



EK

Totally not a hacker Email Twitter Github

Linux-Unix-IT Tips and Tricks #1

Different Linux / Unix / IT tips, notes, howto

Other parts

[Part 1](#) [Part 2](#) [Part 3](#)

Phpinfo

```
echo "<?php phpinfo(); ?>" > file  
root@linux: ~ # php -i
```

Apache VirtualHost

```
<Directory ~ ".*/.svn">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</Directory>  
  
<VirtualHost 127.0.0.1:80>  
    ServerName vhost1  
    ErrorLog /logs/vhost1/error_log  
    DocumentRoot /www/vhost1/htdocs
```

CONTENTS

[Other parts](#)[Phpinfo](#)[Apache VirtualHost](#)[Grant all access to MySQL DB](#)[Check disk with DD](#)[Manual install php extensions](#)[Create MySQL DB](#)[Update user grants for MySQL DB](#)[Skip MySQL DB Replication error](#)[Simple MySQL DB backup/restore commands](#)[Run crontab task with lockf or lockfile](#)[Run crontab everyday except Saturday](#)[Create rsync share](#)

```
<Directory /www/vhost1/htdocs>
    AllowOverride All

</Directory>
</VirtualHost>
```

Grant all access to MySQL DB

```
grant all privileges on user.* to user@"%" identified by "P@ssw0rd";
```

Check disk with DD

```
dd if=/dev/da0 of=/dev/null bs8
for progress Ctrl+T to send USR1 or kill -USR1 DDPID
```

Manual install php extensions

```
${PHP_PREFIX}/bin/pecl uninstall memcache
${PHP_PREFIX}/bin/pecl install memcache

add to php.ini
extension=memcache

Build extension from source
wget memcache-2.2.5.tgz
/usr/local/php/bin/phpize
./configure --with-php-config=/usr/local/php/bin/php-config
```

Create MySQL DB

Count /tmp used disk space

Clean Yum pkg cache

MySQL Master-Slave replication

Change MySQL root password

Megarc raid cmd

Megacli raid cmd

SSH tunnel

Ram disk FreeBSD

Add firewall rules Linux or FreeBSD

Add 6G swap to FreeBSD

Clone disk with dd

Access to svn server with ssh-key

Create iso image with dd

Create swap in Linux

Split big file in small parts

Show PostgreSQL status

Soft Raid FreeBSD

```
CREATE DATABASE TEST default charset utf8;
grant all on testuser.* to 'testuser'@'SOMEIP' identified by 'P@ssw0rd'
```

Update user grants for MySQL DB

```
update user set password=PASSWORD('P@ssw0rd') where user='testuser';
show grants for 'testuser'@'SOMEIP';
revoke all on TEST.* from 'testuser'@'SOMEIP';
```

Skip MySQL DB Replication error

```
show slave status;
SET GLOBAL SQL_SLAVE_SKIP_COUNTER = N;
```

Simple MySQL DB backup\restore commands

```
Backup all db server
mysqldump --master-data=2 --opt --single-transaction --all-databases |

Backup only tables
mysqldump test --tables tag_table > /backup/test.tag_table.sql
mysqldump -u USER -pPASSWORD DATABASE TABLE1 TABLE2 TABLE3 > /path/to/f

Backup one database
mysqldump -u USER -pPASSWORD DATABASE > /path/to/file/dump.sql
mysqldump -u USER -pPASSWORD DATABASE | gzip > /path/to/outputfile.sql.
mysqldump -u USER -pPASSWORD DATABASE | gzip > `date +%path/to/outputfi
```

Test channel speed

Patch apply

Revoke all grant and remove MySQL user

Cron task start at reboot

Simple rsyncd.conf

Open files limit

Use curl to check web site answer time

Run vncserver at port 5999

Create 200 host records in /etc/hosts

Com terminal for cisco

Disable php for custom dir in Apache

Print all ps pids with owner root

Set expired password for FreeBSD user

Enable coredumps in FreeBSD or Linux

Enable coredumps in NginX

```
Backup database scheme
mysqldump --no-data -u USER -pPASSWORD DATABASE > /path/to/file/schema

Restore database
mysql -u USER -pPASSWORD DATABASE < /path/to/dump.sql
gunzip < /path/to/outputfile.sql.gz | mysql -u USER -pPASSWORD DATABASE
zcat /path/to/outputfile.sql.gz | mysql -u USER -pPASSWORD DATABASE
```

Add alias to network interface in FreeBSD

Print all locked request in MySQL

Redirect port from 5190 to 5222 with iptables

Limit POST request size to 20MB

Apache error log limit 2G

Change nginx version number

Run crontab task with lockf or lockfile

```
*/5 * * * * userlogin /usr/bin/lockf -st 0 /tmp/5min.lock sh cron1.sh
*/5 * * * * userlogin lockfile -s0 -r0 /tmp/5min.lock && (sh cron1.sh ;
```

Run crontab everyday except Saturday

```
0 * * * 0,1,2,3,4,6 userlogin sh cron1.sh
```

Create rsync share

```
/etc/rsyncd.conf
[www]
path = /www
list = yes
read only = yes
hosts allow = 10.0.0.0/8
dont compress = *.*

rsync -avP server::www/ /www1/
```

Count /tmp used disk space

```
sudo du -hs /tmp/*
```

Clean Yum pkg cache

```
Pkg cache live here - /var/cache/yum/updates  
yum clean all
```

MySQL Master-Slave replication

```
# at master /etc/my.cnf  
server-id=1  
binlog-format = mixed  
log-bin=mysql-bin  
datadir=/var/lib/mysql  
innodb_flush_log_at_trx_commit=1  
sync_binlog=1  
  
CREATE USER replicant@<<slave-server-ip>>;  
GRANT REPLICATION SLAVE ON *.* TO replicant@<<slave-server-ip>> IDENTIFIED BY '<password>';  
  
$ mysqldump --skip-lock-tables --single-transaction --flush-logs --hex-  
$ head dump.sql -n80 | grep "MASTER_LOG_POS"  
$ scp ~/dump.sql.gz mysql-user@<<slave-server-ip>>:~/  
  
# at slave /etc/my.cnf  
server-id = 101  
binlog-format = mixed
```

```

log_bin          = mysql-bin
relay-log        = mysql-relay-bin
log-slave-updates = 1
read-only        = 1

$ gunzip ~/dump.sql.gz
$ mysql -u root -p < ~/dump.sql

CHANGE MASTER TO MASTER_HOST='<master-server-ip>',MASTER_USER='replic
START SLAVE;
SHOW SLAVE STATUS \G;

# fix some replication errors
STOP SLAVE;SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 1;START SLAVE;

```

Change MySQL root password

```

use mysql;update user set password=password('P@ssw0rd') where user='root';
flush privileges;

```

Megarc raid cmd

```

megarc -dispCfg -a0
run rebuild
megarc -dorblld -a0 -rbldarray[0:1]
http://www.freebsdwiki.net/index.php/Megarc

```

Megacli raid cmd

```
http://hwraid.le-vert.net/wiki/LSIMegaRAIDSG
http://tools.rapidsoft.de/perc/perc-cheat-sheet.html
server:~# megacli -AdpGetProp RebuildRate -a0
Adapter 0: Rebuild Rate = 30%
server:~# megacli -AdpSetProp RebuildRate 60 -a0
Adapter 0: Set rebuild rate to 60% success.
LINUX
server:~# cat /proc/sys/dev/raid/speed_limit_min
1000
server:~# cat /proc/sys/dev/raid/speed_limit_max
200000

Speed up limit to 50 mb\s
server:~# echo 50000 > /proc/sys/dev/raid/speed_limit_min
```

SSH tunnel

```
Local Port Forwarding
ssh -L 3306:localhost:3306 username@hostname
telnet localhost 3306

Remote Port Forwarding
ssh -R 9000:localhost:8001 username@hostname
ssh -R 2222:localhost:22 username@hostname - SSH
ssh -R 2223:localhost:5902 username@hostname - VNC
autossh -M 20000 -f -R 2222:localhost:80 username@hostname
telnet someserver 9000 -> go to localhost:8001
```

Ram disk FreeBSD

```
/etc/rc.conf
mdconfig_md0="-t malloc -s 512m -u 0"
```

```
mdconfig_md0_newfs="-U"  
mdconfig_md0_owner="www"  
mdconfig_md0_perms="775"  
mdconfig_md0_cmd="chown www:www /www/ramdisk"  
mkdir -p /www/ramdisk  
/etc/rc.d/mdconfig start  
df -h
```

Add firewall rules Linux or FreeBSD

```
Linux  
iptables -A INPUT -s 127.0.0.1 -p tcp --dport 8080 -j ACCEPT  
service iptables save  
iptables -nL | grep 8080  
  
FreeBSD  
ipfw add pass tcp from 127.0.0.1 to me 8080 setup  
sh /usr/share/examples/ipfw/change_rules.sh  
ipfw list | grep 8080
```

Add 6G swap to FreeBSD

```
dd if=/dev/zero of=/www/swap0 bs=1M count=6144  
chmod 0600 /www/swap0  
/etc/rc.conf  
swapfile="/www/swap0"  
mdconfig -a -t vnode -f /www/swap0 -u 0 && swapon /dev/md0
```

Clone disk with dd


```
dd if=/dev/old_disk of=/dev/new_disk conv=noerror,sync
```

Access to svn server with ssh-key

```
useradd svnuser  
ssh-keygen -t rsa  
mv ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys  
add to ~/.ssh/authorized_keys  
no-pty,no-port-forwarding,no-X11-forwarding,no-agent-forwarding,command
```

Create iso image with dd

```
sudo dd if=/dev/scd0 of=/tmp/isoimage.iso  
sudo mount -t udf,iso9660 -o loop /tmp/isoimage.iso /mnt/cdrom/
```

Create swap in Linux

```
fdisk /dev/sda  
mkswap -L 'swap' /dev/sda6  
swapon -a
```

Split big file in small parts

```
split -b1m large.mp3 large.mp3.  
collect back  
cat large.mp3.* > large.mp3
```

Show PostgreSQL status

```
select * from pg_stat_activity;
```

Soft Raid FreeBSD

```
echo 'geom_mirror_load="YES"' >> /boot/loader.conf
sysctl kern.geom.debugflags=16
gmirror label -v -b round-robin -s 131072 gm0 /dev/mfid0
change /etc/fstab /dev/mfid0s2a to /dev/mirror/gm0s1a
reboot
gmirror list
gmirror status
gmirror insert gm0 /dev/mfid1
gmirror configure -b round-robin -s 131072 gm*
gmirror remove gm0 /dev/mfid1
gmirror foreget -v gm0
gmirror rebuild -v gm1 /dev/mfid3
```

Test channel speed

```
at server:
iperf -s
at client:
iperf -V -c SERVERIP -t 6000
```

Patch apply

```
patch --dry-run -p1 < patchfile.patch
if all ok
patch -p1 < patchfile.patch
```

Revoke all grant and remove MySQL user

```
REVOKE ALL PRIVILEGES on `DBNAME`.* from 'USERNAME'@'IP' ;
drop user 'USERNAME'@'IP';
select * from mysql.user\G;
select * from mysql.user where user='USERNAME'\G;
```

Cron task start at reboot

```
/etc/crontab
@reboot      username /usr/bin/bash /etc/startAtReboot.sh
```

Simple rsyncd.conf

```
/etc/rsyncd.conf
pid file = /var/run/rsyncd.pid
use chroot = yes
uid = nobody
gid = nobody
hosts allow = 192.168.56.0/24
hosts deny = *
pid file = /var/run/rsyncd.pid
[test]
path=/test
comment=test
```

```
read only = yes
hosts allow = 10.0.0.1
hosts deny = *

To start
rsync --daemon
```

Open files limit

```
http://ss64.com/bash/limits.conf.html
/etc/security/limits.conf
*          hard    nofile      65535
*          soft    nofile      65535
username   hard    nofile      65536
```

Use curl to check web site answer time

```
curl -w '\nLookup time:\t%{time_namelookup}\nConnect time:\t%{time_conn
```

Run vncserver at port 5999

```
vncserver -geometry 1024x768 -depth 24 :99
```

Create 200 host records in /etc/hosts

```
192.168.99.1-200 n001-n200
P=1; for i in $(seq -w 200); do echo "192.168.99.$P n$i"; P=$(expr $P +
```

Com terminal for cisco

```
apt-get install cutecom(minicom)
```

Disable php for custom dir in Apache

```
<Directory /var/www/htdocs/images>
    RemoveHandler .php .phtml .php3
    AddType application/x-httpd-php-source .php .phtml .php3
    Allow from all
</Directory>
```

Print all ps pids with owner root

```
ps -aux | awk '{if ($1 == "root") print $2}'
```

Set expired password for FreeBSD user

```
echo P@ssw0rd|sudo /usr/sbin/pw usermod username -h0 -p 1
```

Enable coredumps in FreeBSD or Linux

```
FreeBSD
sysctl kern.coredump=1
sysctl kern.sugid_coredump=1
sysctl kern.corefile=/core/%N-%P.core
%N - name of process
%P - PID
%U - username login

Linux
mkdir /core
chmod 777/core
echo "/core" > /proc/sys/kernel/core_pattern
sysctl kernel.core_pattern=/core/%P.core
```

Enable coredumps in NginX

```
Need to compile nginx with --with-debug
Add to nginx.conf
worker_rlimit_core 50m;
working_directory /core/;
echo 'www soft core unlimited' >> /etc/security/limits.conf
/etc/init.d/httpd restart
```

Add alias to network interface in FreeBSD

```
ifconfig em0 alias 192.168.56.100/32
```

Print all locked request in MySQL

```
mysql -e "show full processlist" | grep "Locked"
```

Redirect port from 5190 to 5222 with iptables

```
iptables -A INPUT -p tcp --dport 5190 -j ACCEPT  
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5190 -j REDIRECT -
```

Limit POST request size to 20MB

```
add to php.ini  
php_value max_execution_time 600  
php_value upload_max_filesize 20M  
php_value post_max_size 20M
```

Apache error log limit 2G

```
[notice] child pid 15443 exit signal File size limit exceeded (25)  
some where you have log file size more than 2G, find it  
find /logs/* -size +2000000k
```

Change nginx version number

```
src/core/nginx.h  
nginx version: nginx/0.7.62  
recompile from source
```

Linux-Unix-IT Tips and Tricks #1 was published on June 22, 2014 and last modified on June 22, 2014.

0 Comments

EK BLOG

1 Login ▾

♥ Recommend

🐦 Tweet

f Share

Sort by Best ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?



Name

Be the first to comment.

✉ Subscribe

Ⓓ Add Disqus

🔒 Disqus' Privacy Policy

DISQUS

YOU MIGHT ALSO ENJOY

(VIEW ALL POSTS)

- Move from HDD to SSD with ArchLinux
- Linux SysAdm/DevOps Interview Questions
- ArchLinux Installation Guide

