Linux Tips and Tricks

- Set system time automatically
- Set correct Timezone
- Start / Stop /Restart BIND DNS Server
- Hardening of Linux
- Tutorial on ufw
- Fix Error fwupd-refresh
- Mount SAMBA shares
- Check for open ports
- Network browsing not working
- Display IP address on Panel in Xfce
- Disable SELinux on Fedora
- Create boot USB
- Install network scanner on Archlinux
- Install xrdp
- Install xrdp on Fedora 42
- Install send mail service on Fedora
- Install sendmail service on Debian
- Install xrdp on Fedora Xfce
- 10 GbE Network Tuning on Fedora
- 10 GbE Network Tuning on Debian
- Autoupdate on Debian
- Add a user to the sudoers group on Debian 13
- Verify NFS export
- .bashrc interactive shell

Set system time automatically

?Introduction

It is possibly to set and synchronize the time in Linux automatically through the systemd service. It's the successor of NTP daemon. In my network the mt-engine01.simmy.ch provides system time. Hence the device can change, I created an DNS alias ntp.simmy.ch. Using this alias allows changes of the time source without problems.

Ubuntu 22.04 LTS

nano /etc/systemd/timesyncd.conf

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the timesyncd.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# See timesyncd.conf(5) for details.

[Time]
NTP=ntp.simmy.ch
FallbackNTP=0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org
#RootDistanceMaxSec=5
#PollIntervalMinSec=32
#PollIntervalMaxSec=2048
```

```
systemctl restart systemd-timesyncd
timedatectl timesync-status

      Server: 192.168.1.74 (192.168.1.74)
Poll interval: 1min 4s (min: 32s; max 34min 8s)
         Leap: normal
      Version: 4
      Stratum: 3
    Reference: 2E8C0F6C
    Precision: 1us (-24)
Root distance: 76.324ms (max: 5s)
       Offset: +1.117ms
        Delay: 326us
       Jitter: 0
 Packet count: 1
    Frequency: -25.696ppm
```

Debian 10

https://www.digitalocean.com/community/tutorials/how-to-set-up-time-synchronization-on-debian-10

```
apt purge ntp
apt install systemd-timesyncd
nano /etc/systemd/timesyncd.conf
```

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the timesyncd.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# See timesyncd.conf(5) for details.
```

```
[Time]
NTP=ntp.simmy.ch
FallbackNTP=0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org
#RootDistanceMaxSec=5
#PollIntervalMinSec=32
#PollIntervalMaxSec=2048
```

```
systemctl start systemd-timesyncd
systemctl status systemd-timesyncd
date
```

Set correct Timezone

Howto set the correct timezone in Linux Ubuntu

Get all possible timezones:

```
timedatectl list-timezones
```

Set the local timezine:

```
timedatectl set-timezone Europe/Zurich
```

Check the local timezone:

```
timedatectl
```

Howto set the correct timezone in Debian 10

```
dpkg-reconfigure tzdata
```

Useful links

https://linuxize.com/post/how-to-set-or-change-timezone-on-ubuntu-20-04/

Start / Stop /Restart BIND DNS Server

Introduction

For testing purposes I am using Univention with bind9. The greater goal is to use AD/SAMBA from Univention. After testing for a couple of weeks suddenly some DNS addresses do not get resolved. The same problems occurred on Zentyal.

So far I couldn't find a reason for this misbehavior. However, a restart of the bind9 service seems to patch the problem.

Debian based Linux

Start the service

```
service bind9 start
```

Stop the service

```
service bind9 stop
```

Restart the service

```
service bind9 restart
```

Reload the service

This will become necessary of a configuration file is changed.

```
service bind9 reload
```

Check status

```
service bind9 status
```

Fedora based Linux

Start the service

```
systemctl start named
```

Stop the service

```
systemctl stop named
```

Restart the service

```
systemctl restart named
```

Check status

```
systemctl status named
```

Hardening of Linux

Introduction

Despite the fact that Linux is Open Source and Linux it comes as a surprise that in the default installation are some hidden trackers and spy software.

Hardening

There is a script that will remove all malware. Originally written for Linux, but it can easily adopted for other distributions.

Ubuntu Secure

This script does:

- System update and software upgrade
- Amazon & advert web apps removing
- AptUrl Removing (tool, which gives possibilities to start installation by clicking on url, can be executed with js, which is not secure)
- Guest session disable for LightDM
- Remote login disable for LightDm
- DNS encryption (dnscrypt-proxy)
  I don't recommend this, hence my DNS server is not working with encryption.
  apt -y remove dnscrypt-proxy
- FireWall (UFW)
- Antivirus (ClamAV)
- Brute Force protection (Fail2Ban)
- Basic Telemetry Removing (ZeitGeist) and unsecure libs and pre-installed software with high and potentional risks

Here is a version for rpm based systems:

```
#!/bin/bash

# This script removes telemetry and enhances system security on an RPM-based Linux
distribution.

# System Up to Date:
```

```
sudo dnf -y update
sudo dnf -y upgrade
# ========

# Remove any pre-installed telemetry or unwanted software (no direct equivalents for `unity-lens-shopping` and `unity-webapps-common` on RPM-based systems):
# Remove pre-installed software that may be tracking or unwanted:
sudo dnf -y remove gnome-online-accounts
sudo dnf -y remove gnome-shell-extension-prefs
sudo dnf -y remove gnome-software
# ========

# Disable Guest session & remote login for LightDM (if LightDM is in use):
if [ -f /etc/lightdm/lightdm.conf.d/50-no-guest.conf ]; then
    sudo sh -c 'printf "[Seat:*]\nallow-guest=false\ngreeter-show-remote-login=false\n" > /etc/lightdm/lightdm.conf.d/50-no-guest.conf'
    sudo dnf -y remove lightdm-remote-session-freerdp
    sudo dnf -y remove lightdm-remote-session-uccsconfigure
fi
# ========

# Remove any equivalent telemetry-related packages:
# Note: zeitgeist is generally specific to Ubuntu/Debian, so we focus on similar tools on RPM systems.

# Remove `tracker`, a GNOME-based file indexing and search tool that collects metadata:
sudo dnf -y remove tracker
sudo dnf -y remove tracker-miners
sudo dnf -y remove tracker3
sudo dnf -y remove tracker3-miners

# Remove `gnome-usage`, a system resource monitor that could collect usage data:
sudo dnf -y remove gnome-usage

# Remove `PackageKit`, which can send data back to package servers:
sudo dnf -y remove PackageKit
# ========

# DNS encryption:
sudo dnf -y install dnscrypt-proxy
```

```
# ========

# FireWall (using firewalld):
sudo dnf -y install firewalld
sudo systemctl start firewalld
sudo systemctl enable firewalld
sudo firewall-cmd --permanent --set-default-zone=block
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --reload
# ========

# ClamAV Antivirus Installation:
sudo dnf -y install clamav
sudo dnf -y install clamav-daemon
sudo systemctl enable clamav-daemon
sudo systemctl start clamav-daemon
# ========

# Fail2Ban installation (protects from brute force login):
sudo dnf -y install fail2ban
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
# ========

# Remove other potentially problematic or unused packages:
# Removing `cups` if you don't need printer support:
# sudo dnf -y remove cups

# Remove `remmina` if you don't use it for remote connections:
# sudo dnf -y remove remmina

# Remove unnecessary GNOME components:
sudo dnf -y remove evolution
sudo dnf -y remove evolution-data-server
sudo dnf -y remove gvfs-fuse
sudo dnf -y remove vino  # VNC server (remote desktop sharing)
sudo dnf -y remove gnome-shell-extension-background-logo  # Fedora logo on desktop background
sudo dnf -y remove gnome-user-share  # Potentially shares user data over the network
sudo dnf -y remove libreport-plugin-bugzilla  # Automatic bug reporting to Bugzilla
```

```
sudo dnf -y remove abrt-addon-xorg  # Automatic bug reporting for Xorg
sudo dnf -y remove abrt-cli  # Command-line tool for automatic bug reporting
sudo dnf -y remove abrt-addon-ccpp  # Automatic bug reporting for C/C++ programs
sudo dnf -y remove abrt-addon-kerneloops  # Automatic bug reporting for kernel oopses
sudo dnf -y remove abrt-addon-pstoreoops  # Automatic bug reporting for pstore oopses
# ========

# Autoremove unnecessary dependencies:
sudo dnf -y autoremove

# ========

# Troubleshooting:
# If the internet does not work, try restarting dnscrypt-proxy:
#   sudo systemctl restart dnscrypt-proxy
# Also, the tool may use another port, detect the port in this output:
#   sudo ss -ntulp
# Then add the port to firewalld:
#   sudo firewall-cmd --permanent --add-port=[portnumber]/tcp
#   sudo firewall-cmd --reload
# ========
```

Tutorial on ufw

UFW, or Uncomplicated Firewall, is a simplified firewall management interface that hides the complexity of lower-level packet filtering technologies such as iptables and nftables. If you’re looking to get started securing your network, and you’re not sure which tool to use, UFW may be the right choice for you.

Here is a link that shows how to set up the firewall on Ubuntu:

How To Set Up a Firewall with UFW on Ubuntu 22.04

Fix Error fwupd-refresh

Introduction

After installing monitoring (check_mk) I realized that the servis fwupd-refresh produces a critical error. However, this is based on a configuration mishap in the service itself. Here is the fix.

The service is able to perform a firmware update on UEFI machines. The service is totally useless on VMs.

Correction Step-by-Step

Edit file /lib/systemd/system/fwupd-refresh.service

Replace SuccessExitStatus=2 with SuccessExitStatus=1

Restart the service:

```
systemctl daemon-reload && sudo systemctl start fwupd-refresh.service
```

Check the service

```
systemctl status fwupd-refresh.service
```

Disable the service

Another possibility is to disable the service:

```
systemctl disable fwupd
```

Useful links

https://askubuntu.com/questions/1404691/fwupd-refresh-service-failed

https://askubuntu.com/questions/1227508/consequences-of-disabling-fwupd

Mount SAMBA shares

Introduction

There a several ways of mounting SAMBA shares on a Linux machine. This manual gives an overview.

Prerequisite

It makes things easier if the Linux client is a member of an Active Directory domain. Hence I use Zorin OS, this can easily be achieved with the correct setting during the installation:

Use Active Directory checkbox

If you want to join a Linux computer to an Active directory, please refer to:

AD Join

Mount SAMBA shares

Manual mount

```
mount -t cifs -o username=<user>,password=<secret-password> //xigma-prime.simmy.ch/backup
/mnt/backup
```

Permanent mount with fstab

In the fstab, I use the following command:

```
//xigma-prime.simmy.ch/images /mnt/images cifs
credentials=/root/.smbcredentials,uid=1000,forceuid,gid=1000,forcegid 0 0
```

This will mount the share images to the mountpoint /mnt/images. The credentials are saved in the file .smbcredentials:

```
username=<username>
password=<password in cleartext>
domain=simmy.ch
```

The file itself is placed in the home directory of root. The access right are limited to read only for the root user. So there is minimal protection for the password in clear text. Only the root user can read it.

The share(s) will be mounted during the boot process. This works most of the times, but not always.

Permanent mount with pam_mount

It is more desirable to mount the SAMBA shares when the user logs in, rather during boot.

Installation of the necessary modules

```
apt install -y libpam-mount keyutils cifs-utils smbclient
```

Configuration entry in /etc/security/pam_mount.conf.xml

The following lines have to be added to the file after the line <mkmountpoint enable="1" remove="true" />:

```
<volume
fstype="cifs"
server="xigma-prime.simmy.ch"
path="images"
mountpoint="~/mnt/images"
options="sec=krb5,cruid=%(USERUID),workgroup=SIMMY,vers=3.0" />
```

<mkmountpoint enable="1" remove="true" /> means that the mount point is created and removed automatically. Also there is no password saved on the computer. I also placed the mount point into the user home directory.

Permanent mount with GPO

It is possible to utilize GPOs to mount SAMBA shares on a Linux machine, that is joined to an Active Directory. However, I did not try this possibility.

Temporary mount

It is possible to mount a share with a file manager. This mount will be lost after log off or a reboot.

Useful resources

Check for open ports

https://phoenixnap.com/kb/linux-check-open-ports

```
lsof -nP -iTCP -sTCP:LISTEN

netstat -tunpl

ss -tunlp

nc -z -v localhost 1-65535 2>&1 | grep succeeded
```

Network browsing not working

Sometimes network browsing or the mapping of a SMB share with a file manager in Linux is not working. I found one of the most likely causes for this problem is a missing package.

After installing gvfs-smb network browsing was working fine.

Installation on Fedora

```
sudo dnf install gvfs-smb
```

Display IP address on Panel in Xfce

Create a small shell script show_ip.sh :

```
#!/bin/bash

# Get all addresses from hostname -I
IP_ADDRESSES=$(hostname -I)

# Split into individual IP addresses
IFS=' ' read -r -a IP_ADDRS <<< "$IP_ADDRESSES"

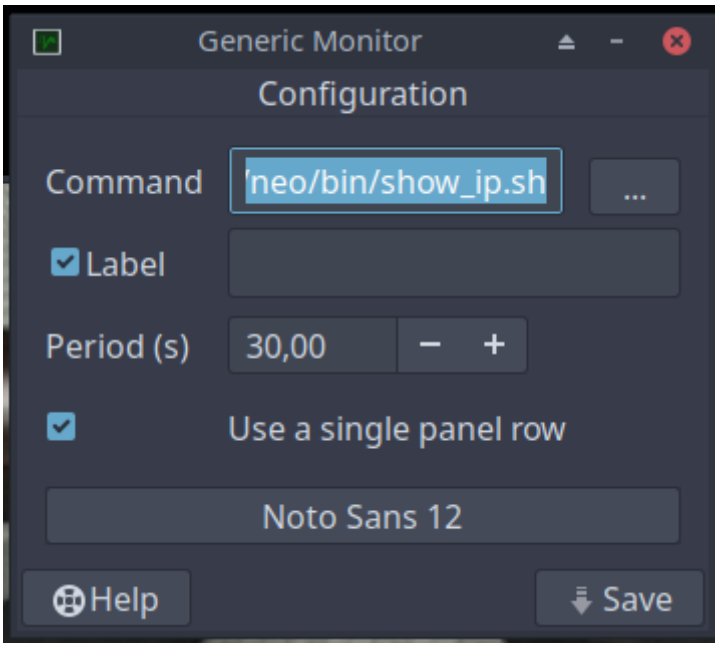
# Find the first IPv4 address
for IP in "${IP_ADDRS[@]}"; do
  if [[ "$IP" =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then
    echo "$IP"
    break
  fi
done
```

```
#!/bin/bash

ip -4 addr show scope global | grep -oP '(?<=inet\s)\d+(\.\d+){3}'
```

```
chmod +x show_ip.sh
```

Add a generic monitor:


Generic Monitor
Configuration
Command
'neo/bin/show_ip.sh
...
Label
Period (s)
30,00
Use a single panel row
Noto Sans 12
Help
Save

That's it.

Disable SELinux on Fedora

Permanently Disable SELinux on Fedora

Update the SELinux configuration file and set `SELINUX=disabled` to permanently disable the SELinux on your system. This will completely disable all the SELinux contexts.

```
sudo nano /etc/selinux/config
```

Set `SELINUX` value to `disabled`:

```
SELINUX=disabled
```

Disable SELinux in Fedora

Reboot your instance after making changes.

Note – You can again activate the SELinux by setting `SELINUX=enforcing` in configuration file.

How to Disable SELinux on Fedora

Create boot USB

Linux

```
sudo lsblk
sudo dd if=./Fedora-KDE-Live-x86_64-41-1.4.iso of=/dev/sdb bs=4M status=progress oflag=sync
```

MacOS

```
diskutil list
diskutil unmountDisk /dev/disk5
sudo dd if=./Fedora-KDE-Live-x86_64-41-1.4.iso of=/dev/rdisk5 bs=4M status=progress oflag=sync
diskutil unmountDisk /dev/disk5
```

Install network scanner on Archlinux

Install the drivers with pamac

run the commandline:

```
sudo brsaneconfig4 -a name=Brother model=MFC-9330CDW ip=192.168.1.108
```

check if it is working

```
scanimage -L
```

```
device `brother4:net1;dev0' is a Brother Brother MFC-9330CDW
device `v4l:/dev/video2' is a Noname Logitech BRIO virtual device
device `v4l:/dev/video0' is a Noname Logitech BRIO virtual devi
```

Install xrdp

Introduction

Fedora

```
sudo dnf install -y xrdp xorgxrdp
sudo systemctl enable xrdp
sudo systemctl start xrdp
sudo systemctl status xrdp
```

Create the group tsusers and add all users using Remote Desktop to it

Open Firewall on port 3389

> Check if the service is listening on tcp4

```
sudo nano /etc/xrdp/xrdp.ini
```

```
port=tcp://:3389



; Some session types such as Xorg and Xvnc start a display server.
; Startup command-line parameters for the display server are configured
; in sesman.ini. See and configure also sesman.ini.
[Xorg]
name=Xorg
lib=libxup.so
username=ask
password=ask
port=-1
code=20
```

Install xrdp on Fedora 42

How to Set Up XRDP on Fedora 42 XFCE

Follow these steps for a reliable and quick XRDP setup with XFCE on Fedora 42:

1. **Install XRDP and XFCE (if not already installed)**

```
sudo dnf install xrdp xorgxrdp
```

2. Enable and Start XRDP Service

```
sudo systemctl enable --now xrdp
```

3. Configure the Firewall

open Port 3389 for rdp

4. Set Up the XFCE Session for XRDP
Create a file named `.Xclients` in your home directory with the following content:

echo "xfce4-session" > ~/.Xclients

```
echo "xfce4-session" > ~/.Xclients
chmod +x ~/.Xclients
```

6. Restart XRDP Services

```
sudo systemctl restart xrdp
sudo systemctl restart xrdp-sesman
```

7. Connect via RDP

- Use Devolutions RDM or any RDP client.
- Enter your Fedora machine's IP address and credentials.

Summary Table

Step	Command/Action
Install XRDP	`sudo dnf install xrdp xorgxrdp`
Enable & start service	`sudo systemctl enable --now xrdp`
Firewall open port	`sudo firewall-cmd --permanent --add-port=3389/tcp; sudo firewall-cmd --reload`
Configure session	`echo "xfce4-session" > ~/.Xclients; chmod +x ~/.Xclients`
(Optional) SELinux fix	`sudo chcon --type=bin_t /usr/sbin/xrdp*`
Restart XRDP	`sudo systemctl restart xrdp xrdp-sesman`

This setup gives you a fast, graphical remote desktop on Fedora XFCE with minimal hassle.

Install send mail service on Fedora

Overview

This guide explains how to set up authenticated email sending from the command line on Fedora using **msmtp** (a lightweight SMTP client) and **s-nail** (a mailx-compatible mail utility). This method is ideal for scripts and system notifications in environments where only authenticated SMTP is allowed.

1. Install Required Packages

```
sudo dnf install -y msmtp s-nail
```

2. Configure msmtp

1. Copy the example configuration (optional):

   ```
   sudo cp /usr/share/doc/msmtp/msmtprc-system.example /etc/msmtprc
   ```

2. Edit `/etc/msmtprc` and adjust to your SMTP provider:

   ```
   sudo nano /etc/msmtprc
   ```

   Example configuration:

   ```
   defaults
   auth           on
   tls            on
   tls_trust_file /etc/ssl/certs/ca-bundle.crt
   logfile        /var/log/msmtp.log

   account        default
   host           mail.hosting.de
   ```

```
port            587
from            admin@simmy.org
user            admin@simmy.org
password        <super-secret>
```

3. Set permissions to protect your password:

```
sudo chmod 600 /etc/msmtprc
```

3. Configure s-nail to Use msmtp

Add the following line to `/etc/s-nail.rc` or your `~/.mailrc`:

```
set mta=/usr/bin/msmtp
```

4. Send a Test Email

```
echo "This is the body" | mail -s "Test Subject" recipient@example.com
```

- If the command returns no errors, the mail was sent successfully.
- Check `/var/log/msmtp.log` for troubleshooting if needed.

5. Notes

- If you receive an error like **"Authenticated user is not permitted to override sender address"**, ensure the `from` address in `/etc/msmtprc` matches the authenticated SMTP user, or configure your SMTP provider to allow the desired sender address.
- For use in scripts (e.g., backup notifications), simply use the `mail` command as shown above.

References

- Sending e-mails via mailbox.org with msmtp on Fedora
- Fedora Docs: Mail Servers
- Fedora Forum: Sending mail with the (mailx) command

Install sendmail service on Debian

Overview

This guide explains how to set up authenticated email sending from the command line on Debian-based systems (including Proxmox Backup Server) using **msmtp** (a lightweight SMTP client) and **s-nail** (a mailx-compatible utility). This is ideal for system notifications, backup/email scripts, and environments with **DMARC/SPF** filtering where authenticated sending is required.

1. Install Required Packages

```
apt update
apt install -y msmtp s-nail
```

2. Configure msmtp

1. **Create/Edit the global configuration file:**

   ```
   nano /etc/msmtprc
   ```

2. **Example `/etc/msmtprc`:**

   ```
   defaults
   auth           on
   tls            on
   tls_trust_file /etc/ssl/certs/ca-certificates.crt
   logfile        /var/log/msmtp.log
   syslog         LOG_MAIL

   account        default
   host           mail.hosting.de
   ```

```
port            587
from            admin@simmy.org
user            admin@simmy.org
password
```

- > **Important:** "from" and "user" should match your authenticated email address for DMARC/SPF.

3. **Example 2 /etc/msmtprc**
   1. ```
      syslog LOG_MAIL

      defaults
      auth on
      tls off
      tls_trust_file /etc/ssl/certs/ca-certificates.crt
      logfile /var/log/msmtp.log

      account ucs-backup
      host ucs-backup.simmy.ch
      port 25
      from pbs01@simmy.ch
      account default : ucs-backup
      ```

   - > Use only plain ASCII spaces (no tabs or Unicode spaces).

4. **Set strict permissions:**

   ```
   chmod 600 /etc/msmtprc
   ```

3. Configure `s-nail` or `mailx` to use msmtp

Add the following line to your `/etc/s-nail.rc` (system-wide) or `~/.mailrc` (per user):

```
set mta=/usr/bin/msmtp
```
```

4. Send a Test Email

Use the mail command to test sending:

```
echo "This is the body" | mail -s "Test Subject" recipient@example.com
```

On success, no output is shown. Check `/var/log/msmtp.log` or `/var/log/mail.log` (if syslog is enabled) for debug info if not delivered.

5. Troubleshooting

- If mail arrives in Junk/Spam, create a filter at your destination mailbox to whitelist the sender or move to Inbox.
- If you see an error like “**account default was already**

Install xrdp on Fedora Xfce

Overview

This document describes how to install and configure the XRDP server on Fedora 43 with the Xfce desktop environment so that Windows, macOS, and Guacamole clients can connect via RDP. Each Linux user who should be able to log in via XRDP needs their own `startwm.sh` to launch Xfce correctly.

Prerequisites

- Fedora 43 VM or physical host with the Xfce desktop environment installed (PRETTY_NAME="Fedora Linux 43 (Xfce)"). [web:14]
- Root or sudo access on the Fedora system.
- Network connectivity from RDP clients (Windows, macOS, Guacamole) to TCP port 3389 on the Fedora host.

Install and Enable XRDP

Install XRDP and its Xorg backend, then enable and start the service. Fedora 40/41 XRDP documentation uses the same pattern and works on Fedora 43. [web:21][web:17]

```
sudo dnf install -y xrdp xorgxrdp
sudo systemctl enable --now xrdp
sudo systemctl status xrdp
```

Open the Firewall for RDP

If `firewalld` is running, open TCP port 3389 permanently and reload the firewall rules. [web:21][web:17]

```
sudo firewall-cmd --permanent --add-port=3389/tcp
sudo firewall-cmd --reload
```

Create `startwm.sh` for Each User

On Fedora, XRDP uses a per-user startup script named `startwm.sh` in the user's home directory to start the desktop session. Fedora's XRDP guide shows this pattern for multiple desktops; for Xfce the command is `dbus-launch --exit-with-session /usr/bin/startxfce4`.

Repeat the following steps for **each user account** that should be able to log in via XRDP:

```
# as the target user (not root)
cat > ~/startwm.sh << 'EOF'
#!/bin/sh
export LANG=en_US.UTF-8
export LC_ALL=en_US.UTF-8
exec dbus-launch --exit-with-session /usr/bin/startxfce4
EOF

chmod 755 ~/startwm.sh
```

Explanation:

- `dbus-launch --exit-with-session` ensures a proper D-Bus session is created for Xfce, which is required for a fully functional desktop over XRDP.
- `/usr/bin/startxfce4` starts the Xfce session.
- `chmod 755` makes the script executable so XRDP can run it at login.

Optional: Global `/etc/xrdp/startwm.sh`

If you want a single configuration for all users, you can copy the same script to `/etc/xrdp/startwm.sh` so XRDP uses it globally. This approach is also referenced in XRDP discussions about custom session commands.

```
sudo cp /home/<username>/startwm.sh /etc/xrdp/startwm.sh
sudo chmod 755 /etc/xrdp/startwm.sh
sudo systemctl restart xrdp
```

Replace `<username>` with a real user name when copying from an existing script.

SELinux Considerations (Optional)

On some Fedora installations, SELinux can interfere with XRDP. Recent XRDP-on-Fedora guides use `chcon` to assign the `bin_t` type to XRDP binaries if SELinux denials occur. [web:17]

```
sudo chcon --type=bin_t /usr/sbin/xrdp
sudo chcon --type=bin_t /usr/sbin/xrdp-sesman
sudo systemctl restart xrdp
```

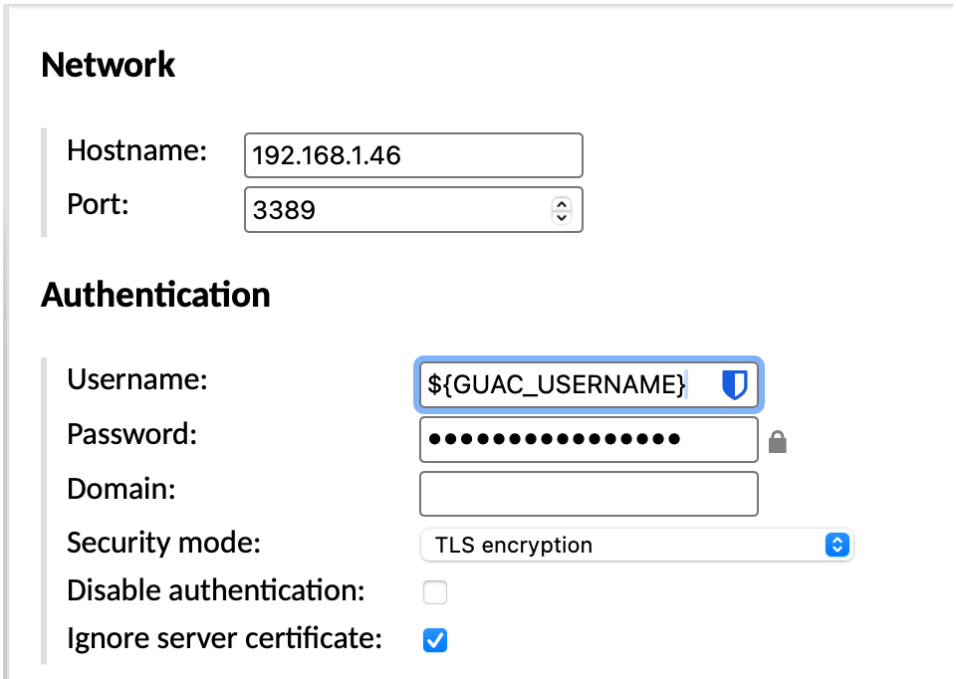
Testing with a Native RDP Client

Test XRDP with a standard RDP client before integrating with Guacamole. Fedora XRDP documentation uses Windows Remote Desktop as the reference client. [web:21]

1. From a Windows machine, open **Remote Desktop Connection** (`mstsc.exe`).
2. Enter the Fedora host name or IP (for example `fedora-xfce.example.local`) and connect. [web:21]
3. Log in using a Fedora user that has a `~/startwm.sh` configured.
4. Verify that an Xfce desktop session appears and is usable.

Using XRDP from Guacamole

Once XRDP and Xfce are working locally, Guacamole can connect using the RDP protocol. The key is to match the security mode and certificate options so that negotiation succeeds.



Setting	Value
Protocol	RDP
Hostname	IP Address
Port	3389
Username	${GUAC_USERNAME}
Password	${GUAC_PASSWORD}
Security Mode	TLS
Ignore server certificate	enable

Summary

- Install XRDP and Xorg backend with `dnf install -y xrdp xorgxrdp`, then enable the service.
- Open the firewall for TCP port 3389 if `firewalld` is running.
- Create a per-user `~/startwm.sh` containing `dbus-launch --exit-with-session /usr/bin/startxfce4` and make it executable.
- Optionally, place the same script at `/etc/xrdp/startwm.sh` for a global configuration.
- Verify RDP access with a native client (e.g., Windows mstsc), then configure an RDP connection in Guacamole pointing at the Fedora 43 XRDP server.

10?GbE Network Tuning on Fedora

For Fedora workstations/servers mounting:
• Prefer kernel CIFS mounts (mount -t cifs) over GVFS/GUI mounts; GVFS is noticeably slower for bulk IO.
• Add a basic 10 GbE-friendly /etc/sysctl.d/10g.conf on Fedora, e.g.:
• net.core.rmem_max = 134217728
• net.core.wmem_max = 134217728
• net.ipv4.tcp_rmem = 4096 87380 134217728
• net.ipv4.tcp_wmem = 4096 65536 134217728 This allows the SMB/TCP stack to actually fill the pipe.
Typical high-throughput cifs mount line for large sequential IO:

```
mount -t cifs //n2/share /mnt/n2 \
  -o vers=3.1.1,sec=ntlmssp,cache=strict,echo_interval=60,actimeo=1,soft \
  ,rsize=1048576,wsize=1048576,uid=1000,gid=1000
```

Those options mirror what Red Hat and others recommend for high-bandwidth SMB on modern kernels.

10?GbE Network Tuning on Debian

Introduction

This document describes a small, focused kernel network tuning for Debian systems connected via 10 GbE, using a custom `/etc/sysctl.d/10g.conf` file and the standard `sysctl` mechanism to apply the settings. These values follow common recommendations for high-bandwidth Linux hosts by increasing TCP buffer limits and using modern congestion control and queuing disciplines.

Purpose of 10g.conf

The goal of `10g.conf` is to allow the TCP stack to efficiently fill a 10 Gb/s link for protocols like SMB and NFS while remaining conservative enough for general-purpose servers. It does this by:

- Increasing the maximum socket buffer sizes for receive and transmit.
- Raising the autotuning ceiling for TCP read and write buffers.
- Improving backlog and connection queue limits for busy hosts.
- Enabling modern congestion control (`htcp`) and fair queuing (`fq`).

10g.conf Content

Create `/etc/sysctl.d/10g.conf` with the following content:

```
# /etc/sysctl.d/10g.conf
# Basic 10 GbE TCP tuning for Debian / Linux.
# Focus: higher throughput for SMB/NFS and other bulk transfers over 10G.

# Allow larger TCP socket buffers for high-bandwidth links
net.core.rmem_max = 33554432
net.core.wmem_max = 33554432
net.core.rmem_default = 262144
net.core.wmem_default = 262144
```

```
# TCP autotuning limits: min, default, max
net.ipv4.tcp_rmem = 4096 87380 33554432
net.ipv4.tcp_wmem = 4096 65536 33554432

# Optional but often helpful for busy 10GbE hosts
net.core.netdev_max_backlog = 250000
net.core.somaxconn = 4096

# Use modern congestion control and fair queueing
net.ipv4.tcp_congestion_control = htcp
net.core.default_qdisc = fq
```

Parameter Notes

- `net.core.rmem_max` / `wmem_max`: Maximum per-socket buffer size; 32 MB is typical for 10 GbE hosts and is widely recommended.
- `tcp_rmem` / `tcp_wmem`: Trio of values (min, default, max) used by TCP autotuning; the higher max allows large windows on clean high-latency or high-bandwidth paths.
- `netdev_max_backlog`: Maximum number of packets that can be queued when the kernel receives them faster than it can process; 250 000 is a common safe value on modern hardware.
- `tcp_congestion_control = htcp`: Uses HighSpeed TCP Congestion Control, designed for fast long-fat-pipe links.
- `default_qdisc = fq`: Fair Queuing with pacing, often recommended for servers with modern kernels.

Applying the Configuration

Immediate Application

After saving `/etc/sysctl.d/10g.conf`, apply the settings to the running kernel:

```
sudo sysctl --system
```

This command reloads all configuration files under `/etc/sysctl.d`, `/run/sysctl.d` and `/usr/lib/sysctl.d`, and applies all settings without requiring a reboot.

Note: The settings take effect immediately for new connections. Existing long-lived TCP sessions may continue using their previous buffer sizes until they are re-established.

Verification

You can verify that the values have been applied by querying a few key parameters:

```
sysctl net.core.rmem_max
sysctl net.core.wmem_max
sysctl net.ipv4.tcp_rmem
sysctl net.ipv4.tcp_wmem
sysctl net.ipv4.tcp_congestion_control
sysctl net.core.default_qdisc
```

The output should reflect the values specified in `10g.conf`, confirming that the tuning is active.

Usage and Testing

Once the configuration is applied, you can re-test SMB/NFS throughput (for example, from your Nextcloud host or other Debian clients) using representative workloads or tools like `iperf3`, large file copies, or application-level benchmarks. The tuning primarily benefits sustained transfers where the network path was previously constrained by default TCP buffer limits rather than disk or CPU.

Autoupdate on Debian

```
sudo apt install unattended-upgrades
sudo dpkg-reconfigure unattended-upgrades
```

Add a user to the sudoers group on Debian 13

```
usermod -aG sudo master
```

How to

Verify NFS export

Check NFS exports on `nas05.simmy.ch` from the Linux:

```
sudo showmount -e nas05.simmy.ch
```

.bashrc interactive shell

```
# at the very top of ~/.bashrc, before any echo/printf/etc.
case "$-" in
  *i*) ;;          # interactive shell, continue
  *) return ;;     # non-interactive (scp/sftp, ssh command), stop here
esac
```

```
# at the very end of ~/.bashrc
alias ll='ls -lsah'

fastfetch
```