

GNU/Linux Administration - Feature #808

Autostart Reverse SSH Tunnel on Arch Linux

05/12/2016 04:58 PM - Daniel Curtis

Status:	Closed	Start date:	05/12/2016
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Shell Scripts	Estimated time:	0.50 hour
Target version:	Arch Linux	Spent time:	1.00 hour

Description

This is a guide on how I setup an automatic reverse SSH tunnel that connects back to a server using Arch Linux.

- Start by creating an SSH keypair; this guide uses the user **bob** (replace as necessary):

```
ssh-keygen -t ed25519
```

- Copy the key over [to the server](#) and add it to the `~/.ssh/authorized_keys` file.
- Create the systemd **tunnel-home.service** unit file:

```
sudo vi /etc/systemd/system/tunnel-home.service
```

- And add the following:

```
[Unit]
Description=Reverse SSH Tunnel Service
ConditionPathExists=/usr/bin
After=network.target

[Service]
User=bob
ExecStart=/usr/bin/ssh -NTC -o ServerAliveInterval=60 -o ExitOnForwardFailure=yes -o StrictHostKeyChecking=no -i %h/.ssh/id_ed25519 -p 10000 -R 12345:localhost:22 bob@server.example.com

# Restart every >2 seconds to avoid StartLimitInterval failure
RestartSec=3
Restart=always

[Install]
WantedBy=multi-user.target
```

NOTE: This connects to `server.example.com` as the user `bob` on port `10000`, creating port `12345` on the remote server to connect back to.

- Start and enable it at boot:

```
sudo systemctl daemon-reload
sudo systemctl enable tunnel-home.service
sudo systemctl start tunnel-home.service
```

Resources

- <http://blog.kylemanna.com/linux/2014/02/20/ssh-reverse-tunnel-on-linux-with-systemd/>
- <http://blog.philippklaus.de/2013/03/start-autossh-on-system-startup-using-systemd-on-arch-linux/>

History

#1 - 05/12/2016 05:01 PM - Daniel Curtis

- *Description updated*
- *Status changed from New to Resolved*
- *% Done changed from 0 to 100*

#2 - 07/10/2016 10:03 AM - Daniel Curtis

- *Status changed from Resolved to Closed*