SSH - run script or command at login

There a multiple use cases to run a script on login. Configuration, starting services, logging, sending a notification, and so on. I want to show you different ways to do so.

Example script

The example script will notify me via push notification on my smartphone as soon as a new SSH connection is established. You can use a simple command or a script, and I will use a script for this blog post.

`/path/to/script/notify-at-login.sh`

```
#!/bin/bash

# 1 - Script without output!
# IMPORTANT: Script with output break non-interactive sessions (scp, rsync, etc)

curl -d "\"$SSH_CONNECTION\" - \"$USER\" logged in" ntfy.sh/reallyecurestringfornotifications
>/dev/null 2>&1

# If you only want to run the script for an interactive SSH login and need the output
displayed, place the script right after section  2 and remove the redirect.

# 2 - Check if session is non-interactive  (remote command, rsync, scp, etc)

if [[ $SSH_ORIGINAL_COMMAND ]]; then
    eval "$SSH_ORIGINAL_COMMAND"
    exit
fi

# 3 - choose your favorite shell for the SSH session

/bin/bash
```

Remember to make it executable:

```
sudo chmod +x /path/to/script/notify-at-login.sh
```

Side note: I am using ntfy to send push notifications to my smartphone. In this example, the push notification would look this:

```
92.160.50.201 40248 195.21.0.14 22 - <user> logged in
```

Output on non-interactive connections

Just a reminder that you have to avoid any output of your script or command on non-interactive connections like rsync. Either prevent output from being displayed for non-interactive connections or all connections. The example script shows you one way to do so.

ForceCommand

I prefer this method, and had been working pretty well so far. The user will run the command and it can't really be avoided by the client.

Use the `ForceCommand` option in your `/etc/ssh/sshd_config` file to run the script:

```
ForceCommand /path/to/script/notify-at-login.sh
```

ForceCommand ignores any command or script supplied by the client and ~/.ssh/rc by default.

PAM_exec

Put the script into a new directory `/etc/pam_scripts`, set the directory's permission to `0755` and the owner and group must be `root`. The files permissions are `0700`, must be executable and the owner and group must be `root` as well.

Directory:

```
sudo mkdir /etc/pam_scripts
sudo chmod 0755 /etc/pam_scripts
sudo chown root:root /etc/pam_scripts
```

Script:

```
sudo chmod 0700 /etc/pam_scripts/notify-at-login.sh
sudo chown root:root /etc/pam_scripts/notify-at-login.sh
```

Enable `UsePAM` in the `/etc/ssh/sshd_config`:

```
UsePAM yes
```

Tell PAM to run the script at SSH login by adding the following line to `etc/pam.d/sshd`:

```
session required pam_exec.so /etc/pam_scripts/notify-at-login.sh
```

All scripts added to the `/etc/pam_scripts/` directory will be run as `root` at login.

Shell startup & sshrc file

You can run the script by your preferred startup file (`.profile` / `.bashrc`, etc) or use the SSH-specific profiles that run additionally before the user shell is loaded.

For all users:
`/etc/ssh/sshrc` *# runs only if there is no user-specific configuration file* `~/.ssh/rc`
Per user configuration in home dir:
`~/.ssh/rc`

```
~/.ssh/rc
        Commands in this file are executed by ssh when the user
        logs in, just before the user's shell (or command) is
        started.  See the sshd(8) manual page for more information.
```

Run the script via the startup file by adding the following line to it:
`. /path/to/script/notify-at-login.sh`

Both the shell startup and sshrc files will be run by the user.

Side note: if security is a concern - like a login notification - it is not recommended to use this method. Profile config files can be avoided by `ssh user@server bash --norc --noprofile` and `~/.ssh/rc` can be changed by the user after the first login.

Created 2025-06-07 23:37:26 UTC by Nicolas
Updated 2025-06-07 23:38:38 UTC by Nicolas