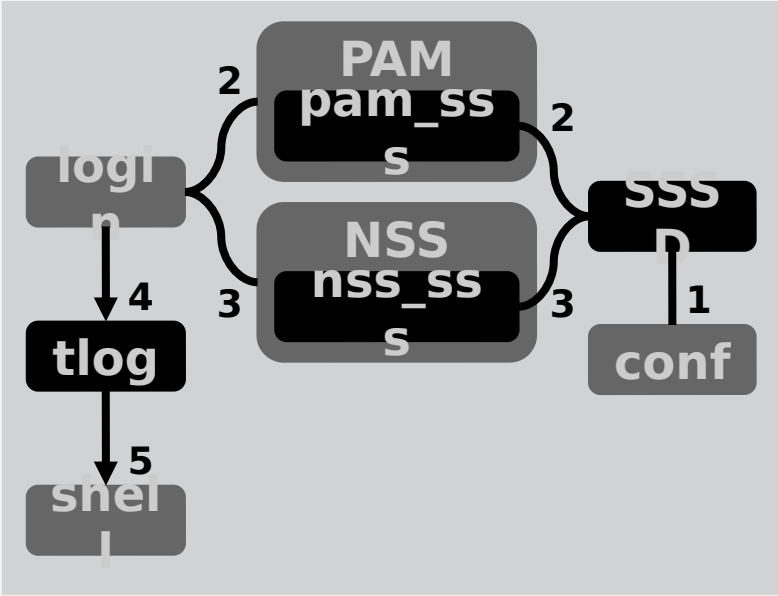


# Control tlog with SSSD

Console login example

When a recorded user logs in:

1. **SSSD** finds a match for the user in its **configuration**
2. **pam_sss** stores the actual user **shell** in the **PAM** environment
3. **nss_sss** tells **login**: **tlog** is the shell
4. **login** starts **tlog** with **PAM** environment
5. **tlog** starts the actual user **shell** retrieved from environment



# Tlog schema

Optimized for streaming and searching:

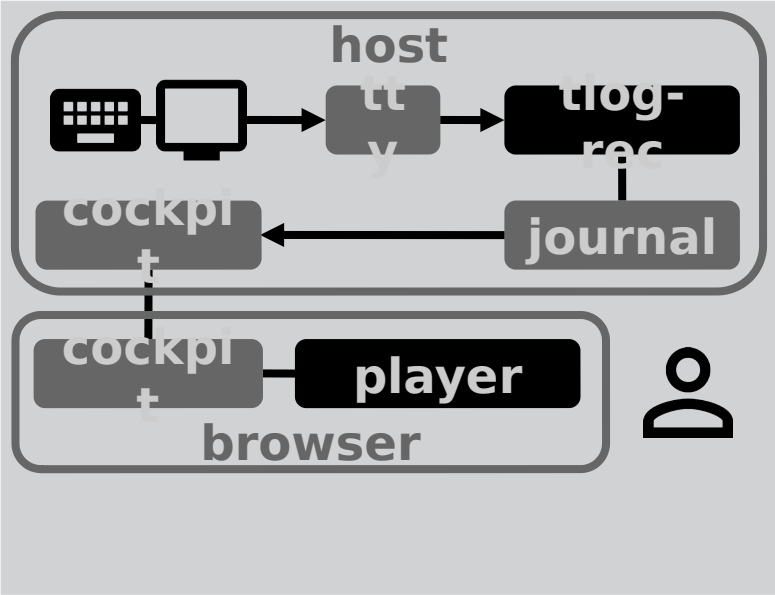
- Chopped into messages for streaming, which can be merged
- Input and output stored separately
- All I/O preserved
- Timing separate, ms precision
- Invalid UTF-8 stored separately
- Window resizes preserved

```
{
    "ver"       : "2.2",
    "host"      : "tlog-client.example.com",
    "rec"       : "c8aa248c81264f5d98d1..."
    "user"      : "user1",
    "term"      : "xterm",
    "session"   : 23,
    "id"        : 1,
    "pos"       : 0,
    "timing"    : "=56x22+98>23",
    "in_txt"    : "",
    "in_bin"    : [ ],
    "out_txt"   : "[user1@tlog-client ~]$ ",
    "out_bin"   : [ ]
}
```

# How UI in Cockpit works?

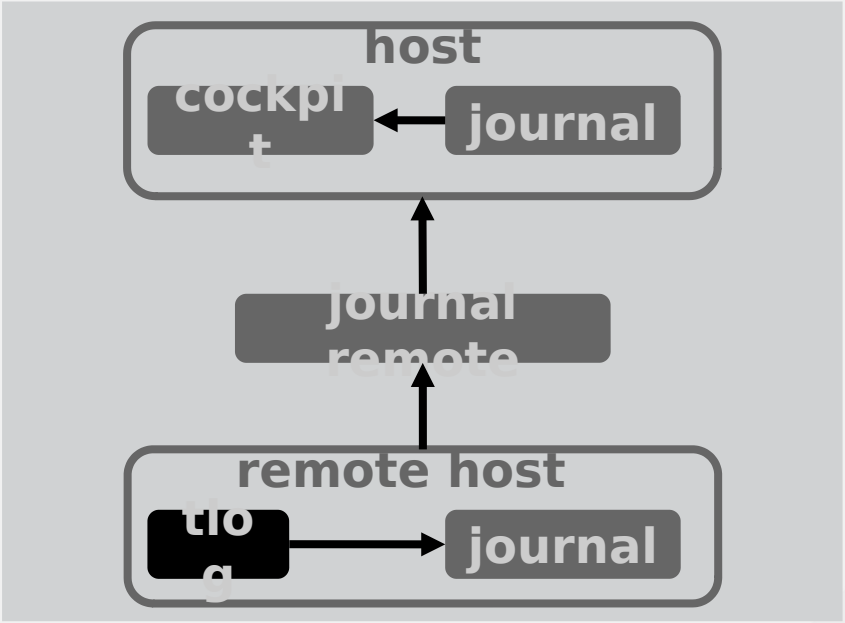
Setup for recordings in Cockpit:

- **Tlog** logs to **Journal**, adding a **recording ID** field
- To list recordings, **Cockpit** looks for **tlog** messages in Journal, groups by **recording ID**
- **JavaScript-based player** reads and plays back Journal messages with **recording ID**.

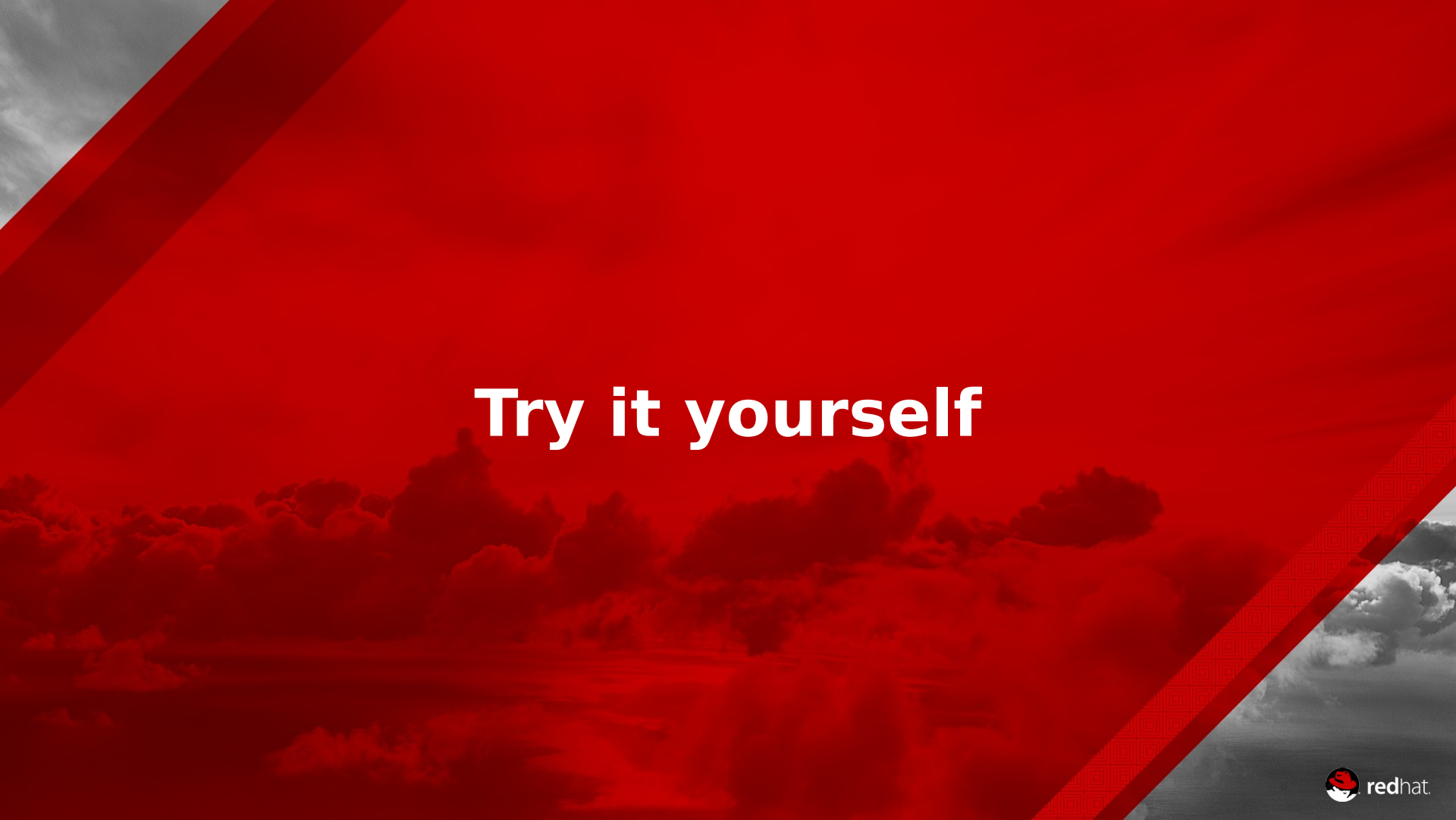


# Systemd-journal-remote

Systemd-journal-remote delivers logs & recorded sessions from other hosts to a main one for analyzing.



# Try it yourself



# As a part of RHEL

Refer to documentation, but basically:

```
# yum install tlog cockpit-session-recording
```

Or use our Ansible role:

github.com/nkinder/session-recording

# Try tlog

https://github.com/Scribery/tlog

- Download and install a release RPM, or
- Build from source
- Create a user with shell set to `/usr/bin/tlog-rec-session`
- Log to and playback from file
  - Easiest, good for testing
- Log to and playback from Journal and/or Elasticsearch
- Instructions in README.md!

# Try cockpit-session-recording

https://github.com/Scribery/cockpit-session-recording

- Install tlog
- Create a user with shell set to `/usr/bin/tlog-rec-session (or use SSSD)`
- Login as that user and do some stuff
- Checkout “Session Recording” page at http://localhost:9090

redhat.

# THANK YOU

**Terminal Session Recording Project**

http://scribery.github.io/