Linux ??

Linux □□

- Linux □□□□□□□□ SFTP
- Linux Expect □□□□□
- Linux NAT MASQUERADE □□
- Linux Rsync □□□□□□
- Linux □□□□□□
- Linux □□ rc.local □□□□□□
- Linux OpenVPN Server □□
- Linux TACACS+ Server □□
- Linux Cpufrequtils CPU□□□□
- Linux MegaRaid CLI □□□□□
- Linux R620 □□ IPMI□□□□□□
- Linux tcpdump□□□□
- Linux □□ Trash □□□
- Linux Looking Glass IP□□□□
- Linux SSH KEY □□□□□
- Linux X11 Forwarding □□□□
- Linux TAB□□□□□
- Linux SSH □□ OTP □□□□
- Linux □□ nftables □ Fail2ban □□□□
- Linux □□ Chroot
- Linux □□ IPV6
- Linux Syslog-ng □□□□□□□
- Linux □□ Python3 FTP
- Linux Crontab □□□□
- Linux □□□□□□□□□

Linux ????????SFTP

????

- **Debian 12**

?????

```
useradd -M testuser -s /usr/sbin/nologin
```

-M □□□□□□

testuser □□□

-s /usr/sbin/nologin □□□□□ bash

```
passwd testuser
```

?????

□□□□□ /ftp/test□□□□□□□ test□□□

```
mkdir /ftp

cd /ftp

mkdir test
```

test□□□□□□□□□□□□□□□□□ root

```
chown testuser:testuser test
```

??SSHD

□□ `/etc/ssh/sshd_config`

```
#Subsystem      sftp    /usr/lib/openssh/sftp-server

Subsystem sftp internal-sftp
Match User testuser
        ChrootDirectory /ftp
```

```
        ForceCommand internal-sftp
        AllowTcpForwarding no
        X11Forwarding no
```

```
service ssh restart
```

??

SFTP□□□□□

```
root@ssh:~# sftp testuser@10.0.0.200
testuser@10.0.0.200's password:
Connected to 10.0.0.200.
sftp> ls
test
sftp>
```

SSH□□□□□

```
root@ssh:~# ssh testuser@10.0.0.200
testuser@10.0.0.200's password:
This service allows sftp connections only.
Connection to 10.0.0.200 closed.
```

Linux Expect ?????

??Expect

```
apt install expect
```

????

??telnet??

□□ `telnet_expect.sh`

```
#!/usr/bin/expect
set username "test"
set password "test123"
set command "use command"
spawn telnet $ip
    expect {
        "*ogin:" {
            send "$username\r"
            expect "*assword:"
            send "$password\r"
            expect ">"
            send "$command\r"
            expect ">"
            send "exit\r"
        }
    }
```

```
chmod +x telnet_expect.sh
```

??????????

□□□□□□□□□ `ip.txt`

```
device_a*192.168.0.1
device_b*192.168.0.2
device_c*192.168.0.3
```

□□□□□□□ `telnet.sh`□□□ **log**□□□□□ **log**

```
#!/bin/bash
rm log/*

for i in `cat ip.txt`
do {
substr=$(echo $i | cut -d"*" -f 2)
kip=$substr
substr=$(echo $i | cut -d"*" -f 1)
kname=$substr
/usr/bin/expect expect.sh $kip $kname
}&
done
wait
```

□□ `expect.sh`

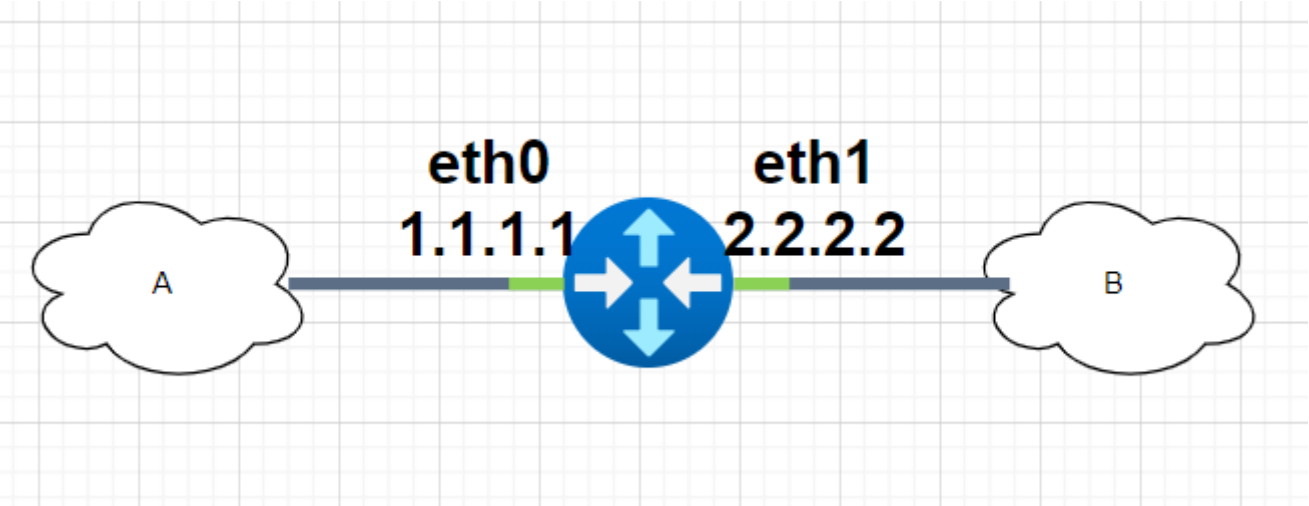
```
#!/usr/bin/expect
set timeout 30
set username "test"
set password "test123"
set command "show configuration |display set|no-more"
set date [ clock format [ clock seconds ] -format "%Y%m%d" ]
set kip [lindex $argv 0]
set kname [lindex $argv 1]

spawn telnet $kip
log_file "log/$date-${kname}.log"
expect {
        "*ogin:" {
                send "$username\r"
                expect "*assword:"
                send "$password\r"
                expect ">"
                send "set cli timestamp\r"
                expect ">"
                send "$command\r"
                expect ">"
                send "exit\r"
```

```
                log_file
        }
}
```

```
chmod +x expect.sh
```

telnet.sh

Linux NAT MASQUERADE ??



A□□□　　192.168.12.0/24

B□□□　　10.0.137.0/24

```
B□□□□□□□□□ eth0(1.1.1.1) □□ A□
iptables -t nat -A POSTROUTING -o eth0 -s 10.0.137.0/24 -j MASQUERADE
```

```
A□□□□□□□□□ eth1(2.2.2.2) □□ B□
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.12.0/24 -j MASQUERADE
```

Linux Rsync ??????

??Rsync

Client□ Server□□ Rsync

```
apt-get install rsync
```

Server Rsync ??

□□ `/etc/rsyncd.conf`

```
[backup]
# target directory to copy
path = /backup
# hosts you allow to access
hosts allow = 10.0.0.5
hosts deny = *
list = true
uid = test1
gid = test1
read only = false
auth users = test1
secrets file = /etc/rsyncd.secrets
```

```
auth users □□□□

secrets file = □□□□
```

```
root@rsync:~# cat /etc/rsyncd.secrets
test1:test1pwd

□□□:□□
□□□□chmod 600 /etc/rsyncd.secrets
```

□□ **/backup□□□□□□ test1**

```
# □□□
groupadd test1
```

```
# 建使用者並指定群組
useradd -g test1 -s /usr/sbin/nologin -M test1

# 建立目錄並賦權
mkdir -p /backup
chown test1:test1 /backup
chmod 750 /backup
```

Client ??

```
列出模組
root@synology:~# rsync -rtn rsync://test1@10.0.0.110
backup
```

```
列出模組內檔案
root@synology:~# rsync -rtn rsync://test1@10.0.0.110/backup
Password:
```

Client新增密碼檔自動輸入密碼

```
新增/rsyncd.secrets ，輸入要讀取密碼
root@synology:~# cat /rsyncd.secrets
test1pwd
新增權限chmod 600 /rsyncd.secrets
```

備份指令

```
rsync -avz --chmod=Du+rwx --delete /volume1/homedata/jason --password-file=/rsyncd.secrets
test1@10.0.0.110::backup
```

- `-a`： **archive** 模式，保留權限、時間、符號連結等
- `-v`：顯示過程
- `-z`：壓縮傳輸（適合跨網路，但區域網路有時反而變慢）
- `--chmod=Du+rwx`：強制目錄允許 **u+rwx**
- `--delete`：刪除目標端多餘檔案

第一次使用為完整備份，之後依照參數為鏡像或增量備份 **(-delete**刪除不存在，即為兩端鏡像 **)**

-v，-verbose，详细输出

-q，-quiet，精简输出

-no-motd，不显示守护进程的MOTD（仅适用于manpage）

-c，-checksum，使用checksum而不是mod-time和size

-a，-archive，归档模式，-rlptgoD（无-H，-A，-X）

-no-OPTION，关闭选项，例如-no-D，

-r，-recursive，递归

-R，-relative，使用相对路径

-no-implied-dirs，不发送隐含的目录，与-relative连用时有用

-b，-backup，备份，-suffix或-backup-dir，

-backup-dir = DIR，将DIR作为备份目录

-suffix = SUFFIX，覆盖备份后缀~w / o -backup-dir，

-u，-update，仅仅进行更新

-inplace，输入文件就是输出文件，SEE MAN PAGE，

-append，追加数据

-append-verify，-append，但是会对已有数据进行验证

-d，-dirs，不递归传输目录

-l，-links，保留软连接

-L，-copy-links，像对待常规文件一样处理软连接

-copy-unsafe-links，仅仅拷贝不安全的连接

-safe-links□□□□□□□□□□□□□□□□□

-k□-copy-dirlinks□□□□□□□□□□□□□

-K□-keep-dirlinks□□□□□□□□□□□□□□□

-H□-hard-links□□□□□

-p□-perms□□□□

-E□-executability□□□□□□□□□

-chmod = CHMOD□□□□□□□□□□□□

-A□-acls□□ACL□□□-perms□

-X□-xattrs□□□□□□

-o□-owner□□□□□□□□□□□□□□

-g□-group□□□

-devices□□□□□□□□□□□□□□□□

-specials□□□□□□

-D□-devices -specials□□

-t□-times□□□□□□

-O□-omit-dir-times□-times□□□□□□□□□□

-super□□□□□□□□□□□□□□□□□□□□□

-fake-super□□xattrs□□□□□□□□□

-S□-sparse□□□□□□□□

-n□-dry-run□□□□□□□□

-W，-whole-file　　delta-xfer

-x，-one-file-system

-B，-block-size = SIZE　　checksum

-e，-rsh = COMMAND　　shell

-rsync-path = PROGRAM　　rsync

　　
-ignore-existing
-remove-source-files　　dirs
-del -delete-during
-delete
-delete-before
-delete-during
-delete-delay
-delete-after
-delete-excluded
-ignore-errors　　I / O
-force
-max-delete = NUM
-max-size = SIZE
-min-size = SIZE
-partial
-partial-dir = DIR
-delay-updates
-m，-prune-empty-dirs
-numeric-ids　　uid / gid
-timeout = SECONDS　　I / O　　s
-contimeout = SECONDS
-I，-ignore-times　　mtime，size
-size-only
-modify-window = NUM
-T，-temp-dir = DIR
-y，-fuzzy
-compare-dest = DIR
-copy-dest = DIR，-compare-dest
-link-dest = DIR，-compare-dest

-z，-compress□□□□□□□
-compress-level = NUM□□□□□□□□□
-skip-compress = LIST□□□□□□□□□□□□
-C，-cvs-exclude，CSV□□□□□□□□□□
-f，-filter = RULE□□□□□□□□□□□□
-F，-filter =’dir-merge /.rsync-filter’□□□□□-filter =’ – .rsync-filter’
-exclude = PATTERN□□□□PATTERN
-exclude-from = FILE□□□□□□□□□□□
-include = PATTERN□□□□□□□□□□□□
-include-from = FILE□□□□□□□□□□□□
-files-from = FILE□□□□□□□□□□□
-0，-from0 all
-from / filter□□□0□□
-s，-protect-args□□□□□□□□□□; □□□□□□□□□
-address = ADDRESS□□□□□□□
-port = PORT□□□□□
-sockopts = OPTIONS□□TCP□□
-blocking-io□□□shell□□□□□I / O.
-stats□□□□□□□□
-8，-8-bit-output□□□□□□□□□□□
-h，-human-readable□□□□□□□□□□□□
-progress□□□□□□
-P，-partial-progress□□
-i，-itemize-changes□□□□□□□□□
-out-format = FORMAT□□□□□□□□□□□□
-log-file = FILE□□□□
-log-file-format = FMT□□□□□□
-password-file = FILE□□□□
-list-only□□□□□
-bwlimit = KBPS□□□□; □□KBytes
-write-batch = FILE□□□□□□□□□
-only-write-batch = FILE，-write-batch□□□□□□□□□□
-read-batch = FILE□□□□□□□□□□□□
-protocol = NUM□□□□□□□□□
-iconv = CONVERT_SPEC□□□□□□□□
-4，-ipv4□□□IPv4
-6，-ipv6□□□IPv6
-version□□□□□
，-h，-help□□□□□□□□□-h□□□□□□□□-help□□□

Linux ??????

指令新增路由後，重啟會消失

```
ip route add 192.168.0.0/24 via 192.168.1.1 dev eth0
```

編輯 `/etc/network/interfaces`，在這個網路介面 啟動（ **up**）時，會執行一條靜態路由的指令

```
up ip route add 10.8.0.6/32 via 10.0.0.5
```

Linux ??rc.local ??????

????????rc.local

```
systemctl status rc-local.service
```

??rc.local??

□□ `/etc/systemd/system/rc-local.service`

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local
[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99
[Install]
WantedBy=multi-user.target
```

□□□□ **rc.local**□ `/etc/rc.local`

```
#!/bin/sh -e
□□□□□
exit 0
```

```
chmod +x /etc/rc.local
```

????

```
systemctl enable rc-local
systemctl start rc-local
systemctl status rc-local
```

Linux OpenVPN Server ??

??????

```
wget https://git.io/vpn -O openvpn-install.sh
```

openvpn-install.sh

```
bash ./openvpn-install.sh
```

□□□□□ **vpn server**

□□□ **client□□□□□ (/root/pve.ovpn)**

```
Finished!

The client configuration is available in: /root/pve.ovpn
New clients can be added by running this script again.
```

Tap??????

```
apt-get install -y bridge-utils
```

□□□□□□□□□ **bridge(br0)□□** `/etc/network/interfaces`

```
auto lo
iface lo inet loopback
#auto eth0
#iface eth0 inet dhcp
auto br0
iface br0 inet static
address 192.168.x.x
netmask 255.255.255.0
gateway 192.168.x.x
network 192.168.x.0
broadcast 192.168.x.255
bridge_ports eth0
#auto eth0
#iface eth0 inet static
```

```
#       address 192.168.x.x/24
#       gateway 192.168.x.x
```

```
service networking restart
```

????????

□□ `/etc/openvpn/server/server.conf`

```
local 192.168.x.x
port 1195
proto tcp
dev tap0
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
server-bridge
push "route 192.168.x.0 255.255.255.0"
route-gateway 192.168.x.x
keepalive 10 120
user nobody
group nogroup
cipher AES-256-CBC
persist-key
persist-tun
verb 3
explicit-exit-notify 1
script-security 2
up "openvpn_up br0"
down "openvpn_down br0"
cipher AES-256-CBC
comp-lzo
status /var/log/openvpn/status.log
duplicate-cn  <<<□□□□□□□□□□□□□□□□
```

```
service openvpn-server@server restart
```

□□ `/etc/openvpn/server/openvpn_up`

```
#!/bin/sh
BR=$1
DEV=$2
MTU=$3
/sbin/ifconfig $DEV mtu $MTU promisc up
/sbin/brctl addif $BR $DEV
/sbin/ifconfig tap0 up
```

□□ `/etc/openvpn/server/openvpn_down`

```
#!/bin/sh
BR=$1
DEV=$2
/sbin/brctl delif $BR $DEV
/sbin/ifconfig $DEV down
```

????????

```
□□comp-lzo□□□□
```

Tun??????

```
local 10.0.0.5
push "route 10.0.0.0 255.255.255.0" (□□□□)
push "route 10.8.0.0 255.255.255.0" (□□□□)
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
server 10.8.0.0 255.255.255.0 (□□□□)
keepalive 10 120
```

```
user nobody
group nogroup
persist-key
persist-tun
verb 3
crl-verify crl.pem
explicit-exit-notify
comp-lzo (□□□□)
log-append /var/log/openvpn/op_server.log (□□log)
status /var/log/openvpn/status.log (□□□□)
duplicate-cn  <<<□□□□□□□□□□□□□□□□□
```

```
service openvpn-server@server restart
```

> □□□□□□□□

```
DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --cipher for cipher negotiations. Add 'AES-256-CBC' to --data-ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning.
WARNING: 'link-mtu' is used inconsistently, local='link-mtu 1602', remote='link-mtu 1586'
WARNING: 'keysize' is used inconsistently, local='keysize 256', remote='keysize 128'
WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
```

Server□□□□□

```
cipher AES-256-CBC
```

Client□□□□□

```
cipher AES-256-CBC
□□
data-ciphers-fallback AES-256-CBC

auth-nocache
```

Linux TACACS+ Server ??

??TACACS+

Ubuntu 18

```
apt-get install tacacs+
```

ubuntu 20□□□□□　　tacacs+□□□□□□□□□□□□

```
apt-get update
cd ~
wget http://www.pro-bono-publico.de/projects/src/DEVEL.tar.bz2
bzip2 -dc DEVEL.tar.bz2 | tar xvfp -
cd PROJECTS
make
make install
mkdir /var/log/tac_plus
mkdir /var/log/tac_plus/access
mkdir /var/log/tac_plus/accounting
mkdir /var/log/tac_plus/authentication
cp tac_plus/extra/etc_init.d_tac_plus /etc/init.d/tac_plus
chmod +x /etc/init.d/tac_plus
```

DEVEL.tar.bz2

??tac_plus

sample□□□　　/usr/local/etc/mavis/sample/tac_plus.cfg

□□□□□□　　/usr/local/etc/tac_plus.cfg

```
#!/usr/local/sbin/tac_plus
id = spawnd {
    listen = { port = 49 }
}
id = tac_plus {
    # Log files
    authentication log = /var/log/tac_plus/authentication.log
```

```
    accounting log = /var/log/tac_plus/accounting.log
    authorization log = /var/log/tac_plus/authorization.log
    retire limit = 3000
    # Define external authentication module
    mavis module = external {
        exec = /usr/local/lib/mavis/mavis_tacplus_passwd.pl
    }
    # Authentication backend
    login backend = mavis
    # Default host for all connections
    host = 0.0.0.0/0 {
        key = "test"
    }
    group = admin {
                default service = permit
                service = shell {
                        default command = permit
                        default attribute = permit
                        set priv-lvl = 15
                }
        }
    user = jason {
                password = crypt <□□□□>
                member = admin
                service = junos-exec {
                    set local-user-name =  SUPER
                }
        }
}
```

□□□□

```
openssl passwd -crypt <□□>
```

```
systemctl enable tac_plus
systemctl restart tac_plus
systemctl status tac_plus
```

Linux Cpufrequtils CPU????

□□ cpufrequtils□□□□□□□□□□

??CPU??

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_driver
```

□□□□ `intel_pstate` □□□□□□□□□□□□ `intel-cpufreq` □□□□

□□ `/etc/default/grub` □□ **GRUB_CMDLINE_LINUX_DEFAULT**□□ **intel_pstate=disable**

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash intel_pstate=disable"

update-grub
reboot
```

□□□□□□ `intel-cpufre`

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_driver
```

??Cpufrequtils

```
apt-get install cpufrequtils
```

□□□□ **cpu**□□□

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
conservative ondemand userspace powersave performance schedutil
```

□□ **CPU**□□

```
cpufreq-info
```

```
root@pve:~# cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: intel_cpufreq
```

```
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 20.0 us.
  hardware limits: 1.20 GHz - 3.50 GHz
  available cpufreq governors: conservative, ondemand, userspace, powersave, performance,
schedutil
  current policy: frequency should be within 1.20 GHz and 3.50 GHz.
                  The governor "ondemand" may decide which speed to use
                  within this range.
  current CPU frequency is 3.05 GHz.
```

□□□□

□□ **/etc/init.d/cpufrequtils**

```
ENABLE="true"
GOVERNOR="powersave"     <<<□□□□ ondemand□powersave
MAX_SPEED="0"
MIN_SPEED="0"
```

```
systemctl daemon-reload
/etc/init.d/cpufrequtils restart
```

```
root@pve:~# cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: intel_cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 20.0 us.
  hardware limits: 1.20 GHz - 3.50 GHz
  available cpufreq governors: conservative, ondemand, userspace, powersave, performance,
schedutil
  current policy: frequency should be within 1.20 GHz and 3.50 GHz.
                  The governor "powersave" may decide which speed to use
                  within this range.
  current CPU frequency is 1.20 GHz.
```

Linux MegaRaid CLI ?????

> □□ **LSI MegaRAID□□□□□□□ RAID□□□□□□ RAID□□□**

?? megacli apt repo

□□ `/etc/apt/sources.list`

```
deb http://hwraid.le-vert.net/debian bullseye main
```

??megacli apt key

```
wget https://hwraid.le-vert.net/debian/hwraid.le-vert.net.gpg.key
apt-key add hwraid.le-vert.net.gpg.key
```

?? megaraid tools

```
apt update
apt install megacli
apt install megaclisas-status
```

□□□□□□
□□ **RAID □□□□**
megacli -AdpAllInfo -aAll
□□□□
□□ **RAID □□**
megaclisas-status
□□□□
□□ **RAID □□□□□□ smart □□ Exp. RAID □ /dev/sdb □□□□□□□□□□**
smartctl -a -d megaraid,0 /dev/sdb

Linux R620 ??IPMI??????

??IPMI??

R620□□□□□□ root/calvin

```
ipmitool -I lanplus -H 192.168.1.149 -U root -P calvin raw
```

????

□□ 10□ 16□□□□

```
ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x01 0x00 □□□□□

ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x02 0xff 0x1e 30%□□

ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x02 0xff 0x0a 10%□□

ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x02 0xff 0x0f 15%□□
ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x02 0xff 0x64 100%□□

ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x02 0xff 0x32 50%□□

ipmitool -I lanplus -H 192.168.1.149 -U root -P clavin raw 0x30 0x30 0x01 0x01 □□□□□
```

??IPMI??

```
ipmitool lan print 1
```

□□□□□

```
ipmitool -I open user list 1
```

```
root@server:~# ipmitool -I open user list 1
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv Limit
1                    true    false      false      NO ACCESS
2   root             true    true       true       ADMINISTRATOR
```

□□ root□□

```
ipmitool -I open user set password 2
```

Linux tcpdump????

tcpdump -i ethX icmp and icmp[icmptype]=icmp-echo

查看 tcpdump 可以監聽的接口列表:
tcpdump -D

在接口 eth0 上監聽:
tcpdump -i eth0

監聽任何可用的接口（不能在混雜模式下完成。需要 Linux 內核 2.2 或更高版本）:
tcpdump -i any

捕獲數據包時要詳細:
tcpdump -v

捕獲數據包時更詳細:
tcpdump -vv

捕獲數據包時要非常詳細:
tcpdump -vvv

詳細並以十六進制和 ASCII 格式打印每個數據包的數據，不包括鏈路級標頭:
tcpdump -v -X

詳細並以十六進制和 ASCII 格式打印每個數據包的數據，還包括鏈路級標頭:
tcpdump -v -XX

捕獲數據包時不那麼冗長（比默認設置）:
tcpdump -q

將捕獲限制為 100 個數據包:
tcpdump -c 100

將數據包捕獲記錄到名為 capture.cap 的文件中:
tcpdump -w capture.cap

將數據包捕獲記錄到一個名為 capture.cap 的文件中，但會在屏幕上顯示實時捕獲了多少數據包:
tcpdump -v -w capture.cap

顯示一個名為 capture.cap 的文件的數據包:
tcpdump -r capture.cap

□□□□ capture.cap □□□□□□□□□□□□□□□□□
tcpdump -vvv -r capture.cap

□□□□□□□□ IP □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ -nn □□□□□□□□
tcpdump -n

□□□□□□□ 192.168.1.1 □□□□□□□□□ IP □□□□□□□
tcpdump -n dst host 192.168.1.1

□□□□□□ 192.168.1.1 □□□□□□□□□ IP □□□□□□□
tcpdump -n src host 192.168.1.1

□□□□□□□□ 192.168.1.1 □□□□□□□□□ IP □□□□□□□
tcpdump -n host 192.168.1.1

□□□□□□□ 192.168.1.0/24 □□□□□□□□□ IP □□□□□□□
tcpdump -n dst net 192.168.1.0/24

□□□□□□ 192.168.1.0/24 □□□□□□□□□ IP □□□□□□□
tcpdump -n src net 192.168.1.0/24

□□□□□□□□ 192.168.1.0/24 □□□□□□□□□ IP □□□□□□□
tcpdump -n net 192.168.1.0/24

□□□□□□□ 23 □□□□□□□□□ IP □□□□□□□
tcpdump -n dst port 23

□□□□□□□□ 1 □ 1023□□□□□□□□□□□□□□□ IP □□□□□□□
tcpdump -n dst portrange 1-1023

□□□□□□□□□ 1 □ 1023□□□□□□ TCP □□□□□□ IP □□□□□□□
tcpdump -n tcp dst portrange 1-1023

□□□□□□□□□ 1 □ 1023□□□□□□ UDP □□□□□□ IP □□□□□□□
tcpdump -n udp dst portrange 1-1023

□□□□ IP □ 192.168.1.1 □□□□□□ 23 □□□□□□□□□ IP □□□□□□□
tcpdump -n “dst host 192.168.1.1 and dst port 23”

□□□□ IP □ 192.168.1.1 □□□□□□ 80 □ 443 □□□□□□□□□ IP □□□□□□□
tcpdump -n “dst host 192.168.1.1 and (dst port 80 and dst port 443)”

□□□□ ICMP □□□□
tcpdump -v icmp

□□□□ ARP □□□□
tcpdump -v arp

□□ ICMP □ ARP □□□□
tcpdump -v “icmp or arp”

□□□□□□□□□□□□□□
tcpdump -n “broadcast or multicast”

□□□□□□□□ 500 □□□□□□□□□□□□ 68 □□□
tcpdump -s 500

□□□□□□□□□□□□□□
tcpdump -s 0

Linux ??Trash ???

□□ **Trash □□□□□□** **RM □□□□□**

??Trash-Cli

```
apt install trash-cli
```

??

□□□□

```
trash 123.txt
```

□□□□□

```
trash-list
```

□□□□□

```
trash-empty
```

□□□□□□□□ `~/.local/share/Trash/files/`

RM????Trash

□□ `/etc/profile.d/00-aliases.sh`

```
alias rm='trash'
```

Linux Looking Glass IP????

????

- **Debian 12**
- **PHP 8**

??Looking Glass

□□ **hybula**□□□□□□□□ **PHP 8**

NGINX□□

```
server {
□listen 80;
□listen [::]:80;

□server_name _;

□root /var/www/html/lookingglass/;
□index index.php ;
□location / {
□□try_files $uri $uri/ =404;
□}

    location ~ \.php$ {
□□include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
    }
}
```

□ `config.dist.php`□□ `config.php`

`ping` □□□□□□□ **www-data** □□□□□□□ **/bin/ping**□□□

□□□□□□□□□□□ **ping root**□□□□□□□ **visudo**

```
chmod u+s /bin/ping
```

□□ **visudo** □□ **www-data**□□ **ping**

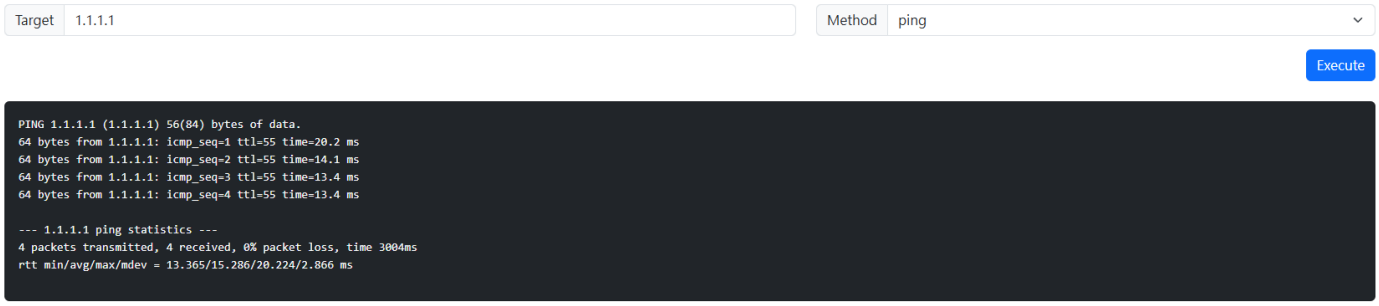
```
visudo

#□□□□□
www-data ALL=(ALL) NOPASSWD: /bin/ping
```

□□ www-data□□□ ping

```
user@test:~# sudo -u www-data ping 1.1.1.1
ping: socktype: SOCK_RAW
ping: socket: Operation not permitted
ping: => missing cap_net_raw+p capability or setuid?


user@test:~# sudo -u www-data sudo ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=12.7 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=13.3 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=13.7 ms
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 12.700/13.254/13.747/0.429 ms
```

□□ `cap_net_raw+p`□□□ □□ `LookingGlass.php`

```
return self::procExecute('ping -4 -c'.$count.' -w15', $host);
□□
return self::procExecute('sudo ping -4 -c'.$count.' -w15', $host);
```



Linux SSH KEY ?????

建立 **Key**

```
ssh-keygen
```

複製到 **Server**上

```
ssh-copy-id -i your_key_path username@server_host
```

手動複製

```
cat ~/.ssh/id_rsa.pub | ssh user@192.168.0.100 'mkdir -p ~/.ssh && chmod 700 ~/.ssh && cat >>
~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys'
```

驗證 `ssh username@server_host` 可直接登入

若登入失敗修正權限

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '/var/services/homes/user/.ssh/id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/var/services/homes/user/.ssh/id_rsa": bad permissions



因為 私鑰 (id_rsa) 權限太寬 (0777), SSH 出於安全性會直接拒絕使用這把 key

# 設定 .ssh 資料夾權限
chmod 700 ~/.ssh

# 設定私鑰權限 (只能自己讀寫)
chmod 600 ~/.ssh/id_rsa

# 公鑰可讀
chmod 644 ~/.ssh/id_rsa.pub
```

Linux X11 Forwarding ????

Windows??

xming□□□□□□

??X11????

```
apt install x11-apps


apt install xauth
```

□□ SSH□□

```
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
```

```
systemctl restart sshd
```

Putty??

□□ ssh x11 enable□□□ IP□ SSH□□□□□ DISPLAY□□□

```
root@x11:# echo $DISPLAY
localhost:10.0
```

```
root@x11:~# xeyes


windows□□□□□□□□□□□□
```

Linux TAB?????

□□□ tab□□□□□□□□□□□□

```
apt-get install bash-completion
```

□□ shell

```
echo "source /etc/bash_completion" >> ~/.bashrc


source /etc/bash_completion
```

Linux SSH ?? OTP ????

??Qrencode?OATH?pam

```
apt install qrencode oathtool libpam-oath
```

??oathtool

□□□□□□□□□□□□□□□□□□□□□□□□ `HEX_SECRET`

```
export HEX_SECRET=$(head -15 /dev/urandom | sha256sum | cut -b 1-30)
```

□□□□□ OTP□□

```
oathtool --totp=SHA256 --verbose --digits=6 $HEX_SECRET
```

□□ Base32 secret□□□□ Qrencode□□□□□□□□

```
root@ssh:~#  oathtool --totp=SHA256 --verbose --digits=6 $HEX_SECRET
Hex secret: 2180dcb885c46412345674e3aa4dba
Base32 secret: EGANZOEFYRddasdsa5STR2UTN2
Digits: 6
Window size: 0
TOTP mode: SHA256
Step size (seconds): 30
Start time: 1970-01-01 00:00:00 UTC (0)
Current time: 2025-10-07 14:56:47 UTC (1759849007)
Counter: 0x37F1B01 (58661633)

961769
```

□□ oath□□

```
touch /etc/users.oath
chmod 0600 /etc/users.oath
```

□□ oath□□

```
echo HOTP/T30 $USER - $HEX_SECRET \ >> /etc/users.oath
```

□□□□□

```
unset HEX_SECRET
```

SSH ??

□ `/etc/pam.d/sshd` □□□□

```
auth required pam_oath.so usersfile=/etc/users.oath
```

□□ `/etc/ssh/sshd_config`

```
ChallengeResponseAuthentication yes
AuthenticationMethods keyboard-interactive:pam
KbdInteractiveAuthentication yes
```

```
service ssh restart
```

??Qrencode

□□ **otp.sh□□□□**

secret□□□ Base32 secret□□

```
#!/bin/bash
secret=EGANZOEFYRddasdsa5STR2UTN2
echo $secret
qrencode -o - $(echo
otpauth://totp/test:test123?secret=${secret}&issuer=test&algorithm=SHA256&digits=6&period=30)
-o qrcode.png -t png -s 5
```

```
test:test123  □□□□
issuer □□□□□
algorithm □□□
digits OTP□□
```

```
period □□□□
```

□□ otp.sh□□□□□□ qrcode□□□□□ OTP APP□□□□

????

□□□□□□□□□ otp□□

```
root@server:~# ssh 192.168.100.100
(root@192.168.100.100) One-time password (OATH) for `root':
(root@192.168.100.100) Password:
```

Linux ?? nftables ? Fail2ban ????

????

- **Debian 12**

`nftables` □ **Linux** □□□□ `iptables` □□□□□□□□□ **NAT** □□□□□ **Linux** □□□□□□□□□ `nftables`□□□□□ `nftables` □ `Fail2ban`□□□□□□□□□

??Fail2ban

```
apt install fail2ban python3-systemd
```

??Fail2ban

□□ `/etc/fail2ban/fail2ban.local`□□□□□ **ipv6**

```
[DEFAULT]
allowipv6 = auto
```

□□ `/etc/fail2ban/jail.local`

```
[DEFAULT]
# Debian 12 has no log files, just journalctl
backend = systemd
# Destination email for action that sends you an email
#destemail = alerts@your-domain.com
# Sender email. Warning: not all actions take this into account. Make sure to test if you rely
on this
#sender    = fail2ban@your-domain.com
# Default action. Will block user and send you an email with whois content and log lines.
action    = %(action_mwl)s
# ignoreip can be a list of IP addresses, CIDR masks, or DNs hosts. Fail2ban
# # will not ban a host which matches an address in this list.
```

```
ignoreip = 127.0.0.1/8 ::1/128

# configure nftables
banaction = nftables-multiport
chain     = input

# regular banning
bantime = 7d
findtime = 1m
maxretry = 3

# "bantime.increment" allows to use database for searching of previously banned ip's to
increase a
# default ban time using special formula, default it is banTime * 1, 2, 4, 8, 16, 32...
bantime.increment = true
# "bantime.rndtime" is the max number of seconds using for mixing with random time
# to prevent "clever" botnets calculate exact time IP can be unbanned again:
bantime.rndtime = 30m
# "bantime.maxtime" is the max number of seconds using the ban time can reach (don't grows
further)
bantime.maxtime = 60d
# "bantime.factor" is a coefficient to calculate exponent growing of the formula or common
multiplier,
# default value of factor is 1 and with default value of formula, the ban time
# grows by 1, 2, 4, 8, 16 ...
bantime.factor = 2

# purge database entries after
dbpurgeage = 30d

[sshd]
mode      = aggressive
enabled   = true
backend   = systemd
port      = 2222
maxretry  = 3
```

- **backend = systemd□ Debian12 □□ journalctl □□ log□□□ systemd**
- **□□□□□□□□□ IPv6 □□□□□**
- **□ nftables □□□□□□□□**
- **□□□□□□□□□□□ port**

□□□ **Recidive□□□□□□□□□□□□□□□□□□□□□□□□□□□□** **fail2ban.log□□□□** **rsyslog**

```
[recidive]
backend = auto
logpath  = /var/log/fail2ban.log
enabled = true
maxretry = 2
banaction = nftables-allports
```

```
systemctl restart fail2ban
systemctl status fail2ban
```

??????

```
fail2ban-client status
fail2ban-client status sshd
fail2ban-client set sshd unbanip <ip>
```

```
root@remote:~# fail2ban-client status
Status
|- Number of jail:      2
`- Jail list:   recidive, sshd
root@remote:~# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:
root@remote:~# fail2ban-client status recidive
Status for the jail: recidive
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- File list:        /var/log/fail2ban.log
`- Actions
```

```
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:
root@remote:~#
```

Linux ?? Chroot

????

- **LXC Debian 12**

??SSH chroot

□□ **chroot jail**

```
mkdir -p /home/user
```

□□□□ `/dev` □□

```
ls -l /dev/{null,zero,stdin,stdout,stderr,random,tty}
```

□□ **LXC□□□□□□** `dev` □□□□□

```
mkdir -p /home/user/dev/

touch /home/user/dev/{null,zero,random,tty}

chmod 666 /home/user/dev/{null,zero,random,tty}
```

□□□□□□□ `mknod`□□□□

```
mkdir -p /home/user/dev/□□
cd /home/user/dev/
mknod -m 666 null c 1 3
mknod -m 666 tty c 5 0
mknod -m 666 zero c 1 5
mknod -m 666 random c 1 8
```

chroot jail □□□□□ **root**□□

```
chown root:root /home/user
chmod 0755 /home/user
ls -ld /home/user
```

```
root@test:~# chown root:root /home/user
root@test:~# chmod 0755 /home/user
root@test:~# ls -ld /home/user
drwxr-xr-x 3 root root 4096 Oct 11 05:24 /home/user
```

??SSH chroot Shell

□□ `bin` □□□□□□ `/bin/bash` □□□□□□□□□□□□□□□□ `/bin/sh` □□□□□□ `/bin/sh` □□□□ `/etc/passwd` □ **Shell**

```
mkdir -p /home/user/bin
cp -v /bin/bash /home/user/bin/
```

```
root@test:~# mkdir -p /home/user/bin
root@test:~# cp -v /bin/bash /home/user/bin/
'/bin/bash' -> '/home/user/bin/bash'
```

□□ **bash**□□□ `lib` □□□□□□□ `lib` □□□□

```
ldd /bin/bash
mkdir -p /home/user/lib/x86_64-linux-gnu
mkdir -p /home/user/lib64

cp -v /lib/x86_64-linux-gnu/libtinfo.so.6 /home/user/lib/x86_64-linux-gnu/
cp -v /lib/x86_64-linux-gnu/libc.so.6 /home/user/lib/x86_64-linux-gnu/
cp -v /lib64/ld-linux-x86-64.so.2 /home/user/lib64/
```

```
root@test:~# ldd /bin/bash
        linux-vdso.so.1 (0x00007ffd43f4f000)
        libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fc543171000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc542f90000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fc5432e9000)
```

```
root@test:~# cp -v /lib/x86_64-linux-gnu/libtinfo.so.6 /home/user/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libtinfo.so.6' -> '/home/user/lib/x86_64-linux-gnu/libtinfo.so.6'
root@test:~# cp -v /lib/x86_64-linux-gnu/libc.so.6 /home/user/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libc.so.6' -> '/home/user/lib/x86_64-linux-gnu/libc.so.6'
root@test:~# cp -v /lib64/ld-linux-x86-64.so.2 /home/user/lib64/
'/lib64/ld-linux-x86-64.so.2' -> '/home/user/lib64/ld-linux-x86-64.so.2'
```

??SSH???

```
useradd user
passwd user
```

□□ `etc` □□□□□□ `/etc/passwd`□`/etc/group`□□□

```
mkdir /home/user/etc
cp -vf /etc/{passwd,group} /home/user/etc/
```

□□□□ **SSH chroot**□□□□ `/etc/passwd`□`/etc/group`□□□□□ `/home/user/etc` □

??SSHD

□□ `/etc/ssh/sshd_config`□□□□□ **ChrootDirectory**□□□ **sftp**□□

```
#Subsystem      sftp    /usr/lib/openssh/sftp-server
Subsystem sftp internal-sftp

Match User user
ChrootDirectory /home/user
```

```
 systemctl restart sshd
```

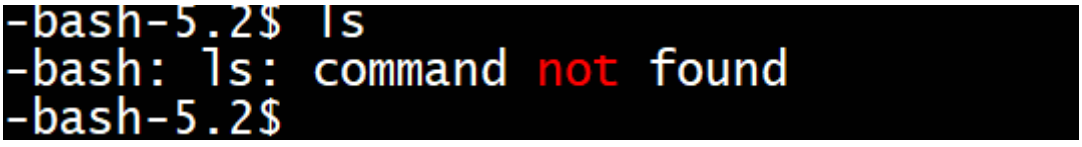
□□□□

```
root@ssh:~# ssh user@192.168.100.1
user@192.168.100.1's password:
-bash-5.2$
```

?? SSH ???????????

```
mkdir -p /home/user/home/user1
chown -R user:user /home/user/home/user1
chmod -R 0700 /home/user/home/user1
```

□□□□□□ **bash**□ **chroot**□□□□□□□□□□□□□□□□□ **bash lib**□□□□□□□□□□□□□ **chroot**□



□□ `/bin/ls`□□□ `ls`□□□ `lib`□□□□□□□ `lib`□□□□

```
root@test:~#  ldd /bin/ls
        linux-vdso.so.1 (0x00007ffd98188000)
        libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007f41fda83000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f41fd8a2000)
        libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007f41fd808000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f41fdadc000)
root@test:~# cp -v /lib/x86_64-linux-gnu/libselinux.so.1 /home/user/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libselinux.so.1' -> '/home/user/lib/x86_64-linux-gnu/libselinux.so.1'
root@test:~# cp -v /lib/x86_64-linux-gnu/libc.so.6 /home/user/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libc.so.6' -> '/home/user/lib/x86_64-linux-gnu/libc.so.6'
root@test:~# cp -v /lib/x86_64-linux-gnu/libpcre2-8.so.0 /home/user/lib/x86_64-linux-gnu/
'/lib/x86_64-linux-gnu/libpcre2-8.so.0' -> '/home/user/lib/x86_64-linux-gnu/libpcre2-8.so.0'
root@test:~# cp -v /lib64/ld-linux-x86-64.so.2 /home/user/lib64/
'/lib64/ld-linux-x86-64.so.2' -> '/home/user/lib64/ld-linux-x86-64.so.2'
root@test:~# cp -v /bin/ls /home/user/bin/
'/bin/ls' -> '/home/user/bin/ls'
```

複製後可使用 `ls` 指令，其餘指令依此步驟安裝

```
-bash-5.2$ ls
bin  dev  etc  home  lib  lib64
```

SFTP測試

```
sftp> ls
bin    dev    etc    home   lib    lib64
sftp> cd /home/
sftp> ls
user1
sftp> cd user1/
sftp> ls
sftp> mkdir 123
sftp> ls
123
```

SFTP是基於 **SSH** 的檔案傳輸協定，只要該使用者對檔案有「寫入權限」且對目錄有「寫入 **+** 執行權限」，就可以操作檔案

所以 **Shell**未安裝 **rm,mkdir**等不能使用，但 **sftp**可以

Linux ?? IPV6

□□ **IPV6□□□□** `/etc/sysctl.conf` □□□□□□□□□□ **IPV6□□**

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

□□ `sysctl -p` □□□□

Linux Syslog-ng ???????

Linux□□□□□□ OpenObserve□□□□□ AxoSyslog□ Syslog-ng□ v4.5□□□ □□□□ OpenObserve□□□□□□□□□□□□□□□□□□□□ syslog-ng□ rsyslog□□□□□ OpenObserve□□□□□□□□□

□□ Server□□□ □□ /etc/syslog-ng/syslog-ng.conf

```
options { keep-hostname(yes);chain_hostnames(off); flush_lines(0); use_dns(no); use_fqdn(no);
          dns_cache(no); owner("root"); group("adm"); perm(0640);
          stats(freq(0)); bad_hostname("^gconfd$");
};
########################################################## options□□□keep-hostname(yes); □□□□□□□□□□
destination d_openobserve_http {
    openobserve-log(
        url("http://192.168.100.1")
        port(5080)
        organization("default")
        stream("syslog-ng")
        user("root@example.com")
        password("b5wE158wERFTaxe0X")
    );
};


log {
    source(s_src);
    destination(d_openobserve_http);
    flags(flow-control);
};
########################################################## □□□□□openobserve□□
source s_test {
    udp(ip(192.168.100.100) port(514));
};
log {
    source(s_test);
    destination(d_openobserve_http);
};
```

```
######################################################## □□□□□□□ udp 514 port □log□□□□openobserve
```

□□□□□

```
# Destinations
########################
# First some standard logfile
#
destination d_remote {
    udp("192.168.100.99" port(514));
};


# Filters
########################
# Here's come the filter options. With this rules, we can set which
# message go where.
filter f_no_temp1 {
    not (
        (program("CRON") and message("temp1.sh"))
        or (program("CRON") and message("smart -u"))
        or message("cron:session")
        or (program("systemd-logind") and message("suspend"))
    );
};


# Log paths
########################
log {
    source(s_src);
    filter(f_no_temp1);
    destination(d_remote);
};
```

?????????Server

####################################
????
??????????
CRON ???? "/root/temp1.sh"?"smart -u"
facility ?"authpriv" ???? "cron:session"
systemd-logind ?? ???? "suspend"

##

□□□□□□□□ filter□□□□□□□□ Server

Linux ?? Python3 FTP

□□□ Python FTP□□□□□□□

??pip

```
apt install python3-pip
```

??pyftpdlib

```
pip3 install pyftpdlib
```

??pyftpdlib

□□□□□□□□□□□

```
python3 -m pyftpdlib -p 21
```

□□□□□□ FTP

```
python3 -m pyftpdlib -p 21 -u test -P 123456 -w
```

Linux Crontab ????

範例

查看自己的 **crontab**
crontab -l
# 查看指定使用者的 **crontab**
sudo crontab -u gtwang -l# 編輯 crontab 內容
crontab -e
編輯指定使用者的 **crontab**
crontab -u gtwang -e#
刪除 **crontab** 內容
crontab -r

┌──────────── 分鐘 **(0 - 59)**
│ ┌──────────── 小時 **(0 - 23)**
│ │ ┌──────────── 日 **(1 - 31)**
│ │ │ ┌──────────── 月 **(1 - 12)**
│ │ │ │ ┌──────────── 星期幾 **(0 - 7, 0** 是週日, **6** 是週六, **7** 也是週日 **)**
│ │ │ │ │
 # * * * * * /path/to/command

每天早上 **8** 點 **30** 分執行
30 08 * * * /home/gtwang/script.sh --your --parameter

每週日下午 **6** 點 **30** 分執行
30 18 * * 0 /home/gtwang/script.sh --your --parameter

每週日下午 **6** 點 **30** 分執行
30 18 * * Sun /home/gtwang/script.sh --your --parameter

每年 **6** 月 **10** 日早上 **8** 點 **30** 分執行
30 08 10 06 * /home/gtwang/script.sh --your --parameter

每月 **1** 日、 **15** 日、 **29** 日晚上 **9** 點 **30** 分各執行一次
30 21 1,15,29 * * /home/gtwang/script.sh --your --parameter

每隔 **10** 分鐘執行一次
***/10 * * * * /home/gtwang/script.sh --your --parameter**

從早上 **9** 點到下午 **6** 點, 凡遇到整點就執行
00 09-18 * * * /home/gtwang/script.sh --your --parameter

Linux ??????????

要建立一個限制極高的專用帳號，使其只能執行 `smartctl -a`，最安全且乾淨的做法是利用 **Linux** 的 **SSH Command Restriction (Force Command)** 機制，搭配 `sudoers` 最小權限設定。

以下是完整的設定步驟：

?? 1????????

在目標 **Linux** 機器上建立一個專門用於監控的帳號（例如命名為 `smartmon`），並停用密碼登入：

```
# 建立使用者 smartmon，禁用密碼並停用標準 shell
sudo useradd -m -s /bin/bash smartmon
```

?? 2??? `sudoers` ????

`smartctl` 需要 `root` 權限才能存取硬碟底層控制介面。透過 `visudo` 允許 `smartmon` 只能免密碼執行 `smartctl`：

1. 開啟 **sudo** 設定檔：

   ```
   sudo visudo -f /etc/sudoers.d/smartmon
   ```

2. 寫入以下內容（嚴格限制只能帶 `-a` 參數，或對所有磁碟裝置放行）：

   程式碼片段

   ```
   # 允許 smartmon 免密碼以 root 權限執行 smartctl -a
   smartmon ALL=(root) NOPASSWD: /usr/sbin/smartctl -a *
   smartmon ALL=(ALL) NOPASSWD: /usr/sbin/smartctl
   ```

?? 3??? SSH Key ????? (Force Command)

這是最關鍵的一步。透過 **SSH** 公鑰中的 `command=` 限制，即使有人取得該公鑰，也絕對無法執行 **SSH** 互動 **Shell** 或任何其他指令。

1. 建立 `.ssh` 目錄：

```
sudo mkdir -p /home/smartmon/.ssh
sudo chmod 700 /home/smartmon/.ssh
```

2. □□ `authorized_keys` □

```
sudo nano /home/smartmon/.ssh/authorized_keys
```

3. □□□□□□□□□□□□□□□□□□□□ □□□□□□ `ssh-ed25519 AAAAC3N...` □□□□□□□ `command="..."` □□□□□□□□

```
command="sudo /usr/sbin/smartctl -a $SSH_ORIGINAL_COMMAND",no-port-forwarding,no-x11-forwarding,no-agent-forwarding,no-pty ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI... (□□□□)
```

> “□□
> - `command="..."` □□ **SSH** □□□□□□ `sudo smartctl` □□□□□□□□□□
> - `no-port-forwarding,no-x11-forwarding,no-agent-forwarding,no-pty` □□□□ **SSH** □□□□ **PTY** □□□□□□□□□□ **Shell**□

4. □□□□□□□□

```
sudo chmod 600 /home/smartmon/.ssh/authorized_keys
sudo chown -R smartmon:smartmon /home/smartmon/.ssh
```

????

□□□□□□□□□□□□□□□□□□

1. ?????? S.M.A.R.T. ????????

```
ssh smartmon@<□□IP> /dev/sda
```

```
[redacted]:~# ssh smartmon@[redacted] /dev/disk/by-id/ata-Samsung_SSD_860_EVO_250GB_S3Y9NX0M550465V
smartctl 7.3 2022-02-28 r5338 [x86_64-linux-6.8.12-9-pve] (local build)
Copyright (C) 2002-22, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:     Samsung based SSDs
Device Model:     Samsung SSD 860 EVO 250GB
Serial Number:    S3Y9NX0M550465V
LU WWN Device Id: 5 002538 e410acb64
Firmware Version: RVT03B6Q
User Capacity:    250,059,350,016 bytes [250 GB]
```