STPA for `ssh-agent` Wrapper: `agentrc`

Purpose of Analysis

Losses

The stakeholders are users of this script who want to simplify their `ssh-agent` workflow. They value minimal manual interaction related to SSH key management, including the ability to effectively handle forwarded agents (via a UNIX socket) and they want to minimize messy, unnecessary resources being used, e.g. multiple stale `ssh-agent` processes.

System boundary: I am considering the script that manages the `ssh-agent` process. Its function is to start a fresh `ssh-agent` process when needed, or reuse an existing and valid `SSH_AUTH_SOCKET`. This system needs to work across multiple systems that may or not may use SSH agent forwarding.

The `agentrc` in this exercise is https://github.com/nycksw/dotfiles-pub/blob/main/agentrc.

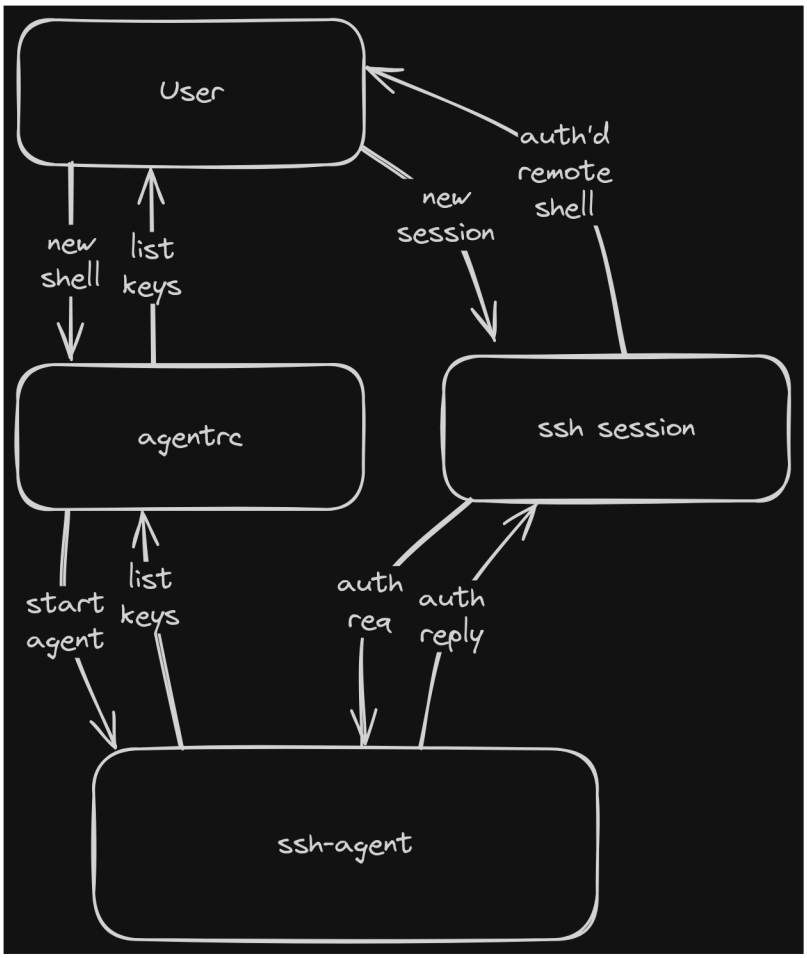
Losses:

- L-1: Loss of system function.
- L-2: Loss of user trust.

Hazards:

- H-1: Authentication functions are unavailable when required. (L-1)
- H-2: Resources are used poorly; `ssh-agent` is started when an existing one is already running. (L-2)
- H-3: User is confused by inconsistent behavior. (L-2)

Model of Control Structure



Worth noting that the main interface between the `ssh-agent` and an SSH session is a UNIX socket identified by the environmental variable `SSH_AUTH_SOCK`. This diagram abstracts that detail away, although control actions may need to be defined with that interface in mind.

Unsafe Control Actions

- UCA-1: `agentrc` doesn't launch `ssh-agent` when a new shell is created. (H-1)
- UCA-2: `agentrc` launches a redundant `ssh-agent` when a new shell is created. (H-2, H-3)
- UCA-3: `agentrc` doesn't update `SSH_AUTH_SOCK` when a new shell is created. (H-1, H-2, H-3)
- UCA-3: `agentrc` updates `SSH_AUTH_SOCK` to invalid `ssh-agent` when a new shell is created. (H-1, H-2, H-3)

Loss Scenarios

Scenario 1

A user creates a new shell session. There is no `ssh-agent` available, and `agentrc` fails to launch a new one (UCA-1). When the user tries to initiate an SSH session, they are prompted for a passphrase because there is no agent running.

Scenario 2

A user creates a new shell session. There is already an `ssh-agent` available, but `agentrc` creates a new one and prompts the user for a passphrase. The user is confused about having to re-enter the passphrase, and discovers redundant unused processes running.

Scenario 3

A user creates a new shell session. The `agentrc` script sets an `SSH_AUTH_SOCK` that points to a nonexistent or otherwise invalid `ssh-agent` process (UCA-3), and the user sees an error.

Scenario 4

A user creates a new shell session on a remote host that's configured for SSH agent-forwarding. The `agentrc` scripts creates a new `ssh-agent` process in spite of a valid `SSH_AUTH_SOCK` being available on the forwarding host (UCA-3). The user is prompted for passphrases (if keys exist on the remote host) or is left with a `ssh-agent` with no keys available.